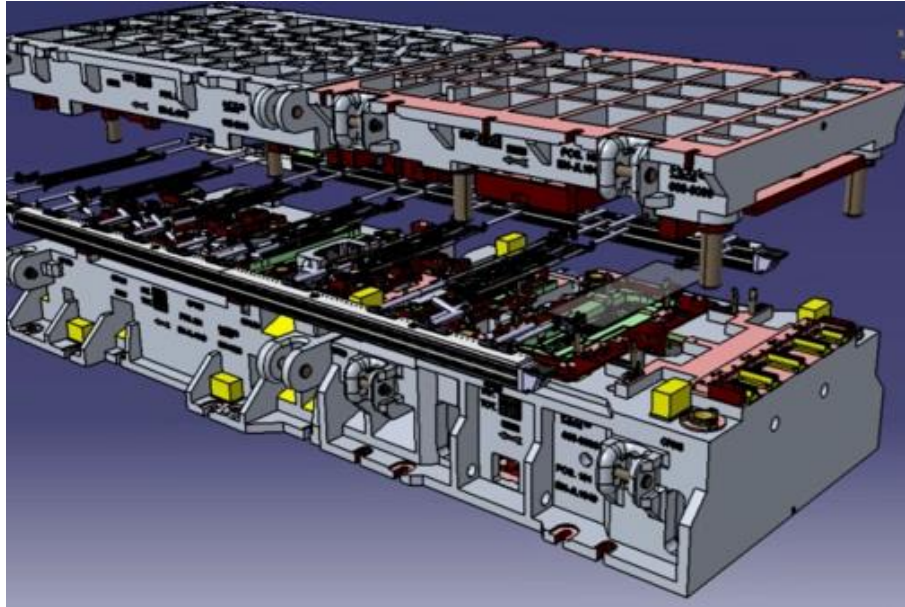




ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Departamento de Engenharia Mecânica



Validação Cinemática Virtual de um Sistema Transfer

Luís Miguel Vicente Alves Ferreira
(Licenciado em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre
em Engenharia Mecânica

(Documento Definitivo)

Orientadores:

Prof. Doutor João Filipe de Almeida Milho
Prof. Doutora Marta Isabel Pimenta Verdete da Silva
Carvalho

Júri:

Presidente: Professor Doutor Silvério João Crespo Marques
Vogais:

Professor Doutor Ricardo José Fontes Portal
Prof. Doutor João Filipe de Almeida Milho

Agosto de 2018



ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Departamento de Engenharia Mecânica

Validação Cinemática Virtual de um Sistema Transfer

Luís Miguel Vicente Alves Ferreira
(Licenciado em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre
em Engenharia Mecânica

(Documento Definitivo)

Orientadores:

Prof. Doutor João Filipe de Almeida Milho
Prof. Doutora Marta Isabel Pimenta Verdete da Silva
Carvalho

Júri:

Presidente: Professor Doutor Silvério João Crespo Marques

Vogais:

Professor Doutor Ricardo José Fontes Portal
Prof. Doutor João Filipe de Almeida Milho

Agosto de 2018

Agradecimentos

Gostaria de deixar aqui uma breve mensagem de agradecimento a todas as pessoas que de forma direta ou indiretamente me deram o seu apoio e tornaram a realização deste trabalho possível.

Gostaria de deixar também um agradecimento à MCG, mind for metal e a todas as pessoas que nela trabalham e que fizeram com que fosse possível o desenvolvimento deste trabalho.

Por fim, gostaria também de deixar aqui um agradecimento a todas as pessoas que já passaram pela minha vida e que de alguma forma contribuíram para me tornar a pessoa que sou hoje.

Um muito obrigado a todos.

Resumo

A validação cinemática em ambiente virtual tem, nos dias de hoje, um papel bastante preponderante na indústria, pois com o aumento da competitividade das empresas existe uma enorme pressão para a redução dos prazos de entrega de ferramentas de estampagem. Assim com o recurso à validação cinemática é possível avaliar, antes de qualquer aço maquinado, colisões assim como zonas de possíveis colisões.

Sendo a oferta de softwares que permitam realizar este tipo de validação cinemática em ambiente virtual por simulação bastante vasta, foi em conjunto com uma empresa que opera no desenvolvimento de ferramentas de estampagem, definido que seria avaliado um software utilizado na empresa, o DS CATIA V5. Para isso iniciou-se a avaliação do módulo de simulação cinemática o DMU Kinematics, tendo sido posteriormente lançado o desafio de realizar este tipo de simulação numa licença base do software.

Para responder a este desafio foi desenvolvida uma metodologia com recurso a uma licença base do software DS CATIA V5 aliado a um algoritmo de otimização baseado em algoritmos genéticos que garanta um caminho realizado pelo sistema transfer sem colisões e com a cadência máxima.

Com a implementação desta metodologia e algoritmo, foi possível verificar que o algoritmo cumpre os objetivos definidos para aplicação da metodologia em ambiente de produção de ferramentas transfer, reduzindo assim o tempo necessário para a simulação de cerca de 8 horas de trabalho manual (aproximadamente estimado 1 dia de trabalho) utilizando o módulo DMU kinematics para cerca de 1h25 minutos utilizando a metodologia desenvolvida neste TFM.

Palavras Chave

- Sistema transfer
- Validação Cinemática
- Ds CATIA
- Colisão
- Otimização
- Algoritmos Genéticos

Abstract

Nowadays, virtual kinematic simulation has a major role in the industry due pressure to reduce lead times of stamping tools. Therefore by using this type of simulations it is possible to assess, before of any machining, if there will be collisions and tight spots.

With a quite large number of softwares that supply this type of simulation, it was decided, in partnership with a stamping tool company, that the software they were currently using was the one to use, DS CATIA V5. The evaluation started with the DMU kinematics module. Moreover was launched a challenge to find a way to perform this type of simulations in a standard license of DS CATIA V5.

To ensure this, a methodology was developed, that with a standard license of DS CATIA V5 and allied to an optimization algorithm based in genetic algorithms that ensure a path for the transfer system without any collisions at the highest press pace.

With the implementation of this methodology was possible to verify that the algorithm accomplished the main objectives, reducing the required time to obtain a path free from collisions from 8 hours, using the DMU kinematics module to about 1h25 minutes.

Keywords

- Transfer system
- Kinematic validation
- Ds CATIA
- Collision
- Optimization
- Genetic Algorithms

Índice

Capítulo 1 – Introdução	1
1.1. O problema	1
1.2. Objetivos.....	1
1.3. Motivações	2
1.4. Organização do documento	2
Capítulo 2 – Enquadramento teórico e pesquisa bibliográfica	4
2.1. As peças estampadas.....	4
2.2. Os processos de conformação de chapa mais comuns.....	5
2.2.1.Estampagem	5
2.2.2.Corte por arrombamento	7
2.2.3.Dobragem	8
2.3. Ferramentas.....	9
2.3.1.Progressivas	10
2.3.2.Transfer.....	10
2.4. Prensas.....	11
2.4.1.Estrutura das Prensas	12
2.4.2.Prensas Hidráulicas.....	13
2.4.3.Prensas Mecânicas	15
2.4.4.Prensas Hidráulicas versus Prensas Mecânicas (Vantagens e Desvantagens)	19
2.4.5.Linhas de Prensas (‘Press lines’)	21
2.4.6.Automatização.....	22
2.4.7.Prensas Transfer	24
2.4.8.Prensas Transfer ‘Crossbar’	27
2.5. Simulação, validação e otimização do sistema transfer.....	30

2.6. Otimização com Algoritmos Genéticos	33
2.6.1.Função objetivo	35
2.6.2.Operadores Genéticos.....	35
2.6.2.1.Seleção.....	35
2.6.2.3.Crossover	36
2.6.2.4.Mutação.....	37
2.6.2.5.Elitismo	38
Capítulo 3 – Modelação de uma ferramenta transfer para validação cinemática virtual.....	39
3.1. Projeto de uma ferramenta transfer	39
3.1.1.Construção do ‘Strip-layout’.....	39
3.1.2Construção da ferramenta	42
3.2 Simulação cinemática de uma ferramenta transfer.....	44
3.2.1Modelação com Ds CATIA V5 – Módulo DMU Kinematics.....	44
3.2.2Modelação com Ds CATIA V5 – Versão Base.....	48
Capítulo 4 – Metodologia de validação cinemática virtual de uma ferramenta transfer	55
4.1. Primeira Abordagem / Algoritmo I	56
4.2. Segunda Abordagem / Algoritmo II.....	59
4.3. Terceira Abordagem / Algoritmo III.....	62
4.4. Quarta Abordagem / Algoritmo IV	63
4.5. Quinta Abordagem / Algoritmo V	66
4.6. Sexta Abordagem / Algoritmo VI.....	70
4.7. Sétima Abordagem/ Algoritmo VII	74
Capítulo 5 – Conclusões.....	81
Capítulo 6 - trabalhos futuros.....	84

Bibliografia.....	85
Anexo A - População testada na 7ª Abordagem/ algoritmo	88
Anexo B – Código VBA utilizado na 7ª Abordagem/algoritmo.....	90

Índice de Figuras

Figura 1- Esquema de uma operação de estampar ([2]).	6
Figura 2 - Aplicação da meia cana no encostador e matriz (Adaptado de [3]).	7
Figura 3 - Ilustração de corte por arrombamento.	7
Figura 4 - Diferente fases das operações de corte (adaptação de [4]).	8
Figura 5 - Esquema de uma operação de dobra (adaptado de [2]).	9
Figura 6 – Esquema de uma banda ([5]).	10
Figura 7 – Esquema de uma banda de uma ferramenta transfer	10
Figura 8 - Design da estrutura de uma prensa direita ([4]).	12
Figura 9 - Princípio funcional de um cilindro hidráulico atuando um teto de uma prensa hidráulica ([4])	14
Figura 10 - Diagrama de Posição(curso) em função do tempo de uma prensa hidráulica ([4])	15
Figura 11 - Quatro tipos de acionamento e arranjo da embraiagem: (a) Acionamento sem engrenagem ou por volante de inercia; (b) Redução única, acionamento por uma engrenagem, Embraiagem na roda dentada; (c) Redução simples, acionamento por duas rodas dentadas, Embraiagem no veio principal; (d) Reduções múltiplas, acionamento por duas engrenagens, Embraiagem no veio intermédio. (Adaptado de [6]).	16
Figura 12 – Classificação da prensas controladas por curso, baseado no seu acionamento principal (Retirado de [2]).	17
Figura 13 – Esquema de três acionamentos por cambota (Adaptado de [2]).	18
Figura 14 - Diagrama de curso em função do tempo - Comparação entre o movimento realizado pelo teto de uma prensa de acionamento direto por cambota, Knuckle joint e link drive (Adaptado de [4]).	18
Figura 15 - Deslocamento do teto e potencia requerida em função do tempo em prensas mecânicas e hidráulicas com o respetivo ciclo e tempo de transporte (Retirado de[4]).	19
Figura 16 - Esquema de uma ‘press line’ automatizada com robôs (Retirado de [4])...	22
Figura 17 - Esquema de uma ‘press line’ com sistema de ‘swing arm feeder’ (Retirado de [4]).	23

Figura 18 - Esquema de uma 'press line' com sistema de alimentação CNC (Retirado de [4]).....	23
Figura 19 - Esquema do transporte de uma peça utilizando o sistema transfer tridimensional (Retirado de [4]).	26
Figura 20 - Diagrama de movimento de um sistema transfer tridimensional (Retirado de [4]).....	27
Figura 21 - Esquema de transporte de uma peça de elevadas dimensões numa prensa com sistema transfer 'crossbar' (Retirado de [4]).	27
Figura 22 - Esquema do posicionamento das ventosas no sistema de transfer 'crossbar' (Retirado de [4])......	28
Figura 23 - Prensa com sistema transfer 'crossbar' – Prensa com 5 tetos individuais (Retirada de [4]).....	29
Figura 24 - Diagrama de movimento de um sistema transfer 'crossbar' (Retirado de [4]).	30
Figura 25 - Esquema de uma fábrica para a produção de painéis de grande dimensão, incluindo uma zona de simulação cinemática (Retirado de [4]).	31
Figura 26 - Código genérico de implementação do método de seleção por roleta (Retirado de [10]).	36
Figura 27 - Esquema da utilização do crossover com um ponto de crossover (Retirado de [13]).	37
Figura 28 – Esquema da utilização da mutação (Retirado de [13]).....	38
Figura 29- Peça em estudo	39
Figura 30 - Superfície obtida utilizando a função "Extract"	40
Figura 31- Superfície da peça, na linha média.....	40
Figura 32 -Planificado da peça a desenvolver	41
Figura 33 - Planificado Final.....	41
Figura 34- Strip-layout do processo produtivo e fases de transformação do componente	42
Figura 35- Representação dos vários elementos da ferramenta. (a) Esquema de montagem dos vários elementos de uma ferramenta (b) Matrizes (c) Punções e porta Punções (d) Pisas e porta pisas (e) Cerras	43
Figura 36- Menu principal CATIA V5 DMU Kinematics.....	44

Figura 37 - Menu de juntas cinemáticas do Ds CATIA DMU Kinematics.....	45
Figura 38 - Representação da junta cilíndrica entre base e topo da prensa.....	45
Figura 39- Representação da junta rígida entre a ferramenta e a prensa.....	46
Figura 40- Curva cinemática da prensa, deslocamento vs. tempo	47
Figura 41 - Deslocamento do transfer segundo o eixo xx, deslocamento vs. tempo. ...	47
Figura 42 – Modelo CAD da ferramenta. (a) parte superior da ferramenta. (b) parte inferior da ferramenta.....	49
Figura 43 - Modelo CAD dos componentes da ferramenta. (a) – Sistema transfer triaxial. (b) - Pisa. (c) - Cerra. (d) - Elevador.....	49
Figura 44 – Modelo CAD utilizado para simular a ferramenta e o sistema transfer.....	50
Figura 45 - Diagrama cinemático da prensa, curso da parte superior em relação à base da prensa em função do ângulo da prensa.	51
Figura 46- Diagrama cinemático do pisa.	51
Figura 47 - Esquema demonstrativo dos ângulos de atuação dos movimentos (Adaptado de [4]). θ_1 – Ângulo final do avanço; θ_2 - Ângulo inicial do recuo; θ_3 – Ângulo final do recuo; θ_4 – Ângulo inicial do avanço.....	53
Figura 48 - Esquema tridimensional dos movimentos individuais do sistema transfer nos eixos xx, yy e zz.	53
Figura 49 – Esquema da interação utilizada entre softwares	56
Figura 50 – Deslocamento de um sistema transfer em função do ângulo da cambota da prensa durante um ciclo.....	58
Figura 51 - Esquema das variáveis de projeto utilizadas.....	59
Figura 52 - Fluxograma do algoritmo II.	61
Figura 53 - Fluxograma do algoritmo IV.	64
Figura 54 - Fluxograma do algoritmo V.	67
Figura 55 - Estrato da população testada, disponível no Anexo A.....	75
Figura 56 - Evolução do número de penalizações devido à violação do constrangimento da velocidade máxima.....	78
Figura 57 - Evolução do número de penalizações devido à violação do constrangimento da ausência de colisão.....	78
Figura 58 - Comparação do valor da função objetivo da melhor solução da 10ª geração em contraste com o as soluções da população inicial (1ª geração).	79

Índice de Tabelas

Tabela 1 - Vantagens e desvantagens de uma prensa hidráulica de acionamento simples (Adaptado de [6]).	20
Tabela 2 - Vantagens e desvantagens de uma prensa mecânica de acionamento simples (Adaptado de[6]).	21
Tabela 3 - Cursos dos componentes e cadência da ferramenta	52
Tabela 4 - Esquema cinemático dos movimentos Y1, Z1 e X1 do modelo simplificado	54
Tabela 5 - Esquema cinemático dos movimentos Z2, Y2 e X2 do modelo simplificado	54
Tabela 6 - Valores considerados para as variáveis de projeto	60
Tabela 7- Tabela 7 - Resultados do algoritmo II.	61
Tabela 8 - Resultados da implementação do algoritmo III.....	62
Tabela 9 - Resultados da implementação do algoritmo IV	65
Tabela 10- Valores das variáveis de projeto na primeira população	68
Tabela 11– Valores das variáveis de projeto após a 5 geração.....	69
Tabela 12 - Constrangimentos aplicados às variáveis de projeto	70
Tabela 13 – População Inicial criada com a sexta abordagem / algoritmo VI.	71
Tabela 14 – População após a 10ª geração	72
Tabela 15 - População criada.....	75
Tabela 16 - População inicial	76
Tabela 17 - População após 10 gerações	77

Capítulo 1 – Introdução

1.1. O problema

Nos dias de hoje devido à pressão dos mercados, para a redução dos prazos de entrega de ferramentas de estampagem, é necessário reduzir também o tempo disponível para projeto da ferramenta, maquinação e montagem, sendo cada vez mais importante que se consiga simular com rapidez alguns destes dos processos.

No projeto de ferramentas com sistema transfer, o transporte da peça a produzir entre as várias estações de processamento da ferramenta é realizado por um sistema mecânico. Neste caso, é vantajoso que seja realizada uma simulação cinemática como forma de garantir que o sistema de transporte da peça, designado sistema transfer, não colide com a ferramenta e/ou a peça e que não existe também colisões da peça com a ferramenta.

Existem no mercado vários programas de simulação que permitem realizar este tipo de análise cinemática, porém além do seu custo ser bastante elevado, associado ao custo de aquisição e manutenção de licença, ainda existe o facto destes softwares comerciais não permitirem a pesquisa de parâmetros ótimos de funcionamento da ferramenta com vista a uma melhor performance.

1.2. Objetivos

Neste Trabalho Final de Mestrado (TFM) será desenvolvido uma metodologia de otimização utilizando uma licença base do software comercial Ds CATIA que permita realizar a validação cinemática e a obtenção dos parâmetros ótimos da ferramenta garantindo que o caminho realizado pelo sistema transfer não apresenta colisões com à cadência máxima.

A metodologia será baseada na criação de um modelo de simulação cinemática de um sistema transfer e posteriormente na utilização de um algoritmo com base em algoritmos genéticos para a determinação os ângulos de ativação/paragem de cada um dos 3 eixos de movimentação de um sistema transfer tridimensional.

Sendo este TFM realizado em parceria com a empresa MCG, Mind for Metal, o objetivo passa por realizar uma aplicação, que possa ser rapidamente utilizada na

licença base do software Ds CATIA V5 já utilizado na empresa para projeto mecânico das ferramentas. Assim pretende-se que durante a fase de projeto seja realizada este tipo de simulação garantindo que o sistema transfer irá conseguir garantir a transferência da peça, sem que ocorram colisões. Espera-se também que esta solução possa ser utilizada quando exista dúvida sobre qual a melhor disposição dos componentes na ferramenta, podendo rapidamente obter-se o design ótimo, o que permitirá atingir uma maior cadência.

1.3. Motivações

A utilização de softwares de simulação é cada vez mais uma realidade das empresas que queiram manter-se competitivas. A motivação para a realização deste TFM, nasce na necessidade de criar um método que permita não só garantir um transporte de peças pelo sistema transfer sem a ocorrência de colisões, como o facto de ser realizado através de uma licença base de um software comercial, num contexto real empresarial

1.4. Organização do documento

Este documento está dividido em seis capítulos. Sendo o primeiro capítulo de carácter introdutório, seguido do segundo onde se apresentam os fundamentos teóricos e o enquadramento dos trabalhos realizados em metodologias de otimização de simulação de sistemas de transfer, objeto deste TFM.

No terceiro capítulo são apresentados os passos a realizar durante o projeto de uma ferramenta do tipo transfer assim como quais as abordagens para a validação cinemática virtual utilizando o modulo DMU Kinematics e uma licença base do software Ds CATIA v5.

No quarto capítulo, é apresentada a metodologia desenvolvida para validação cinemática virtual de uma ferramenta transfer, consistindo em varias abordagens e algoritmos assim como a análise dos resultados.

Segue-se o quinto capítulo, de conclusões onde se efetua uma comparação das vantagens que esta abordagem traz relativamente à utilização de outros métodos e um sexto capítulo, onde são apresentadas ideias para trabalhos futuros.

No decorrer deste documento optou-se por utilizar alguns termos em língua inglesa específicos da área de estudo em que se enquadra este TFM. Esta decisão foi tomada tomando em consideração que muitos destes termos são usualmente utilizados em ambiente empresarial/industrial em língua inglesa e não em língua portuguesa.

Capítulo 2 – Enquadramento teórico e pesquisa bibliográfica

2.1. As peças estampadas

Na indústria automóvel, uma grande maioria das peças utilizadas na montagem de um automóvel são produzidas através da transformação de chapa de metal, nas ferramentas com recurso a prensas. Nestas prensas são realizadas varias operações de transformação (estampagem, cortes, abocadados, dobras, etc..) da chapa de forma a criar a peça final.

A estampagem é um processo de enformação de chapa amplamente utilizado na indústria automóvel, consistindo na deformação plástica, normalmente de uma chapa ou rolo de chapa, aplicada pelas prensas. É um processo bastante complexo, devido ao elevado número de variáveis que definem o processo, porém estas podem ser agrupadas, de uma forma sumária, em quatro categorias e respetivas características:

1) Características da peça:

- Propriedades mecânicas do material
- Espessura do material
- Geometria da peça

2) Características de punções e matrizes:

- Superfícies de acabamento
- Limpeza da superfície
- Geometria

3) Características da prensa:

- Velocidade de funcionamento
- Modos de Vibração

4) Variáveis de interação:

- Lubrificação
- Força do encostador

Existem assim várias características e variáveis que condicionam todo o processo de estampagem.

2.2. Os processos de conformação de chapa mais comuns

É importante referenciar que existem vários tipos de ferramentas, mas de uma forma geral as ferramentas utilizadas na indústria automóvel, podem ser divididas em dois grandes grupos:

- 1) As que realizam apenas uma estação por ferramenta, normalmente utilizadas em peças de elevadas dimensões como portas e laterais da carroceria;
- 2) As que realizam várias estações por ferramenta, normalmente peças de reforço com dimensões mais reduzidas, sendo que este tipo de ferramentas podem ser ferramentas do tipo transfer ou ferramentas progressivas, conceitos que serão introduzidos posteriormente neste documento.

Independentemente do número de estações por ferramenta, a grande maioria das peças produzidas tem, pelo menos, uma estação de estampar, seguida de várias estações de corte, dobragem entre outros.

Serão apresentadas seguidamente os processos de conformação de chapa mais utilizados na indústria.

2.2.1. Estampagem

O processo de estampagem permite conferir formas complexas às chapas. Neste tipo de operação existem componentes que se mantêm comuns nomeadamente uma matriz '*die*', um punção '*punch*' e, poderá ter ou não, um encostador '*blank holder*', sendo que entre a matriz e o punção existirá uma folga que no limite será a espessura da chapa, conforme representado na Figura 1.

O encostador também designado por cerra ou '*blank holder*' é um elemento que tem como principal função controlar o escoamento da chapa durante a operação de estampagem. Sendo o controlo deste escoamento realizado através da aplicação de pressão sobre a chapa criando assim o atrito necessário para contrariar o referido escoamento [1].

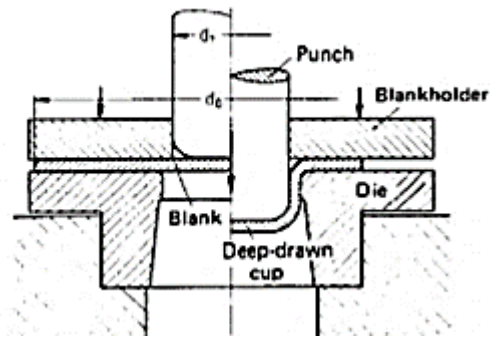


Figura 1- Esquema de uma operação de estampar ([2]).

Devido à complexidade deste tipo de operações, a sequenciação do processo deverá ser a seguinte:

1. A chapa é colocada sobre a matriz ou 'die'
2. O encostador ou '*blank holder*' começa a exercer pressão sobre a chapa antes do punção ou 'Punch' toque na chapa.
3. O punção inicia a conformação da chapa forçando-a a escoar entre o punção e a matriz;
4. Ao terminar a conformação da chapa entre o punção e a matriz termina também a operação de estampagem.

Seguidamente será necessário extrair a peça do exterior do punção ou do interior da matriz para tal será utilizado um sistema de extração que irá expulsar mecanicamente a peça estampada.

Em componentes complexos, geralmente, são utilizados cerras segmentados produzindo uma distribuição não uniforme de força sob a peça. Permitindo assim variar o escoamento de metal em diferentes locais da peça.

Outra forma de limitar o escoamento do metal durante o processo é utilizando meias canas no encostador e matriz. Tal como se pode verificar na figura 2, a introdução da meia cana obriga o material a passar pelos pontos A, B, C, D, E e F, obrigando o material a percorrer este caminho, é criada uma força de atrito, causando assim uma limitação do escoamento do material [3].

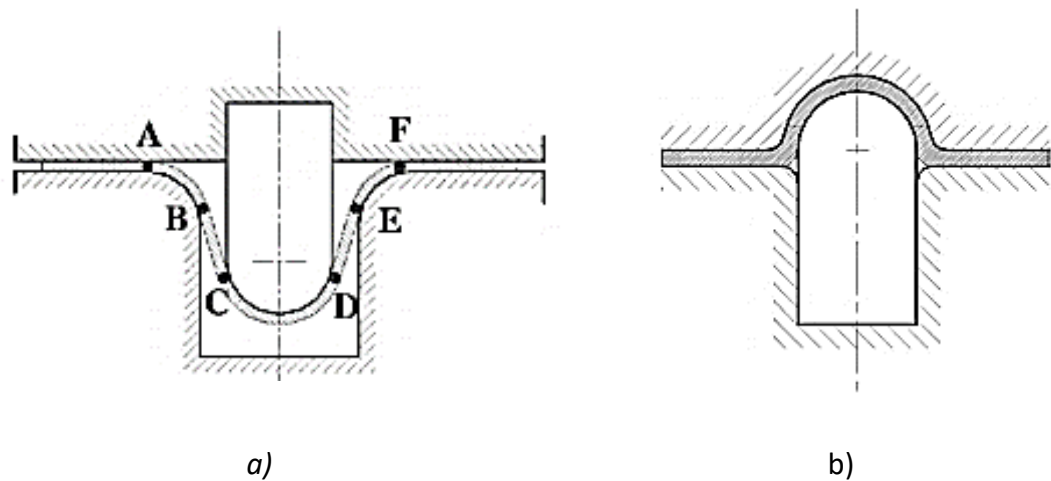


Figura 2 - Aplicação da meia cana no encostador e matriz (Adaptado de [3]).

2.2.2. Corte por arrombamento

O processo de corte por arrombamento consiste na remoção de material presente no formato inicial, desnecessário à produção da peça final. Estas remoções podem ser feitas ao longo do contorno exterior da peça, assim como no interior dos contornos da mesma, sob a forma de furos, rasgos, entre outros.

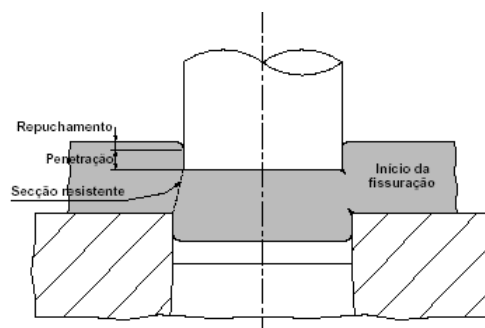


Figura 3 - Ilustração de corte por arrombamento

O processo de corte por arrombamento é tipicamente efetuado antes das operações de dobragem e estampagem, quando a chapa ainda se encontra plana. No entanto, em alguns casos específicos, pode ser vantajoso conferir primeiro alguma geometria à peça final e só posteriormente executar as operações de corte, conferindo assim às peças uma menor deformação dos contornos de determinados cortes, furos ou rasgos.

Mais em pormenor, o processo de corte por arrombamento é executado por dois componentes principais, o punção de corte e a matriz de corte, sendo que em casos particulares poderão existir também o encostador ou *'blank holder'* e os extratores. (de notar que na indústria, se utiliza a terminologia de pisas para operações onde os encostadores são utilizados em operações de corte ou dobra e cerras, quando utilizados em operações de estampagem). Entre a aresta de corte do punção e a da matriz existirá uma folga que será definida como folga de corte.

Com base na figura 4, pode-se definir claramente que este tipo de processo pode ser dividido em 5 fases distintas:

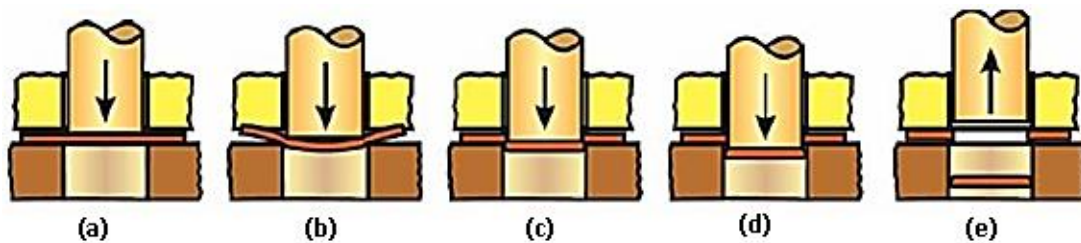


Figura 4 - Diferente fases das operações de corte (adaptação de [4]).

Numa primeira fase existe o encosto do encostador sobre a chapa, que se encontra sobre a matriz, sendo seguido pelo contacto entre o punção e a chapa, que provoca uma força de compressão, Figura 4a. Essa força de compressão leva a uma ligeira flexão da chapa no inicio da deformação plástica Figura 4b. Essa flexão ocorre devido à força de compressão aplicada pelo punção e à existência de folga de corte, faz com que ocorra uma deformação plástica do material com a formação de fendas, Figura 4c. Quando a deformação atinge o limite de rotura do material, dá-se a separação completa, Figura 4d. A operação de corte fica finalizada com a separação do retraço do punção, definido pelo pedaço de chapa que após a operação de corte é descartado, a extração do punção do interior da chapa e a inversão do movimento do punção, Figura 4e.

2.2.3. Dobragem

O processo de dobragem, consiste em efetuar dobras em chapas, normalmente feito com recurso a prensas, ou quinadeiras. Neste tipo de operações o punção de dobra ou

'punch' sujeitará a chapa a dobrar sobre o raio de canto de matriz de dobra ou 'die', sendo que entre ambos deverá figurar uma folga que no limite deverá ter o valor da espessura da chapa, tal como representado na Figura 5.

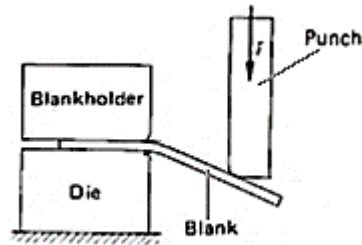


Figura 5 - Esquema de uma operação de dobra (adaptado de [2]).

Neste processo, existem dois fatores importantes, o raio de curvatura e a elasticidade do material. Devem-se evitar cantos vivos, sendo que normalmente se utilizam raios de curvatura que correspondem a uma a duas vezes a espessura da chapa para materiais dúcteis, e três a quatro vezes para materiais mais frágeis.

2.3. Ferramentas de Estampagem

Na estampagem de chapa, a geometria da peça é completamente definida pela geometria da matriz. Devido à elevada complexidade das peças, a estampagem não é normalmente realizada em apenas uma operação. Dependendo da geometria da peça, a sua produção é alcançada através de várias operações que pode conter um ou vários processos, tais como a estampagem ou o corte [4].

Na fase de projeto, onde é definido o número de operações a realizar, o principal objetivo é diminuir o número de operações sem que se comprometa a qualidade da peça produzida. Obviamente quanto maior o número de operações mais complexa e cara será a ferramenta.

O tipo de ferramenta e o tipo de transporte relativo entre operações é determinado de acordo com a forma de estampar, o tamanho da peça a produzir e o volume a produzir.

2.3.1. Progressivas

Uma ferramenta progressiva é um dispositivo mecânico que tem como objetivo transformar uma tira de chapa, num componente. As ferramentas progressivas consistem em várias estações individuais onde cada uma delas realiza uma ou mais operações na peça. A sua característica é que o transporte da peça de estação para estação é realizado pela banda, ou 'strip layout', até a operação final onde existe a separação [5], conforme exemplificado na Figura 6.

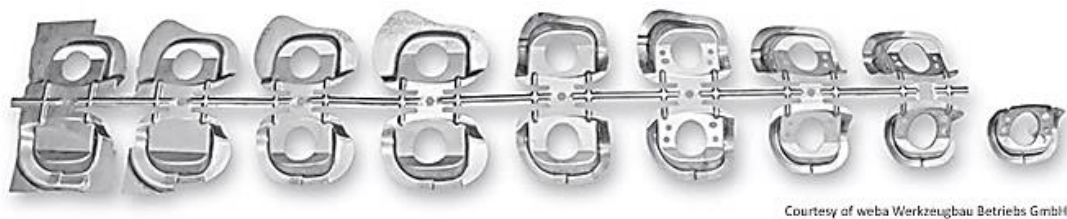


Figura 6 – Esquema de uma banda ([5])

As ferramentas progressivas são utilizadas sempre que se pretenda a produção de um elevado número de peças, mantendo um custo baixo, pelo que as suas cadencias são elevadas.

2.3.2. Transfer

As ferramentas do tipo transfer têm um funcionamento em todo semelhante as ferramentas progressivas, sendo a principal diferença o facto da peça não estar ligada por uma banda e o transporte da peça entre estações ser realizado por um sistema mecânico, conforme representado na Figura 7.

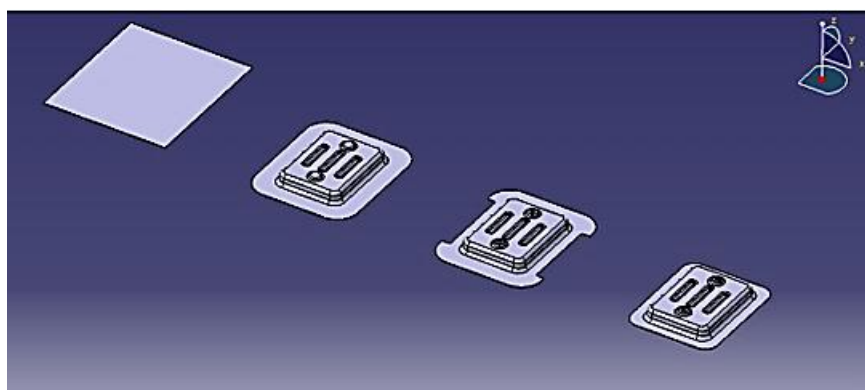


Figura 7 – Esquema de uma banda de uma ferramenta transfer

O facto deste tipo de ferramenta não carecer de uma ligação física entre as várias estações, leva a uma diminuição do consumo de chapa e conseqüentemente uma diminuição do preço da peça. A utilização deste tipo de ferramentas permite também realizar operações que numa ferramenta progressiva seriam impossíveis devido aos constrangimentos de espaço e da ligação das estações.

As várias estações deste tipo de ferramentas encontram-se dispostas sequencialmente na ferramenta, tal como acontece nas ferramentas progressivas, separadas por si por uma distância fixa, denominada como o passo da ferramenta. No entanto a progressão da chapa no interior da ferramenta é conseguida à custa de um sistema mecânico, que movimenta os vários formatos ao longo da ferramenta. Esta atuação da movimentação dos formatos ao longo da ferramenta é sincronizada com o funcionamento da prensa, de modo a que se consiga reduzir os tempos mortos e assim rentabilizar o funcionamento da mesma.

2.4. Prensas

Tal como visto anteriormente, para realizar a conformação de chapa são necessárias ferramentas e prensas. No que às prensas diz respeito, existe uma enorme variedade pelo podem ser classificadas em três grupos tendo como base parâmetros característicos tais como [2] :

- 1) de energia e força
- 2) dependentes de tempo
- 3) de precisão em carga em repouso

Em qualquer processo de conformação de chapa, uma quantidade de força é transmitida durante um período de tempo, pela ferramenta para a peça. Isto requer, portanto, uma quantidade de energia, que irá depender do tipo de processo de transformação [6]. Existe uma grande variedade de categorias de máquinas, onde a principal diferença é a fonte de energia que utilizam, sendo que as mais utilizadas na indústria automóvel são [7]:

- 1) Prensas Mecânicas – onde a força que irá produzir trabalho provem de algum componente mecânico, normalmente uma cambota ou alavanca.

- 2) Pressas hidráulicas – onde a força que irá produzir trabalho provém de um fluído sob pressão.

2.4.1. Estrutura das Pressas

A função da estrutura da prensa é absorver as forças, para providenciar um guiamento preciso ao teto e dar suporte ao sistema de transmissão de energia e outros elementos secundários. Assim o desenho estrutural depende essencialmente de:

- 1) Força de pressão – Que irá definir a rigidez necessária
- 2) As dimensões da ferramenta
- 3) A acessibilidade à área de trabalho, que determina a forma da estrutura
- 4) O grau de precisão requerido para o guiamento, que influencia não só a forma como a rigidez da estrutura.

As pressas com uma força de pressão nominal superior a 4000 kN, são construídas exclusivamente utilizando o design em forma de pórtico, conforme exemplificado na Figura 8. Este tipo de pressas é conhecido como prensa direita [4].

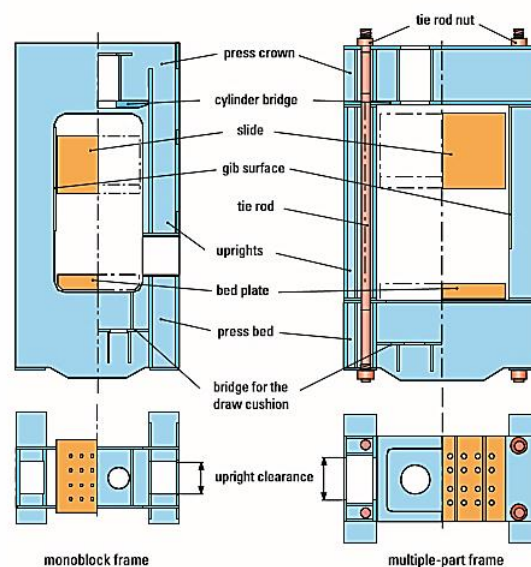


Figura 8 - Design da estrutura de uma prensa direita ([4]).

Neste tipo de prensa, a cama da prensa '*press bed*', a base da prensa '*bed plate*', as 4 colunas verticais '*uprights*' e o topo da prensa '*press crown*' formam a estrutura da prensa, [4].

Tal como referido anteriormente uma ferramenta, poderá ser composta por uma estação ou por várias, mas independentemente do número de estações, a ferramenta será sempre composta por uma parte superior e uma parte inferior. De acordo com a Figura 7, a parte superior é fixada ao teto da prensa 'slider', e a parte inferior fixa à base da prensa 'bed plate'.

2.4.2. Prensas Hidráulicas

A primeira patente de uma prensa hidráulica aparece no ano 1795, através do industrial inglês Joseph Bramah (1749-1814), sendo uma aplicação direta da experiência realizada em 1650 pelo físico e matemático francês Blaise Pascal (1623-1662). Pascal descobriu que aplicando uma pressão a um fluido que se encontre no seu estado de repouso num recipiente fechado será transmitida, sem qualquer perda, a todas as moléculas de fluido assim como às paredes do recipiente [2,8].

As prensas hidráulicas acabam por ter uma utilização bastante elevada devido à sua capacidade de gerar elevadas forças de pressão, obtida através do bombeamento de fluido a elevada pressão para um ou mais pistões, convertendo a energia hidráulica em trabalho mecânico. A maioria das prensas hidráulicas, possui uma bomba hidráulica de pistão concêntrico que permite obter volume e pressão variável, providenciando assim velocidades de abertura e de fecho rápidas e também, uma velocidade de trabalho lenta com a máxima força de pressão [2,6].

A potência P [W] de uma prensa hidráulica, depende do caudal de fluido hidráulico \dot{V} [m³/s] e conseqüentemente da velocidade do teto, assim como da pressão hidráulica p , das forças que atuam no teto, e das perdas mecânicas e hidráulicas η_{ges} [4], de acordo com a equação 1.

$$P = \frac{\dot{V} \cdot p}{\eta_{ges}} \quad (1)$$

Nas prensas hidráulicas a direção do fluxo é realizada num sentido e posteriormente revertida para fechar e abrir a prensa, tal como se pretende ilustrar através da figura 9.

Inicialmente para realizar o fecho do teto, o fluido hidráulico é bombeado para dentro da câmara do cilindro, obtendo um aumento de caudal que se traduz num aumento de velocidade (zona definida como 'high speed closure'), sendo esse fluido, posteriormente comprimido para que se obtenha a força de pressão máxima (zona definida como 'forming') e posteriormente o fluxo é invertido, passando a ser criada pressão na parte inferior do cilindro, sendo o fluido conduzido novamente para fora da câmara do cilindro, o que provoca novamente um aumento da velocidade (zona definida como 'return stroke') [4].

Em suma, numa prensa hidráulica as forças geradas são diretamente proporcionais ao caudal de fluido utilizado, os cursos que podem ser utilizados serão atingidos através da utilização de mais ou menos fluido, a velocidade do teto é diretamente proporcional ao caudal de fluido e os movimentos de fecho e abertura são controlados com a inversão do fluxo do fluido [4].

Tendo como base os pressupostos anteriormente descritos, a potência da prensa, é obtida com base no curso efetuado pelo teto em função do tempo, que é dependente de constantes de tempo definidas para cada ferramenta, tais como o curso total, a força necessária tanto para a estampagem como para outras operações e a força dos encostadores. Cada uma destas variáveis tem um impacto direto no tempo de ciclo e conseqüentemente no rendimento da prensa [4].

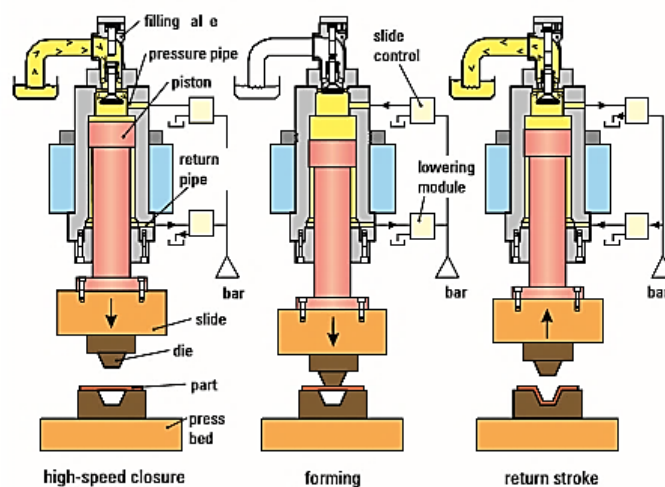


Figura 9 - Princípio funcional de um cilindro hidráulico atuando um teto de uma prensa hidráulica ([4])

Com base no diagrama de posição (curso) em função do tempo de uma prensa hidráulica, tal como representado na figura 10, pode concluir-se que o tempo de ciclo da ferramenta será composto pelo somatório de vários troços temporais. O tempo de ciclo definirá o tempo que a peça demorará a fazer. Analisando o diagrama de curso em função do tempo pode-se identificar que nas prensas hidráulicas existe uma zona de abrandamento, 'braking' antes de entrar na zona de estampagem. Esse abrandamento permite que a zona definida para a estampagem seja realizada a uma velocidade constante, que permite que o material escoe naturalmente.

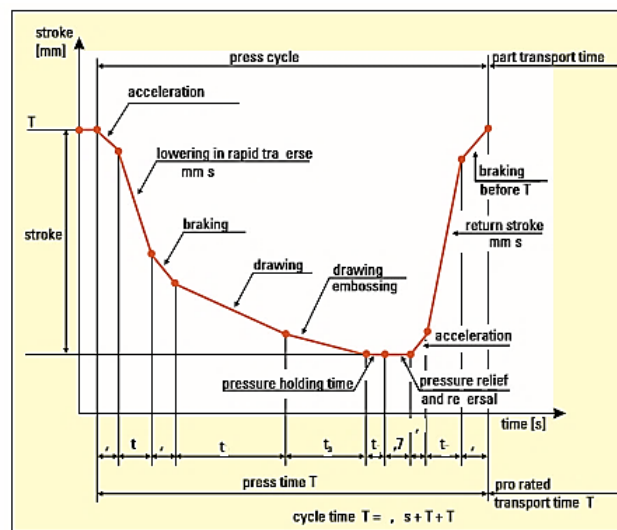


Figura 10 - Diagrama de Posição(curso) em função do tempo de uma prensa hidráulica ([4])

2.4.3. Prensas Mecânicas

As prensas mecânicas fazem parte do conjunto de prensas controladas por curso, são por isso das mais utilizadas na indústria, devido a sua elevada versatilidade. Existem dois grandes grupos de prensas mecânicas [2]:

- 1) Prensas de acionamento por cambota 'crank drive'
- 2) Prensas de acionamento por excêntricos 'cam drive'

O sistema que permite o acionamento das prensas mecânicas é constituído por um motor elétrico, um volante de inércia, uma embraiagem e um travão e em alguns casos sistemas de engrenagens, conforme representado na Figura 11.

As prensas com cambota tanto podem ter cambotas simples como mecanismo de extensão. Em prensas com cambota convencionais o curso é fixo enquanto em prensas excêntricas o curso total pode variar. Estas últimas pertencem às prensas de acionamento simples. Se uma articulação ou alavanca é utilizada para aumentar o curso das prensas com acionamento por cambota, passam a chamar-se prensas de *'knuckle joint'* ou *'link drive'* tal como esquematizado na Figura 11.

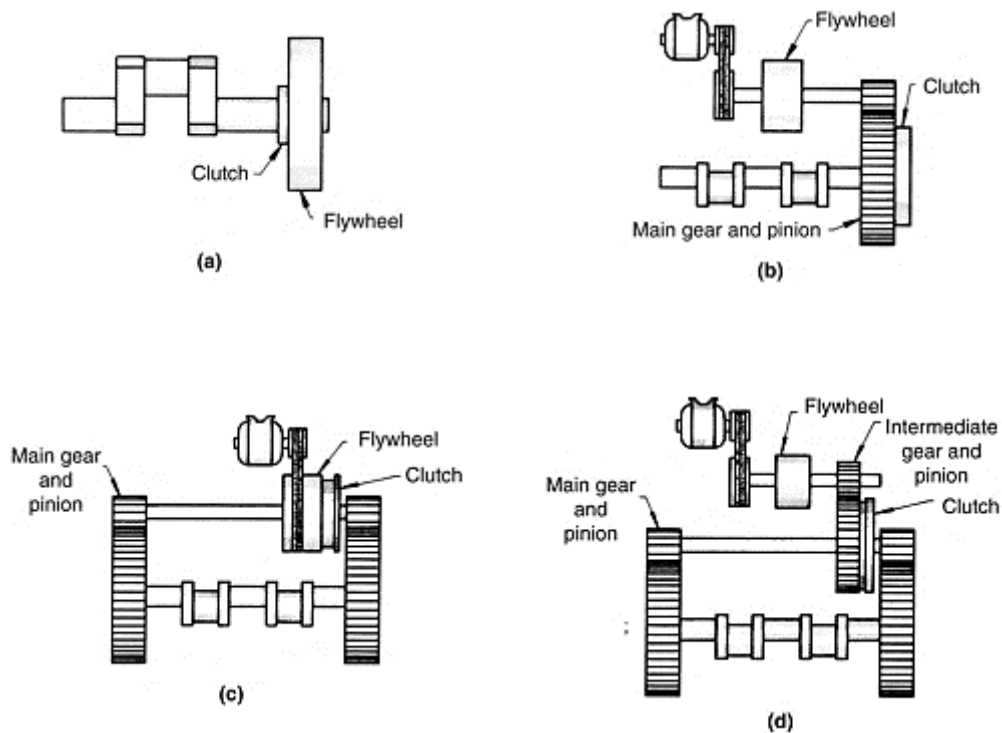


Figura 11 - Quatro tipos de acionamento e arranjo da embraiagem: (a) Acionamento sem engrenagem ou por volante de inercia; (b) Redução única, acionamento por uma engrenagem, Embraiagem na roda dentada; (c) Redução simples, acionamento por duas rodas dentadas, Embraiagem no veio principal; (d) Reduções múltiplas, acionamento por duas engrenagens, Embraiagem no veio intermédio. (Adaptado de [6]).

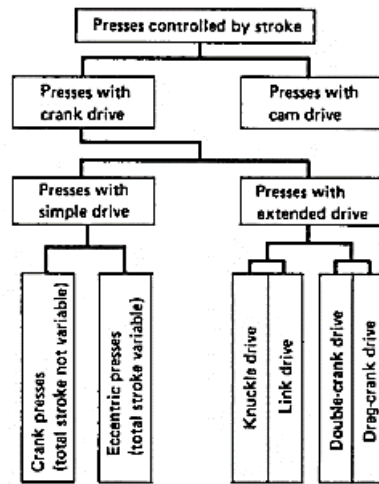


Figura 12 – Classificação da prensas controladas por curso, baseado no seu acionamento principal (Retirado de [2]).

Na maioria das prensas mecânicas o volante de inércia é a principal fonte de energia que é aplicada à cambota, engrenagens, excêntrico ou às interligações durante o ciclo. O volante de inércia roda continuamente, por intermédio de um motor elétrico, e é acionado através de uma embraiagem quando é necessário realizar um ciclo [2].

O comportamento cinemático de uma prensa mecânica será influenciado pelo tipo de acionamento. Uma vez que as prensas mecânicas de acionamento simples são as mais usuais na indústria automóvel, são estas que irão ser abordadas neste TFM, sendo que na Figura 13 estão representados os esquemas de acionamento de cada um dos três acionamentos simples.

O acionamento direto por cambota, foi durante vários anos o sistema de acionamento mais utilizado em prensas mecânicas. O movimento sinusoidal característico deste tipo de acionamento pode ser visto na Figura 13. A utilização deste tipo de acionamento em operações de estampagem tem alguns lados negativos nomeadamente a velocidade de impacto durante o fecho da ferramenta ser relativamente elevada e a diminuição da velocidade do teto, desvantagens que levam a que este tipo de acionamento seja normalmente evitado em operações de estampagem profunda a elevada cadencia[4].

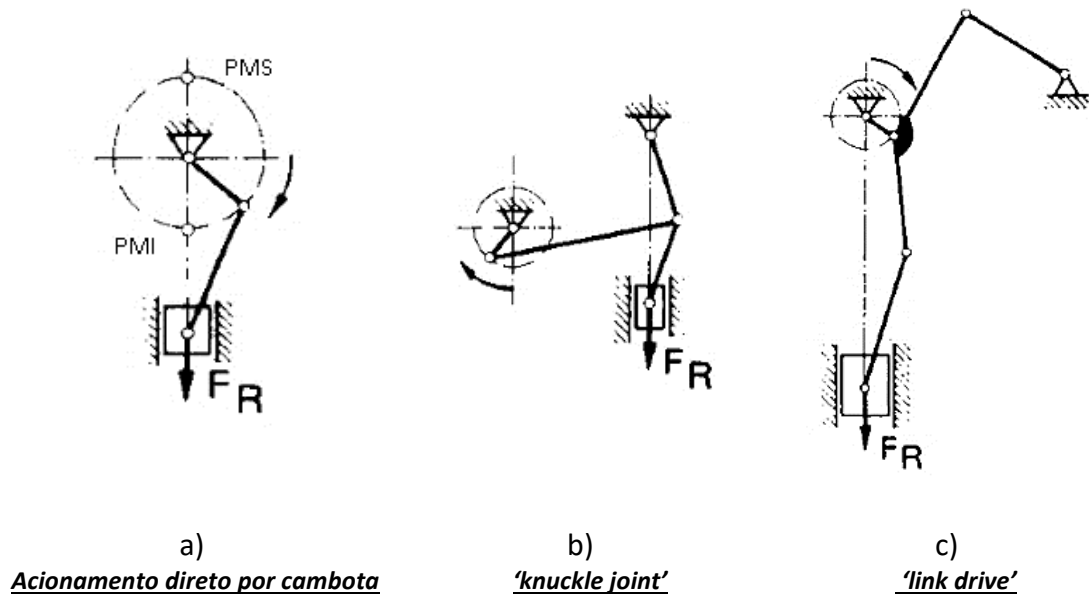


Figura 13 – Esquema de três acionamentos por cambota (Adaptado de [2]).

Contudo, em prensas com uma força nominal superior a 5000 kN, o acionamento direto por cambota é o sistema mais eficiente, principalmente quando se utilizam sistemas automatizados, onde o acionamento direto por cambota oferece o melhor compromisso entre o tempo necessário para o transporte da peça e o tempo necessário para realizar a operação [4].

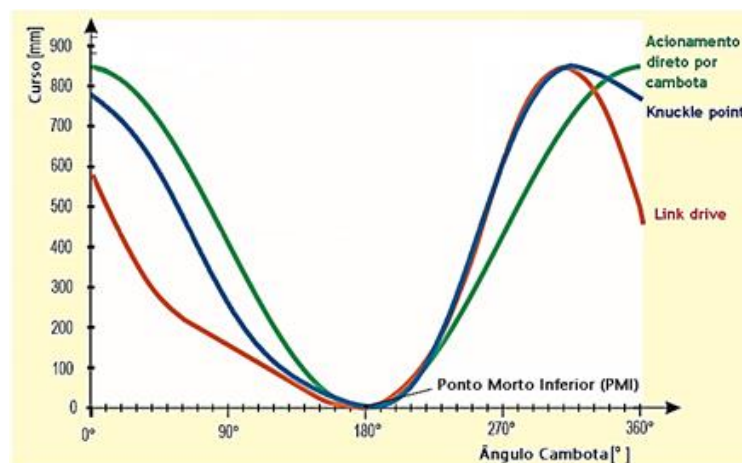


Figura 14 - Diagrama de curso em função do tempo - Comparação entre o movimento realizado pelo teto de uma prensa de acionamento direto por cambota, Knuckle joint e link drive (Adaptado de [4]).

2.4.4. Comparação entre Prensas Hidráulicas e Prensas Mecânicas (Vantagens e Desvantagens)

Comparando as prensas mecânicas com as hidráulicas e considerando um curso de estampagem de 100 mm, a curva de curso em função do tempo, o tempo requerido para um ciclo e o tempo de transporte da peça estão representados através da figura 15.

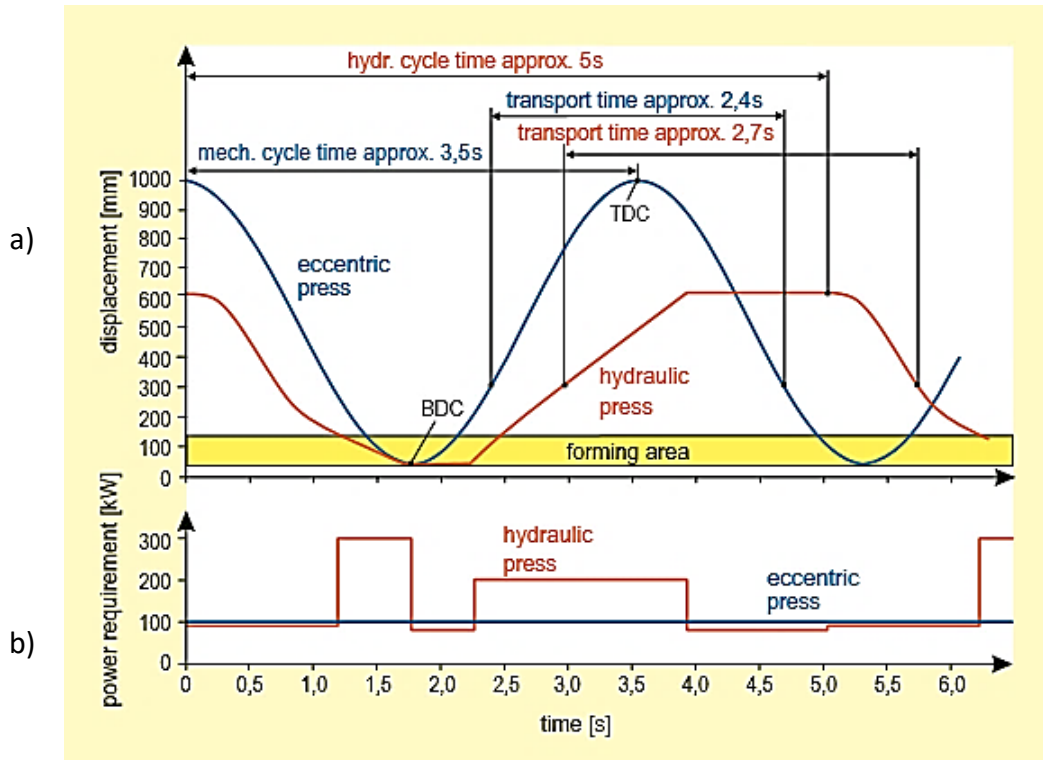


Figura 15 - Deslocamento do teto e potencia requerida em função do tempo em prensas mecânicas e hidráulicas com o respetivo ciclo e tempo de transporte (Retirado de[4]).

Analisando a Figura 15 pode-se verificar que a prensa mecânica se encontra a trabalhar com um tempo de ciclo de 3.5s o que corresponde a cerca de 17 golpes por minuto (GPM) enquanto a prensa hidráulica apresenta um tempo de ciclo de 5s o que corresponde a apenas a 12 GPM. Uma vez que nas prensas hidráulicas a velocidade do teto é ajustada ao processo, isso produz uma velocidade no impacto muito menor quando comparada com a velocidade de impacto de uma prensa mecânica. Com essa redução na velocidade de impacto consegue-se obter menores tensões no material da

ferramenta e um melhor escoamento do material, porém estas vantagens levam a um aumento considerável do tempo de ciclo.

Além das diferenças notórias em termos de tempo de ciclo, estas duas prensas possuem também uma diferença considerável no consumo de energia elétrica. Tal como visto anteriormente, nas prensas mecânicas a energia provém do volante de inércia, o que causa uma diminuição da velocidade desta (prensa mecânica). Assim, existe a necessidade do motor garantir o fornecimento da energia gasta em cada ciclo, criando assim um consumo constante de energia elétrica. Nas prensas hidráulicas, por outro lado, a velocidade do teto é controlada diretamente pelo motor elétrico, assim tal como demonstrado pela Figura 15 e pela equação 1, sempre que existe um aumento da velocidade e ou velocidade exigida ao sistema levará a um aumento do consumo elétrico.

Seguidamente são apresentadas as tabelas 1 e 2 onde se encontram, de forma sumária, as principais vantagens e desvantagens da utilização de prensas hidráulicas e prensas mecânicas, respetivamente.

Tabela 1 - Vantagens e desvantagens de uma prensa hidráulica de acionamento simples (Adaptado de [6]).

<u>Prensas Hidráulicas</u>	
<u>Vantagens:</u>	<u>Desvantagens:</u>
<ul style="list-style-type: none"> - A prensa não pode ser sobrecarregada - Permite obter a pressão máxima em todo o curso - Permite o ajuste da pressão exercida até ao máximo permitido pela máquina - A velocidade mantém-se constante em operações de estampagem. 	<ul style="list-style-type: none"> - As prensas hidráulicas não necessitam de tanta energia elétrica quando comparada às prensas mecânicas, mas necessitam de um motor muito maior quando comparado com o necessário para uma prensa mecânica. - Por norma não são utilizadas para processos de corte, pois o choque resultante é prejudicial a sistemas como as tubagens, conceções da prensa. - A sua manutenção é mais complexa quando comparada à das prensas mecânicas.

Tabela 2 - Vantagens e desvantagens de uma prensa mecânica de acionamento simples (Adaptado de[6]).

<u>Prensas Mecânicas</u>	
<u>Vantagens:</u>	<u>Desvantagens:</u>
<ul style="list-style-type: none"> - A prensa mecânica de acionamento simples, é mais rápida que a hidráulica - É de longe a mais recomendada para operações que tenham processos de corte. - Não requer um motor tão grande como as prensas hidráulicas - Podem ser facilmente adaptadas a rolos, alimentação por transfer e a ferramentas progressivas - De uma forma geral as prensas mecânicas com cursos mais pequenos são mais económicas que as hidráulicas. 	<ul style="list-style-type: none"> - O ponto de pressão máxima encontra-se, por norma a 13 mm do ponto de fim de curso, isto significa que em pontos acima desse valor a pressão existente é bastante inferior ao que a prensa pode exercer. - Quando as alturas de estampagem são muito elevadas como a velocidade tende a diminuir até à chegada ao ponto morto inferior (PMI), a estampagem não ocorre a uma velocidade constante como seria desejável e acontece nas prensas hidráulicas.

2.4.5. Linhas de Prensas ('Press lines')

Regra geral, a produção de peças de tamanho médio e grande é realizado em linha de prensas, '*press lines*', ou em prensas transfer. Uma '*press line*', consiste numa série de prensas mecânicas ou hidráulicas dispostas de forma sequencial e interligadas através de sistemas de transporte de peças [4], tal como representado na Figura 16.

As '*press lines*' modernas devem garantir rigorosamente as exigências de cadência, qualidade da peça produzida, vida útil da ferramenta, tempo de operação e flexibilidade. Assim, como forma de garantir estes requisitos, são utilizadas prensas de acionamento mecânico com sistemas '*link drive*'. A utilização deste sistema permite reduzir a velocidade de impacto durante o fecho da ferramenta em cerca de um terço, quando comparada ao acionamento direto por cambota. Para uma mesma força de pressão nominal e força-deslocamento, uma prensa '*link drive*' permite uma maior abrangência de trabalho, pois, tal como as prensas hidráulicas, permite obter uma força superior a uma distância maior do ponto morto inferior (PMI), conforme representado na Figura 14, quando comparada com uma prensa de acionamento direto por cambota.[4]

Consequentemente, este tipo de prensas permite obter uma cadência mais elevada que uma prensa de acionamento direto por cambota, dependendo da altura de estampagem. Por norma quanto maior a altura de estampagem maior é o ganho de cadência. Se não existir um aumento de cadência, a diminuição da velocidade de impacto, resulta num melhoramento das condições de estampagem [4].

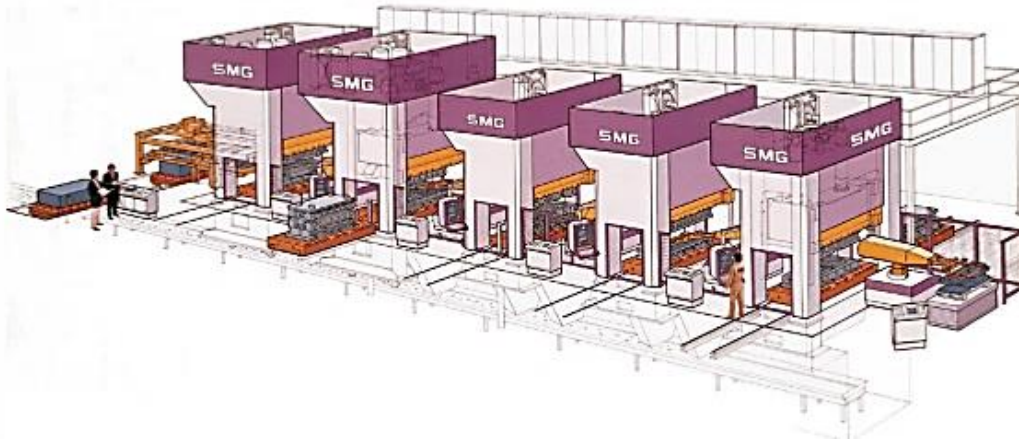


Figura 16 - Esquema de uma 'press line' automatizada com robôs (Retirado de [4]).

2.4.6. Automatização

A sequência de trabalho de uma 'Press line' inicia-se com o desempilhamento das chapas, seguido de uma lavagem e lubrificação, sendo seguidamente são introduzidas na primeira prensa. Nos dias de hoje, tanto a alimentação da chapa como o transporte entre prensas, são realizados de forma automática. Existem vários sistemas automatizados que podem ser utilizados, nomeadamente: alimentadores de balanço lateral '*swing arm feeders*', alimentadores CNC, ou Robôs [4].

Os '*swing arm feeders*' são acionados por motores elétricos através de cames e ligações, conforme ilustrado na Figura 17. O alimentador da saída da prensa de estampar, retira a peça estampada que se encontra elevada na ferramenta por meio de ejetores, através de ventosas e coloca-a na estação intermedia seguinte. Sempre que existe a necessidade de virar a peça, por forma a simplificar o processo, é utilizada uma estação intermedia [4].

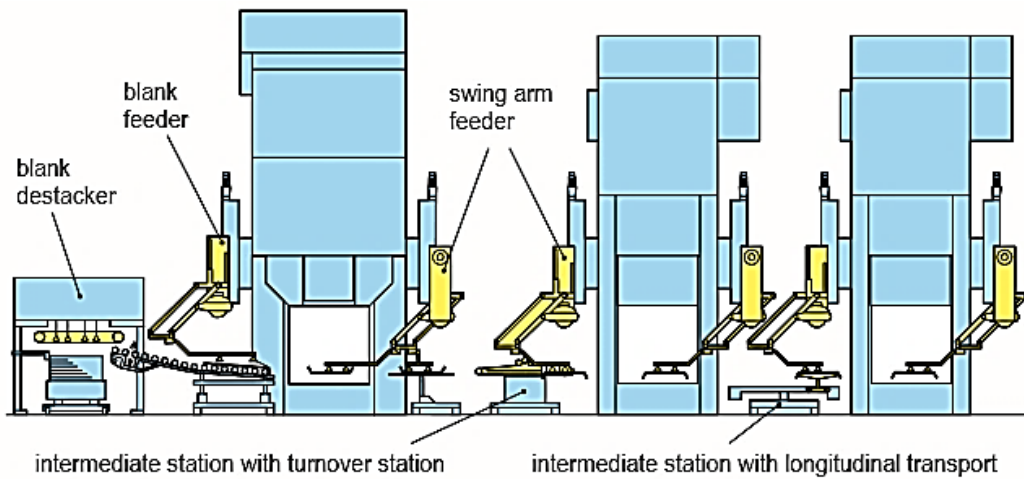


Figura 17 - Esquema de uma 'press line' com sistema de 'swing arm feeder' (Retirado de [4]).

O sistema 'swing arm feeders' têm curvas de movimento fixas, fazendo delas a solução ideal para trabalhos em que a distancia entre prensas seja pequena e o tamanho das peças a transportar de pequeno a médio [4].

Os alimentadores CNC são controlados eletricamente e como tal podem ser livremente programados em dois eixos e podem ser utilizados para uma grande variedade de tipo de peças, conforme ilustrado na Figura 18. Este tipo de alimentador permite retirar as peças de uma ferramenta para a próxima sem o recurso a ejetores [4].

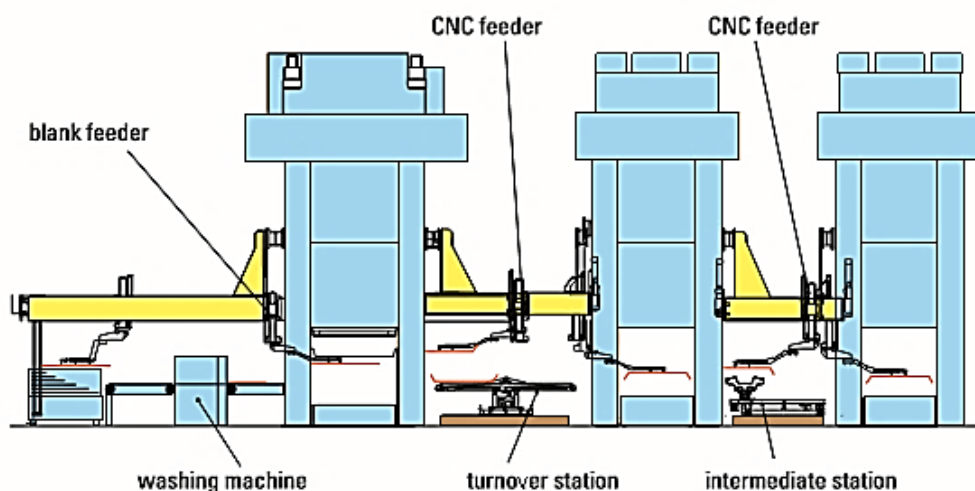


Figura 18 - Esquema de uma 'press line' com sistema de alimentação CNC (Retirado de [4]).

Quando se utiliza ambos os sistemas de alimentação *'swing Arm feeders'* e alimentadores CNC é necessário recorrer a estações intermédias, estando estas equipadas com programação de três a cinco eixos para o transporte das peças na direção longitudinal, ajustes na direção transversal, ajustes de altura e também algum movimento basculante adicional [4].

A automatização utilizando robôs elimina a necessidade das estações intermédias, tal que um robô é suficiente para servir uma prensa. Cada robô é capaz de transportar a peça diretamente de uma ferramenta para a ferramenta seguinte, tal como ilustrado na Figura 15. A principal desvantagem da utilização de robôs prende-se com o transporte de peças pesadas ser relativamente lento ou impossível devido às forças centrífugas e aos caminhos de transporte relativamente grandes [4].

2.4.7. Prensas Transfer

As prensas mecânicas transfer e prensas hidráulicas transfer são utilizadas para fabricar peça prontas para montagem na indústria automóvel. Estas prensas garantem uma cadencia elevada porque todas as operações necessárias para fabricar a peça estão incluídas numa única linha. Além disso quase todas as operações são realizadas quase em simultâneo, permitindo produzir uma peça a cada ciclo [4].

Independentemente do tipo de prensa transfer, mecânica ou hidráulica o transporte das peças de estação para estação é assegurado por intermédio de um sistema de calha tipo CNC, na indústria este sistema é conhecido como sistema transfer, que possui um conjunto de *'Grippers'*, nome dado na indústria ao equipamento estilo pinça, normalmente de atuação pneumática que fixa a peça durante o transporte entre estações.

Os sistemas de transfer bidimensionais, executam apenas os movimentos de abertura-fecho e avanço-recuo. Assim se existir a necessidade de elevar as peças durante o transporte esse movimento adicional é realizado por intermedio de um *'Gripper'* especial. O sistema tridimensional é utilizado sempre que o transporte das peças necessite de um movimento de elevação entre operações, sendo todo o sistema transfer munido de um movimento de elevação [4].

A escolha do tipo de prensa é definida essencialmente pelo tipo de peça a produzir, sendo que essencialmente a utilização da prensa hidráulica transfer é na maior parte dos casos utilizada para operações de estampagem invertida. Consoante a morfologia da peça o sistema transfer poderá ser bidimensional ou tridimensional, sendo que o sistema tridimensional é utilizado preferencialmente em peças de grandes dimensões ou que exijam um movimento de elevação para realizar o transporte entre operações [4].

O design de prensas transfer para peças grandes e médias depende do número de estações e da distribuição de forças. Nas prensas mais recentes o controlo do sistema de alimentação, remoção da peça e em alguns casos o empilhamento das peças concluídas assim como todo o controlo da prensa transfer é totalmente automatizado [4].

Uma prensa transfer pode ser alimentada por formatos ou por rolo, sendo que quando a alimentação se efetua por rolo existirá seguidamente uma estação para corte do formato sendo o restante processo de transformação idêntico ao de alimentação por formatos. O sistema transfer será responsável pelo transporte das peças entre estações, tal como representado na Figura 19. Considerando a alimentação por formatos, o processo produtivo pode ser descrito na sequência dos seguintes sete pontos:

1. Os formatos são colocados num dispensador de formatos;
2. Um sistema 'pick-and-place', sistema de alimentação do tipo CNC que pode ser programado em 2 ou 3 eixos, retira um formato da pilha de formatos e coloca-lo no sistema de transporte;
3. O sistema transfer pega no formato e coloca-o na primeira estação;
4. O cabeçote da prensa desce até ao seu ponto morto inferior, permitindo assim à ferramenta executar as várias operações existentes nas várias estações;
5. Durante a ascensão do cabeçote o sistema *transfer* realiza o transporte da peça presente na primeira operação para a segunda operação e assim sucessivamente;

6. Um sistema de posicionamento *'pick-and-place'* retira novamente um formato e coloca-o no sistema de transporte;
7. Volta ao ponto 3 e recomeça o processo.

Dependendo da geometria da peça o sistema transfer pode estar equipado com Grippers pneumáticos ou pás para garantir uma correta fixação das peças durante o transporte.

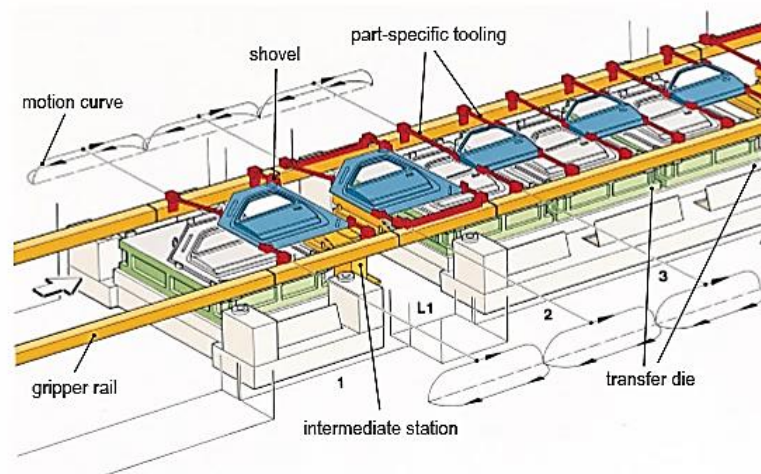


Figura 19 - Esquema do transporte de uma peça utilizando o sistema transfer tridimensional (Retirado de [4]).

É bastante comum serem utilizados sistemas transfer de acionamento elétrico, por permitirem programar livremente a trajetória, velocidade e aceleração. Esta capacidade de programação permite que este sistema seja ajustado a um maior número de ferramentas [4]. O respetivo diagrama do movimento do sistema transfer tridimensional é representado na Figura 19.

Analisando a Figura 20, pode-se concluir que o movimento do sistema transfer nos eixos xx (gráfico a verde), yy (gráfico a azul) e zz (gráfico a vermelho) são dependentes do valor do ângulo da cambota da prensa. Normalmente parâmetros como os cursos a percorrer nos 3 eixos assim como a que valor do ângulo da cambota da prensa o movimento se inicia e termina, são programados para cada ferramenta.

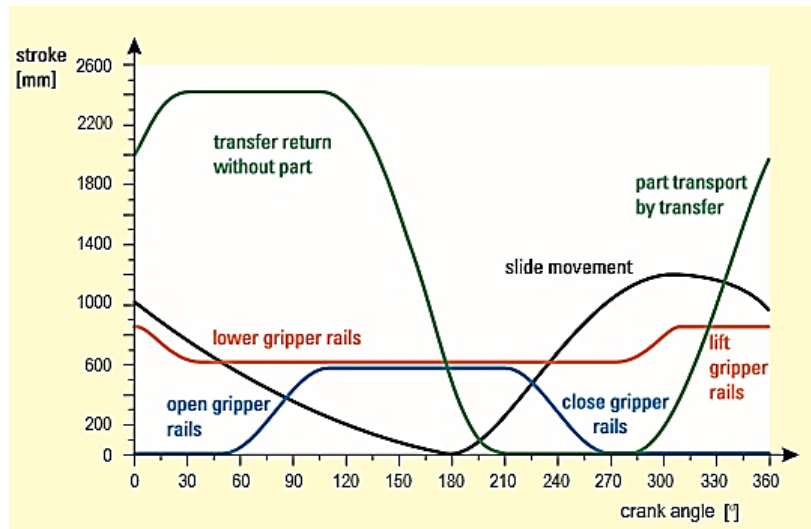


Figura 20 - Diagrama de movimento de um sistema transfer tridimensional (Retirado de [4]).

2.4.8. Prensas Transfer 'Crossbar'

Quando se pretende trabalhar formatos com um tamanho grande, cerca de 2.5 x 1.5 m o sistema de pega que fixa apenas as extremidades do sistema transfer tridimensional, tal como ilustrado na Figura 19, deixa de ser suficiente pelo que para peças com dimensões mais elevadas se utilizam prensas com sistema transfer 'crossbar', tal como representado na Figura 21.

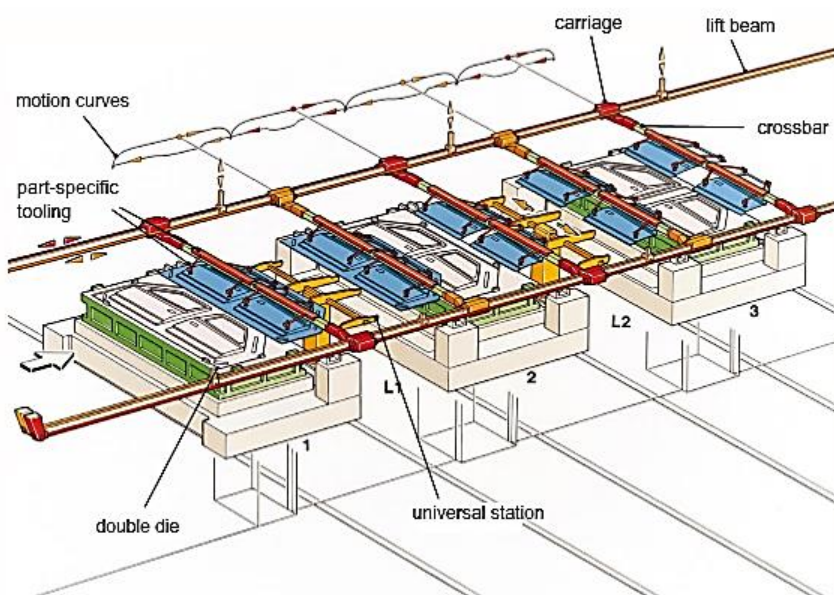


Figura 21 - Esquema de transporte de uma peça de elevadas dimensões numa prensa com sistema transfer 'crossbar' (Retirado de [4]).

A utilização deste tipo de solução permite também realizar a produção de uma peça esquerda e direita de um só formato ou formatos separados. Este tipo de produção dupla permite duplicar a cadência de produção ao mesmo que aumenta a economia do sistema quando comparado com o sistema transfer tridimensional [4].

Para realizar o transporte de peças instáveis e de tamanho considerável, é utilizado um sistema de transfer bidimensional com o sistema de 'crossbar' e são utilizadas ventosas como elemento de fixação das peças durante o transporte, conforme representado na Figura 22. Como a distribuição das ventosas é realizada de forma uniforme em torno do centro de massa das peças, são permitidas acelerações superiores quando comparado com outros sistemas de alimentação.

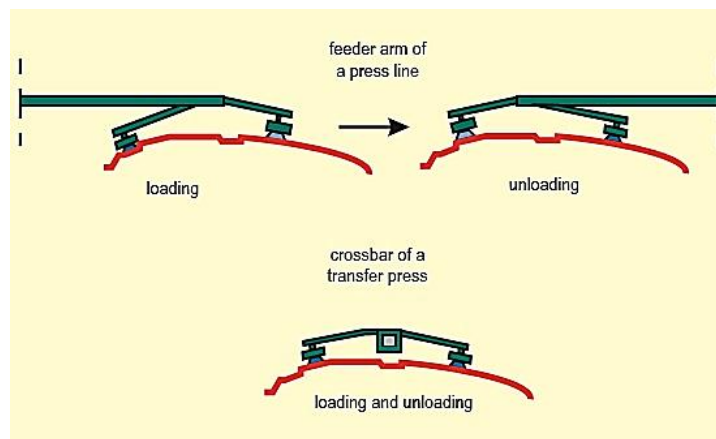


Figura 22 - Esquema do posicionamento das ventosas no sistema de transfer 'crossbar' (Retirado de [4]).

A estrutura da prensa é também diferente pois para cada estação existirá um teto individual, conforme ilustrado na Figura 23. Uma vez que existirá uma ferramenta que corresponde a uma estação por teto, os parâmetros como alturas e pressões necessárias podem ser ajustadas individualmente [4].



Figura 23 - Prensa com sistema transfer 'crossbar' – Prensa com 5 tetos individuais (Retirada de [4]).

Em algumas aplicações pode existir a necessidade de utilizar, tal como acontecia com as 'press lines', as estações intermédias ou universais, que podem ser programadas até cinco eixos, permitindo assim uma maior flexibilidade no transporte das peças. A utilização das estações universais permite diminuir a complexidade das estações e consequentemente diminuir o custo destas ao mesmo tempo que aumenta a sua fiabilidade. Além destas vantagens a utilização destas estações permite que o sistema transfer 'crossbar' assuma uma posição assimétrica de descanso durante o fecho da ferramenta. Este movimento, assim como o diagrama de movimento em função do tempo, que se diferencia bastante daquilo que foi apresentado para o sistema transfer tridimensional está descrito na Figura 24 [4].

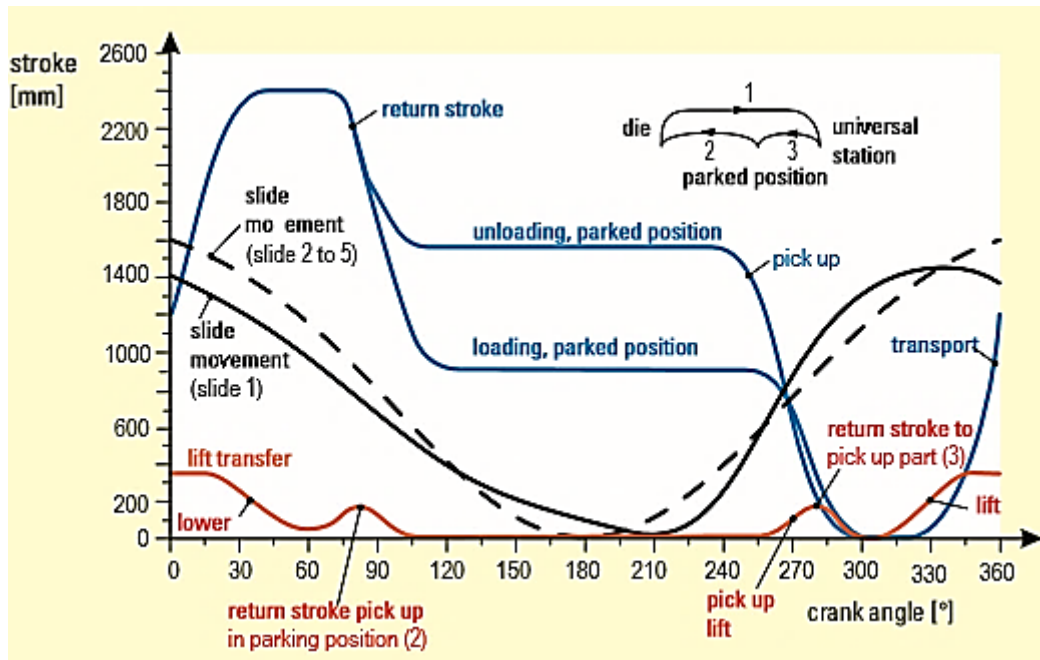


Figura 24 - Diagrama de movimento de um sistema transfer 'crossbar' (Retirado de [4]).

2.5. Simulação, validação e otimização do sistema transfer

Um processo de fabrico automatizado como é a produção de peças através de uma prensa com sistema transfer possui bastantes funções complexas dependentes do tempo. Todos os movimentos e interações críticas entre as peças em movimento devem ser sincronizadas de forma a evitar colisões e garantir uma maior cadência. Este processo é realizado previamente à entrada em produção e designa-se usualmente por validação cinemática virtual.

Quando se utilizam prensas transfer é usual realizar-se a deteção de eventuais colisões entre o sistema transfer e a ferramenta, numa zona da linha de produção dedicada a realizar simulações cinemáticas, conforme ilustrado na Figura 25, onde se pretende não só ajustar os parâmetros do transfer, mas também garantir o correto posicionamento das peças nas estações universais, no caso do sistema transfer 'crossbar'.

Os movimentos do sistema transfer, peça e ferramenta devem estar sincronizados em relação de uns aos outros e conseqüentemente não deve existir qualquer colisão entre o sistema transfer, a peça e a ferramenta. Nos sistemas

tridimensionais considera-se também uma distância de segurança de 20 mm, que deve ser garantida entre a ferramenta e qualquer elemento móvel do sistema transfer e peça transportada [4]. Garantindo o cumprimento destes requisitos ao nível da simulação, o sistema considera-se validado cinematicamente e a informação obtida durante a simulação é compilada num programa que é posteriormente comunicado à linha de produção.

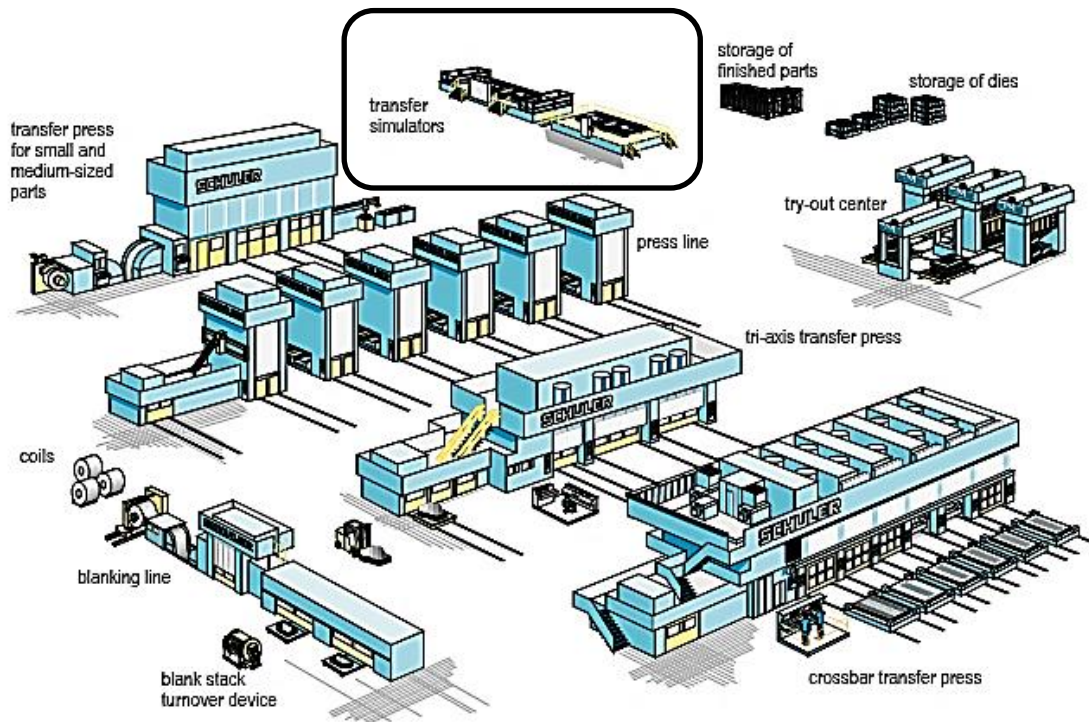


Figura 25 - Esquema de uma fábrica para a produção de painéis de grande dimensão, incluindo uma zona de simulação cinemática (Retirado de [4]).

O processo de simulação para validação cinemática virtual envolve a definição de um elevado número de parâmetros associados a todas as funções dependentes do tempo relativas ao fabrico automatizado. Neste enquadramento, considerando que os processos industriais de fabrico de peças estampadas em linhas de prensas, são demasiado complexas para poderem ser modeladas analiticamente, a otimização baseada em simulação é uma ferramenta poderosa para determinar o valor ótimo dos parâmetros, tal como destacado por April *et al* [8].

Uma linha de prensas, é formada por um conjunto de prensas em série, em que cada uma possui um alimentador e extrator de peça. Normalmente são agrupadas

quatro a seis prensas. O controlo de cada estação é garantido através de sistemas de controlo industrial, que gerem todos os movimentos, feedback contínuos, eventos discretos e controlo de segurança e supervisão. Por norma a componente associada aos eventos discretos é separada da gestão do movimento utilizando sistemas lógicos programáveis (SLP). Goor [9], demonstra um aumento da ocupação de uma linha de prensas real, através de uma alteração na estratégia de controlo. Uma coordenação entre dispositivos, por exemplo, alimentador, extrator e a prensa, foi implementada com o cálculo da previsão das posições futuras. Lin e Kuo [10], por seu lado desenvolveram um sistema integrado de CAD/CAE/CAM para uma linha de prensas da indústria automóvel, na qual se focam no benefício da redução do tempo de produção e custo de uma ferramenta.

Emel e Tasci [11], apresentam uma abordagem de otimização do agendamento temporal para um dado número de tarefas numa linha de prensas utilizando um modelo de Programação Inteira Misto. Estes autores consideram cada tarefa da linha de prensas como um elemento, onde a distância percorrida é definida em unidades de tempo, representando a conclusão de um lote. O objetivo é maximizar a capacidade de utilização da linha de prensas utilizando ferramentas matemáticas em janelas temporais deslizantes. Contudo, no cálculo do tempo requerido para completar um lote, o tempo para processar cada chapa de metal é definido como constante, não destinado a ser otimizado. Com esta abordagem, todos os parâmetros de controlo das prensas, como por exemplo velocidades e trajetórias, são considerados à partida como ótimos, sem que seja contabilizada a influência destes parâmetros no processo de produção.

Ozalp [12], continua a abordagem de Emel e Tasci, mas com recuso uma técnica baseada em algoritmos genéticos. Kuk *et al.* [13], construíram uma fábrica virtual para facilitar o desenvolvimento de produtos e agilizar a produção de uma empresa automóvel coreana.

Hoffman e Kohnhäuser [14], propõem uma estratégia para otimizar o transporte das peças numa prensa com sistema transfer 'crossbar'. Eles desenvolveram um algoritmo que permite o cálculo automático do melhor

posicionamento das ventosas, para uma determina peça, de forma a diminuir as forças de fixação envolvidas.

Garcia-Sedano *et al.* [15], descrevem uma abordagem utilizando o ambiente de simulação virtual, para automatizar o calculo de caminhos ótimos de robôs transportadores de peças em linhas de prensas. O método é baseado em algoritmos genéticos em conexão com o simulador virtual da linha, permitindo assim simular a execução de processos sobre condições quase reais.

Svensson *et al.* [16], propõem a combinação de dois algoritmos, combinando algoritmos de busca global com busca local. Utilizando uma combinação dos algoritmos Lipschitzian e Simplex, ao qual dão o nome de CoLiS. O algoritmo CoLiS, inicia-se com a utilização do algoritmo de Lipschitzian, realizando uma busca global trocando posteriormente para o algoritmo Simplex de Nelder-Mead, para realizar a busca local. Assim segundo Svensson *et al.* consegue-se utilizar o melhor de ambos os algoritmos, sem as desvantagens associadas aos mesmos.

2.6. Otimização com Algoritmos Genéticos

Considerando o estado da arte apresentado na secção anterior em relação à validação cinemática das prensas com sistema transfer, neste trabalho é explorada a utilização da otimização com algoritmos genéticos na pesquisa dos parâmetros ótimos relativos ao fabrico automatizado.

A teoria dos Algoritmos Genéticos foi inicialmente desenvolvida por John Holland em 1960 [17]. Os Algoritmos Genéticos (GAs) são métodos de procura estocásticos, que assentam o seu conceito na teoria da evolução das espécies de Darwin e na transmissão de informação genética entre gerações, a genética Mendeliana [18,19]. Contrariamente a outros tipos de algoritmos de busca que se limitam a utilizar apenas um ponto (ou hipótese) no espaço de possíveis soluções, os GAs iniciam a sua busca com um conjunto de hipóteses [20].

Uma vez que os GAs tentam recriar os processos biológicos de reprodução e seleção natural, muita da terminologia utilizada é similar à utilizada na biologia. Assumindo que uma potencial solução de um problema pode ser representada por um

conjunto de parâmetros (por exemplo as dimensões de um equipamento num problema de otimização do seu peso), estes parâmetros na terminologia GA são definidos como Genes. Um conjunto de Genes será chamado de Cromossoma [21]. Assim, um cromossoma será uma solução válida, constituída por um conjunto de parâmetros (Genes). Por sua vez, um conjunto de Cromossomas será definido como uma população.

Na natureza, os indivíduos de uma população competem ou com os outros por recursos como comida, água, abrigo e com bastante frequência para atrair uma companheira. Assim, é espectável que os indivíduos que têm mais sucesso na sobrevivência e na atração de uma companheira, tenham um maior número de descendentes. Os indivíduos, por outro lado, que não consigam obter sucesso podem nem chegar a obter descendência. Isto significa que os Genes dos indivíduos mais adaptados ou *'fit'* irão propagar-se a um elevado número de descendentes em cada geração [21].

O cromossoma irá evoluir através de sucessivas iterações chamadas gerações. Em cada geração os cromossomas são avaliados, sendo utilizada uma medida de aptidão [22].

Um Algoritmo genético é, portanto, um algoritmo cuja implementação considera os seguintes requisitos [20]:

1. Uma população de soluções para o problema
2. Uma forma de calcular se cada solução na população é boa ou má
3. Um método para misturar os fragmentos das melhores soluções de forma a criar novas.
4. Um operador de mutação para evitar a perda permanente de diversidade nas soluções.

Pode-se definir também que um algoritmo genético utiliza tipicamente os seguintes operadores genéticos: Seleção, Crossover e Mutação [20].

Para garantir que as soluções que contêm os melhores genes são escolhidas, preferencialmente, para criar descendência é utilizada um método de comparação de cromossomas, a função 'fitness'.

2.6.1. Função objetivo

A função objetivo ou função 'fitness' é a função que se pretende que o algoritmo otimize, maximizando ou minimizando. A função 'fitness' testa e quantifica o quanto 'fit' ou apta uma dada solução é [23].

A função 'fitness' é então responsável por definir o critério para que as potenciais soluções contidas na população sejam classificadas de acordo com a sua aptidão [24]. É com base na classificação que os cromossomas serão selecionados para a criação de descendentes.

2.6.2. Operadores Genéticos

2.6.2.1. Seleção

Uma vez realizada a classificação da população de acordo com a sua aptidão, é realizado o processo de seleção. Existem vários métodos de seleção sendo os mais conhecidos: a seleção por Roleta ou '*Roulette wheel selection (RWS)*', amostragem universal estocástica ou '*Stochastic universal sampling (SUS)*', seleção por ranking linear ou '*Linear rank selection (LRS)*', seleção por ranking exponencial ou '*Exponential rank selection (ERS)*', seleção por torneio ou '*Tournament selection (TOS)*' e seleção por truncatura ou '*Truncation selection (TRS)*' [18].

O método de seleção deve ser ajustado consoante o problema a otimizar, porém não existe nenhuma diretriz que defina em que tipo de situação de deve optar por um método específico em detrimento dos restantes [18]. O tipo de método utilizado para a seleção irá influenciar a forma como o algoritmo converge para a solução ótima.

Neste TFM, o método de seleção utilizado foi o de seleção por Roleta por ser um dos mais utilizados, independentemente do risco de uma convergência prematura associado à utilização deste método. Este método de seleção atribui, a cada elemento

da população i de uma população de tamanho N , uma probabilidade $p(i)$ de ser selecionado que é proporcional ao valor da sua função 'fitness' $f(i)$, dada por:

$$p(i) = \frac{f(i)}{\sum_{j=1}^N f(i)} \quad (2)$$

Um código genérico para a implementação deste método é apresentado na Figura 26, onde se pode verificar a existência de um ciclo de programação que percorre cada elemento da população.

- Calculate the sum $S = \sum_{i=1}^n f(i)$;
- For each individual $1 \leq i \leq n$ do {
 - Generate a random number $\alpha \in [0, S]$;
 - $iSum = 0$; $j = 0$;
 - Do {
 - $iSum \leftarrow iSum + f(j)$;
 - $j \leftarrow j + 1$;
 - } while ($iSum < \alpha$ and $j < n$)
 - Select the individual j ;

Figura 26 - Código genérico de implementação do método de seleção por roleta (Retirado de [18]).

2.6.2.3. Crossover

O crossover é o principal operador genético. Este processo pega em dois cromossomas selecionados anteriormente, pelo método de seleção, e mistura a informação dos dois cromossomas criando dois novos descendentes, tal como representado na Figura 27. Usualmente o crossover é realizado em apenas um ponto, denotado por ponto de crossover, mas podem ser utilizados dois ou mais pontos [22,24].

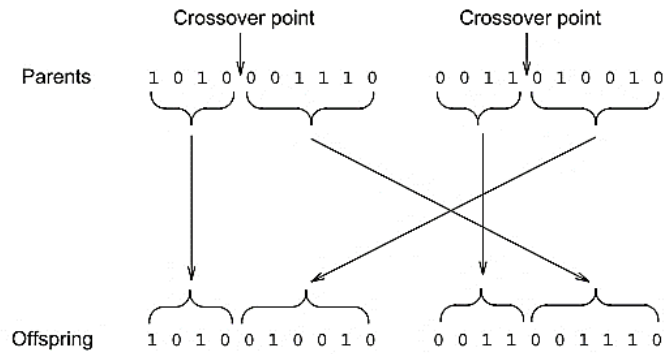


Figura 27 - Esquema da utilização do crossover com um ponto de crossover (Retirado de [21]).

O ponto em que o crossover é realizado pode ser fixo, porém pode existir a necessidade deste poder variar pelo que quando tal acontece a sua posição é definida pela criação de um número aleatório.

Outro conceito a reter é o da taxa de crossover, que define a taxa com que o crossover acontece. Em cada geração antes de realizar o processo de crossover é criado um número aleatório entre 0 e 1 e se esse número foi inferior à taxa de crossover o crossover é realizado. Quando não existe crossover a informação dos cromossomas selecionados é diretamente copiada para os descendentes, garantindo assim que a informação genética é passada aos descendentes em que exista a perturbação do crossover [20,25]. Uma taxa de crossover mais elevada permite que o espaço de soluções seja mais explorado evitando a convergência das soluções para um falso ótimo, porém se este valor foi demasiado elevado irá resultar num desperdício de recursos computacionais [22]. Neste TFM será utilizado um ponto de crossover com posição variável para que não ocorra a convergência prematura do algoritmo.

2.6.2.4. Mutação

A mutação é um processo genético que introduz alterações aleatórias. É aplicada individualmente a todos os descendentes depois do crossover. A taxa de mutação, normalmente valor muito baixo (0.01), define se o gene em questão sofre mutação [21].

Tal como acontecia no crossover o gene onde ocorre a mutação pode ser fixo, ou pode ser criado um número aleatório que define a posição do gene a mutar. Na Figura 28 é ilustrada a utilização da mutação.

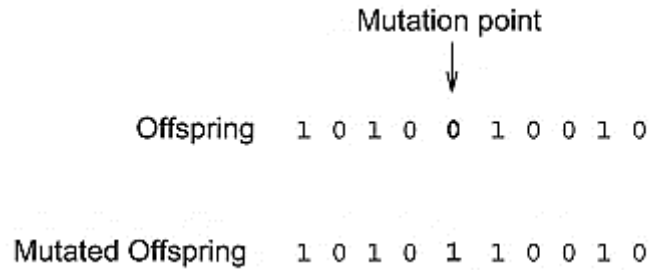


Figura 28 – Esquema da utilização da mutação (Retirado de [19]).

A mutação tem a função crucial de: (a) substituir genes perdidos pela população durante o processo de seleção ou (b) providenciar os genes que não estavam presentes na população inicial [20].

2.6.2.5. Elitismo

A seleção proporcional ao fitness, não garante a seleção de nenhum indivíduo em particular, apenas a do mais apto. Com este tipo de seleção a melhor solução pode muitas vezes ser excluída. Para evitar esse acontecimento muitas vezes é utilizado o elitismo que garante que o elemento mais apto passa sempre para a nova geração sem ser sujeito aos processos de crossover e mutação [18].

Capítulo 3 – Modelação de uma ferramenta transfer para validação cinemática virtual

3.1. Projeto de uma ferramenta transfer

O projeto de uma ferramenta inicia-se com o estudo do processo de estampagem, recorrendo a *softwares* de simulação específicos para o efeito, sendo esta fase uma das mais importantes no projeto de uma ferramenta. Através destes *softwares*, o estudo de enformabilidade é realizado através da simulação de várias geometrias de forma a que o processo de estampagem se torne o mais estável possível.

Seguidamente é realizada a construção do ‘*strip-layout*’, para tal é necessário definir-se o passo e a distância entre operações. É com base no ‘*strip-layout*’ que todo o projeto da ferramenta nasce, assim sendo é a partir deste que são obtidas as geometrias para as matrizes, punções, pisas e cerras. Sendo que para tal é também necessário saber quais as forças de corte, forças de estampagem e respetivas forças de extração.

3.1.1. Construção do ‘Strip-layout’

Considere-se, para efeitos demonstrativos e explicativos que se irá realizar o desenvolvimento do ‘*strip layout*’ que permita a produção da peça que se encontra na Figura 29, com recurso ao software Ds CATIA V5.

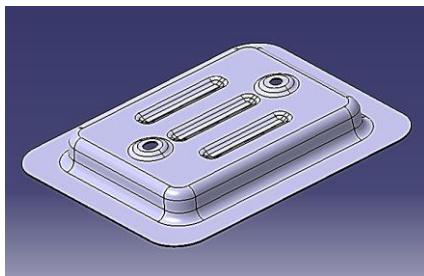


Figura 29- Peça em estudo

Como primeira fase do projeto desenha-se o planificado da peça. É a partir deste que nasce todo o ‘*strip-layout*’, assim como o projeto da ferramenta.

Assim, o ficheiro STEP com a peça a desenvolver foi importado no *software CATIA V5*. Posteriormente, utilizando o módulo de superfícies, foi utilizada a função *"Extract"* para extrair a superfície da peça em estudo, conforme a Figura 30.

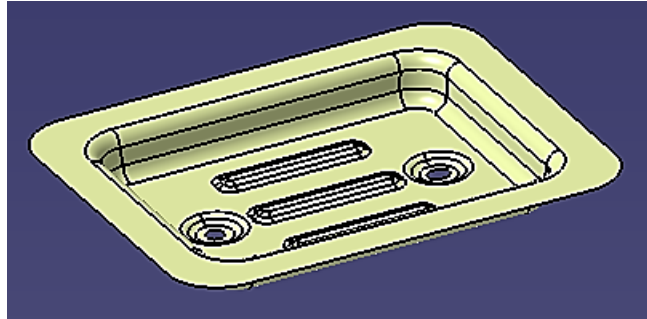


Figura 30 - Superfície obtida utilizando a função *"Extract"*

Posteriormente, foi utilizada a função *"Fill"* para preencher os furos existentes na peça, sendo posteriormente utilizada a função *"Join"* para, como o nome sugere, juntar as superfícies criadas. Após esta etapa foi utilizada a função *"Offset"* para criar uma superfície idêntica, mas a distância fixa da superfície original, conforme a Figura 31, sendo que nesta distância foi considerado metade do valor da espessura h , ficando assim esta nova superfície à distância $h/2$ da linha média.

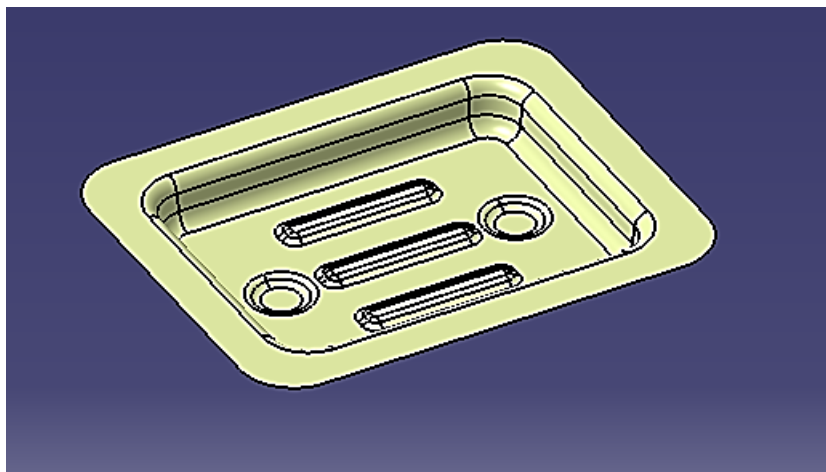


Figura 31- Superfície da peça, na linha média

Seguidamente e tendo como base a superfície ilustrada na Figura30, foi utilizada a função *"Unfold"* para planificar a superfície, conforme a Figura 32.

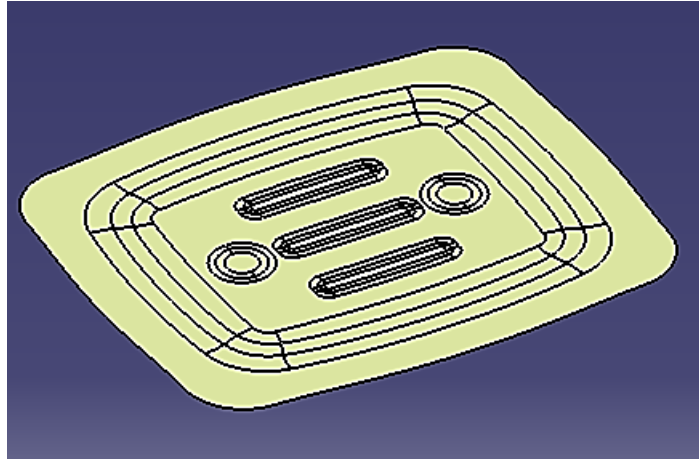


Figura 32 -Planificado da peça a desenvolver

Com a superfície ilustrada na Figura 31, foi utilizada a função “Boundary” para criar a linha que delimita a superfície. Para a construção do planificado final, conforme representado na Figura 33, teve-se como base a superfície da Figura 32 e foi considerado alguma matéria excedente para que possam ser executadas todas as operações sem comprometer a peça final.

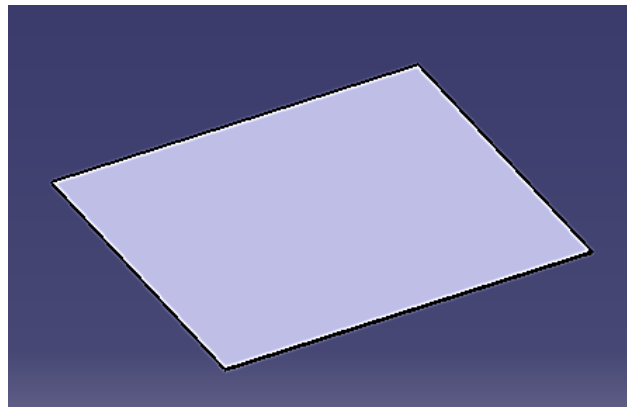
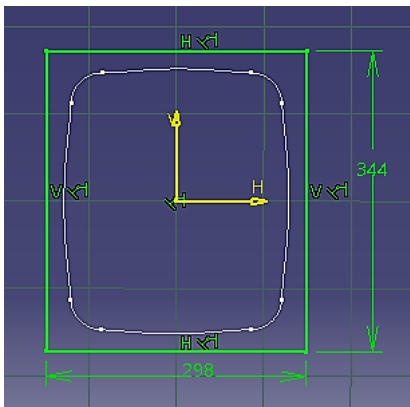


Figura 33 - Planificado Final

Após a definição do planificado final foi elaborado o número de passos necessários para transformar o planificado, na peça final (ver Figura 29), sendo que foi considerado um passo, para a distância entre operações, de 450 mm.

Assim foi definido que as diferentes fases de transformação do componente, serão as seguintes:

1. O corte do planificado;
2. Operação de estampagem;
3. Corte dos furos assim como o corte parcial do excedente das abas;
4. Corte final de forma a obter a peça final.

Na Figura 34 são representadas as várias fases de transformação do componente, obtendo assim o respetivo 'strip-layout'.

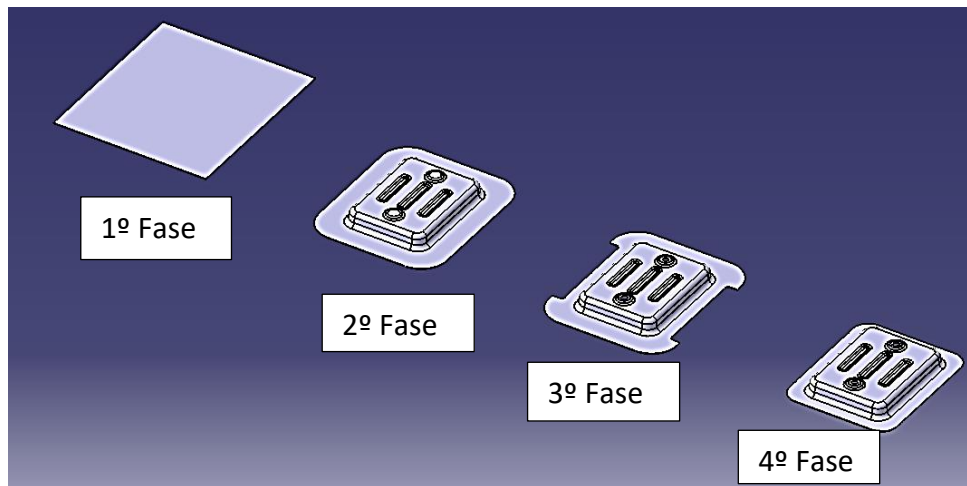
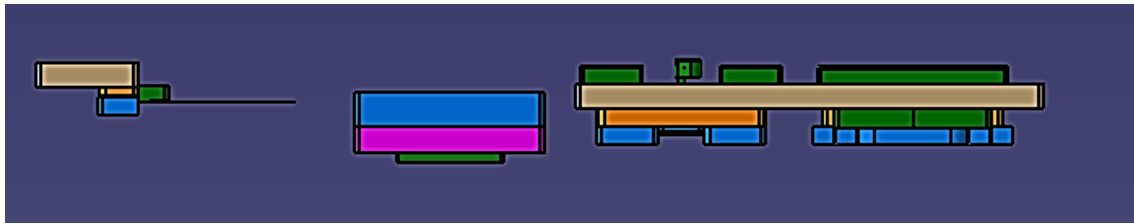


Figura 34- Strip-layout do processo produtivo e fases de transformação do componente

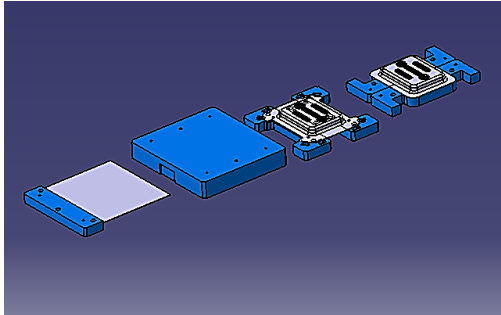
3.1.2 Construção da ferramenta

Tal como dito anteriormente, após a definição do strip layout são realizados cálculos para as forças de estampagem, corte e extração.

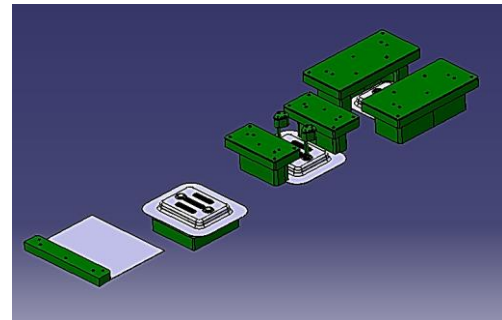
Depois de realizados estes cálculos são desenhadas todas as geometrias que entram em contato direto com a chapa tais como matrizes, de corte e de estampagem, punções de corte e de estampagem, pisas e cerras. Na figura 35 é apresentado um exemplo de vários elementos de uma ferramenta.



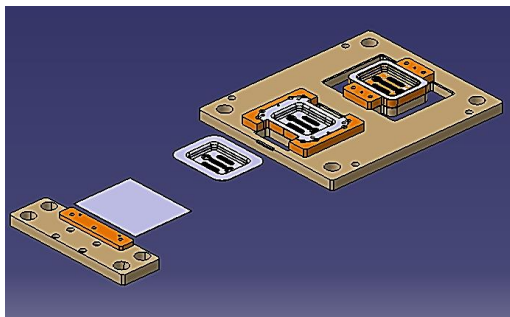
(a)



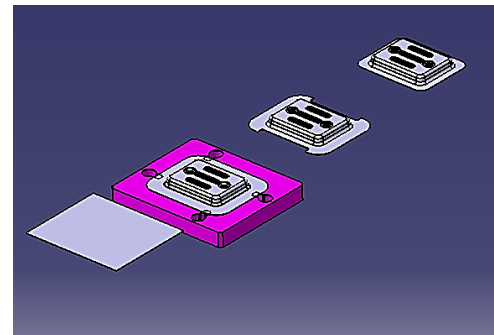
(b) Matrizes



(c) Punções



(d) Pisas



(e) Cerras

Figura 35- Representação dos vários elementos da ferramenta. (a) Esquema de montagem dos vários elementos de uma ferramenta (b) Matrizes (c) Punções e porta Punções (d) Pisas e porta pisas (e) Cerras

Após o desenho mecânico dos componentes da ferramenta, é esboçada toda a estrutura envolvente da ferramenta, finalizando assim o projeto.

Apesar do caso de estudo apresentado para demonstrar e explicar o processo de desenvolvimento de um 'strip layout' e consequente construção da ferramenta ser um caso simples, estes processos são bastantes complexos sendo que dependendo do tamanho da peça o 'strip layout' poderá ter mais de 8 operações e a ferramenta poderá ter facilmente mais de 2000 componentes.

3.2 Simulação cinemática de uma ferramenta transfer

Uma vez concluído o projeto da ferramenta é vantajoso realizar uma análise cinemática à ferramenta de forma a poder detetar eventuais colisões entre componentes e zonas que poderão ser problemáticas. Para o projeto realizado no software Ds CATIA V5, são exploradas duas soluções de modelação para simulação cinemática, com recurso ao módulo adicional Digital mock-ups (DMU) Kinematics e à versão base do software sem módulos adicionais.

3.2.1 Modelação com Ds CATIA V5 – Módulo DMU Kinematics

O módulo de simulação DMU Kinematics oferece um conjunto de ferramentas especialmente desenvolvido para a simulação cinemática. O menu principal deste módulo podemos dividido em três grandes grupos, conforme ilustrado na Figura 36.



Figura 36- Menu principal CATIA V5 DMU Kinematics

Considerando os grupos de comandos na vertical e da esquerda para a direita na Figura 36, encontram-se nesses grupos:

1. Menus de simulação, tanto por comandos manuais como comandos ativados através de leis do movimento.
2. Menus de seleção de uniões, fixação e sistemas de aceleração.
3. Menus que permitem a realização da simulação, a captura da simulação e criação de vídeo assim como uma ferramenta que permite obter informação de possíveis colisões e uma ferramenta que permite extrair o volume ocupado pelos componentes durante a simulação.

Como forma de iniciar uma simulação é necessário que os modelos geométricos CAD dos sistemas a simular sejam importados e seguidamente sejam criadas juntas cinemáticas.

Neste módulo de simulação cinemática, existem várias juntas cinemáticas, tal como demonstrado na Figura 37, porém para a simulação de uma ferramenta transfer são utilizadas essencialmente uma combinação de juntas do tipo cilíndricas, prismáticas, rígidas, e 'Point curve' (em que um ponto segue uma curva paramétrica tridimensional).



Figura 37 - Menu de juntas cinemáticas do Ds CATIA DMU Kinematics

Ao importar os modelos geométricos CAD da base e o topo da prensa, é considerada uma junta cilíndrica, restringindo o curso e a rotação, entre as duas peças, assim como também é considerada a base da prensa como um elemento fixo, como representado na Figura 38.

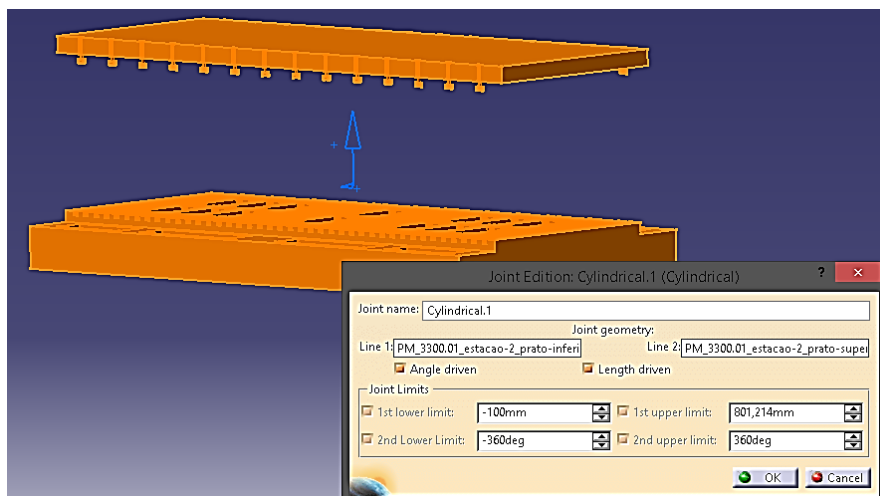


Figura 38 - Representação da junta cilíndrica entre base e topo da prensa

Seguidamente são importadas, a base e o topo da ferramenta, sem os elementos móveis, sendo considerada uma junta do tipo rígida entre o cabeçote da prensa e o topo da ferramenta e outra entre a base da ferramenta e a base da prensa, conforme representado na Figura 39.

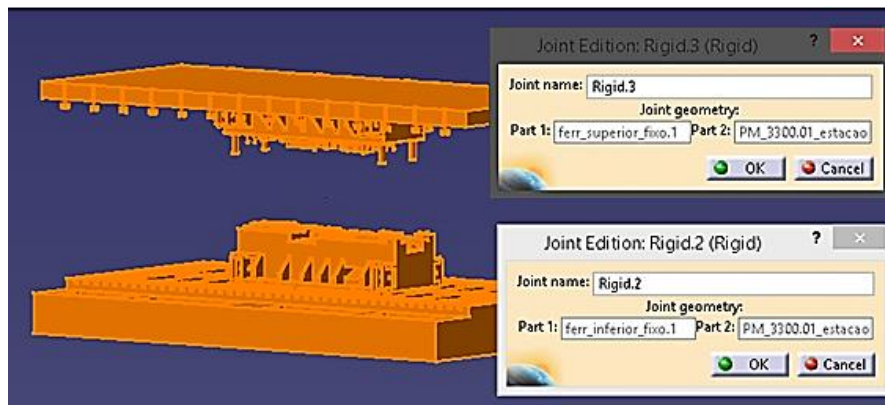


Figura 39- Representação da junta rígida entre a ferramenta e a prensa

No caso dos pisas, dos cerras e dos elevadores das operações foi definida uma junta do tipo cilíndrica entre cada um destes e a ferramenta, restringindo assim a sua rotação e o seu curso.

Para a definição do sistema transfer foi utilizado um sistema auxiliar para que fosse possível restringir o movimento do sistema *transfer* em cada um dos 3 eixos. Neste sistema auxiliar foram consideradas três juntas do tipo 'Point curve', uma para cada eixo, sendo que este tipo de junta cinemática, necessita que se defina um ponto num corpo e uma linha ou curva pela qual o ponto se irá deslocar, existiu a necessidade de remover uma possível rotação com a utilização de uma junta do tipo prismática.

Tendo como base este mesmo raciocínio foi criado um sistema auxiliar, que permitisse realizar os movimentos relativos das peças nas várias operações, tendo-se considerado o mesmo tipo de juntas cinemáticas.

Após a realização destes passos já é possível a simulação cinemática do conjunto, sendo possível controlar o valor do deslocamento das juntas cinemáticas manualmente e de forma independente. No entanto este módulo de simulação permite que a simulação seja automatizada, sendo para isso necessário associar a cada junta cinemática que se pretenda automatizar um desenho paramétrico das curvas cinemáticas a partir do qual o sistema irá ler os valores.

Assim foram criadas as várias curvas cinemáticas, uma para cada junta a controlar, sendo a Figura 40 um exemplo da curva cinemática da prensa.

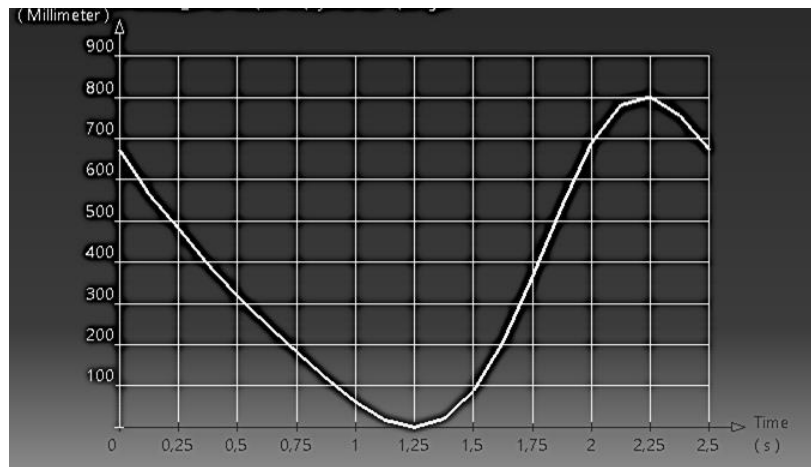


Figura 40- Curva cinemática da prensa, deslocamento vs. tempo

A curva cinemática considerada para o sistema transfer foi desenvolvida para que os ângulos que controlam o início e fim do movimento possam assumir vários valores, conforme se apresenta na Figura 41.

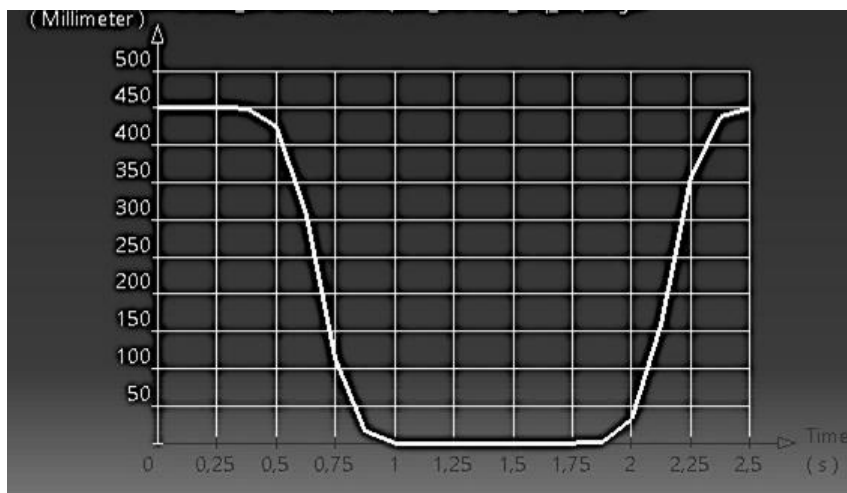


Figura 41 - Deslocamento do transfer segundo o eixo xx, deslocamento vs. tempo.

Com a utilização deste recurso do módulo de simulação já é possível realizar o estudo cinemático de uma ferramenta transfer, porém, sempre que exista a necessidade de ajustar os ângulos de início ou fim de um dos eixos esse ajuste terá que ser realizado de forma manual. Além disso, todo o sistema de modelação da simulação

acaba por ser um processo que consome bastante tempo, podendo chegar a cerca de 8 horas de trabalho para que se consiga obter um caminho livre de colisões, sem qualquer garantia que o mesmo seja ótimo. Aliado a estes fatores existe ainda o facto do módulo DMU Kinematics ser bastante caro (cerca de 20 000 €/ano sensivelmente, dados 2018).

3.2.2 Modelação com Ds CATIA V5 – Versão Base

Tendo por base exclusivamente a questão do custo, existe a necessidade de se explorar outras alternativas, sendo que uma delas passa por realizar a simulação cinemática utilizando um código de Algoritmo genético a uma licença base do software Ds CATIA V5. Isto porque analisando ao pormenor uma junta do tipo cilíndrica do módulo DMU Kinematics pode perfeitamente ser substituída por um constrangimento do tipo ‘offset’ da versão base do Ds CATIA, que garante uma distância fixa entre corpos, continuando a utilizar as curvas cinemáticas para a definição do valor destes ‘offsets’ de distância.

É com base neste princípio que é desenvolvida uma metodologia que permite a validação cinemática virtual numa licença base de DS CATIA V5, assim como o desenvolvimento de um algoritmo de otimização baseado em algoritmos genéticos que garanta um caminho realizado pelo sistema transfer sem colisões e com a cadência máxima. Para este efeito foi desenvolvido um modelo CAD mais simplificado da ferramenta transfer, por forma a disponibilizar mais recursos computacionais para as simulações. Neste modelo foi considerada uma parte superior da ferramenta, parte inferior da ferramenta, o sistema transfer e elementos moveis da ferramenta.

Para a parte superior considerou-se um bloco de dimensões 2500x2000x500 [mm] segundo xx, yy e zz, respetivamente, para a parte inferior considerou-se um bloco de dimensões 2500x2000x300 [mm] segundo xx, yy e zz, respetivamente, encontrando-se a representação destes blocos na Figura 42.

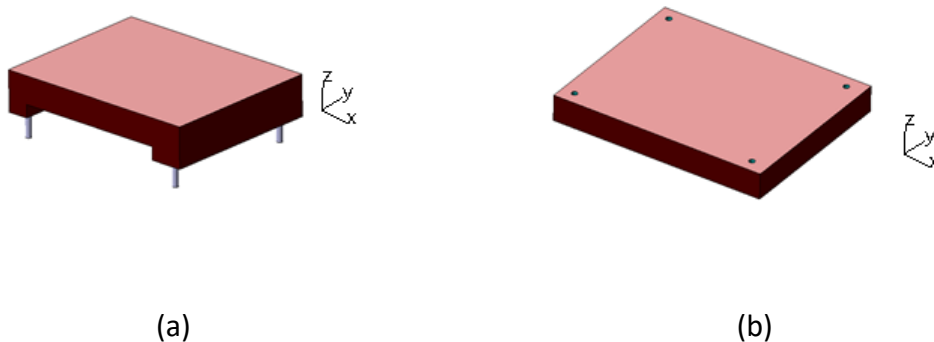


Figura 42 – Modelo CAD da ferramenta. (a) parte superior da ferramenta. (b) parte inferior da ferramenta.

Considerou-se também um sistema transfer triaxial, conforme representado na Figura 43(a), de dimensões 2000x100x282 [mm] segundo xx, yy e zz respetivamente, e elementos móveis da ferramenta onde neste caso foram considerados elementos como: o pisa, Figura 43(b), cuja função é garantir que a chapa é pisada durante a estampagem garantindo um melhor escoamento de material; um elevador, Figura 43(d), que eleva a peça para que o sistema transfer possa pegar a peça sem entrar em contacto com a ferramenta; e por fim um cerra ou encostador, Figura 43(c), cuja função é limitar o escoamento do material durante a estampagem.

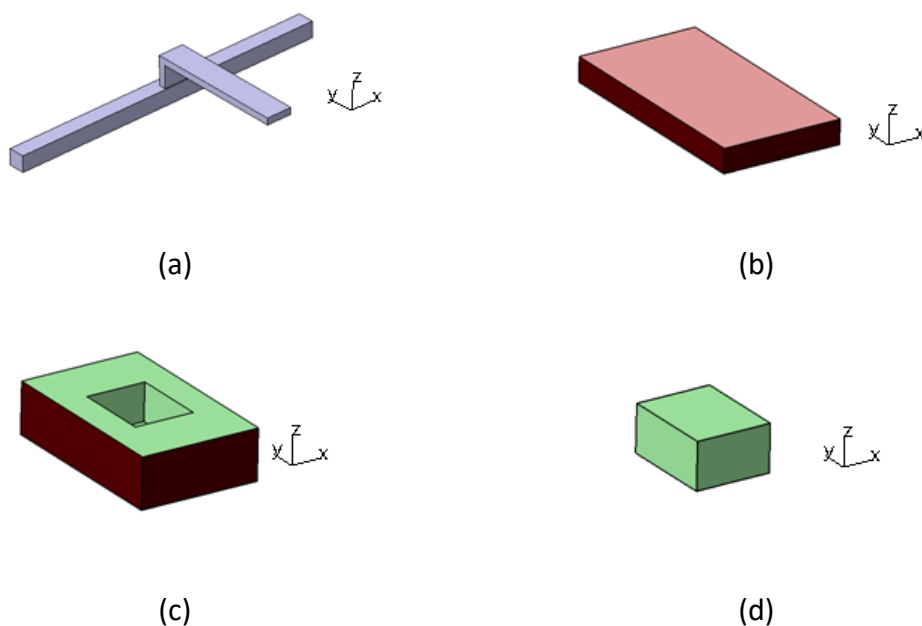


Figura 43 - Modelo CAD dos componentes da ferramenta. (a) – Sistema transfer triaxial. (b) - Pisa. (c) - Cerra. (d) - Elevador.

Neste caso optou-se por não se proceder à modelação de punção e matriz, pois estes são os últimos elementos a entrar em contacto com a chapa e apenas quando a ferramenta já se encontra praticamente fechada.

Na Figura 44, encontra-se representado a montagem de todos os componentes do modelo CAD simplificado utilizado neste TFM.

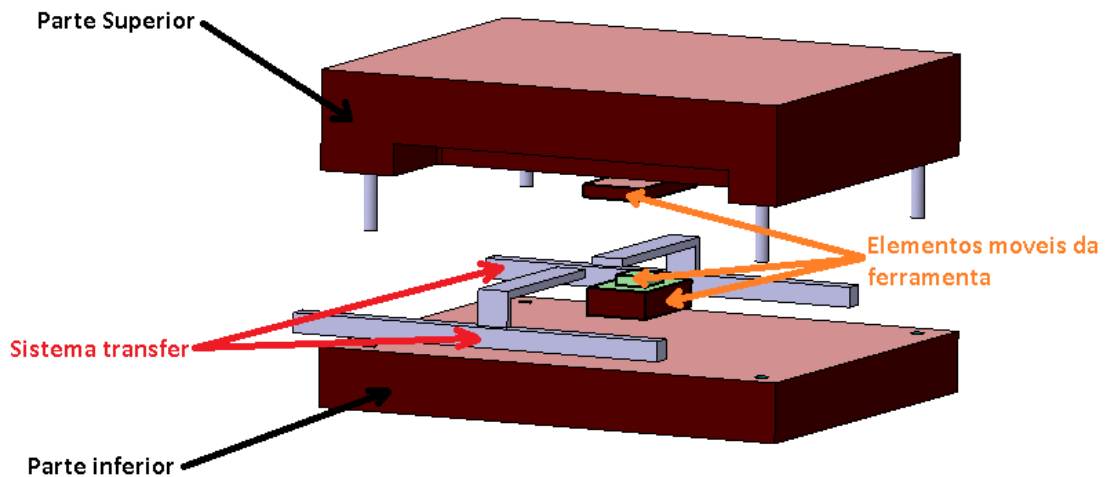


Figura 44 – Modelo CAD utilizado para simular a ferramenta e o sistema transfer

Na figura 44 não se representa a prensa, pois considera-se que o topo da parte superior se encontra acoplada à parte móvel da prensa e a base da parte inferior acoplada à cama da prensa, tendo sido por esse motivo excluída do modelo, porém a nível cinemático considerou-se que a parte superior, uma vez que se encontra acoplada à parte móvel da prensa terá o mesmo movimento e, portando, o seu movimento será descrito por uma equação do tipo sinusoidal tal como descrito na Figura 45.

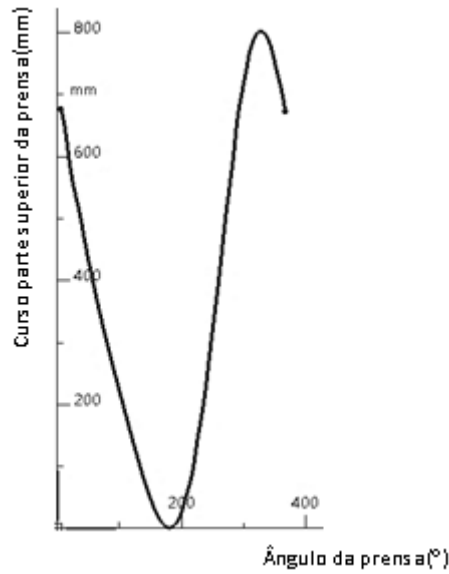


Figura 45 - Diagrama cinemático da prensa, curso da parte superior em relação à base da prensa em função do ângulo da prensa.

Os elementos móveis da ferramenta possuem um curso associado, tal que a sua curva cinemática será constante no valor do seu curso até que o curso da prensa é de tal modo pequeno que o pisa encosta na peça que se encontra no elevador e cerra, momento a partir do qual estes elementos passam a ter um deslocamento similar ao movimento sinusoidal da prensa, tal como ilustrado na Figura 46.

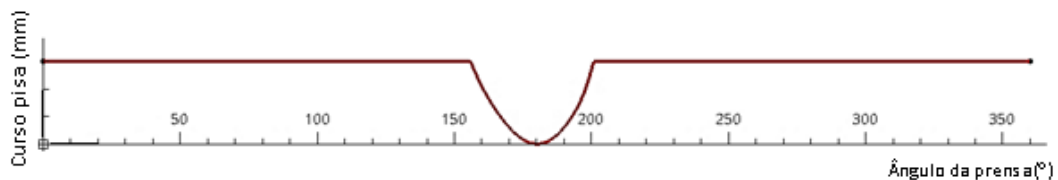


Figura 46- Diagrama cinemático do pisa.

Para o sistema transfer triaxial tal, como demonstrado anteriormente na Figura 20, existirá um diagrama de curso em função do ângulo da prensa para cada um dos 3 eixos xx, yy e zz.

Neste TFM além destes pressupostos foram também considerados os seguintes valores de cursos e cadência, conforme a tabela 3.

Tabela 3 - Cursos dos componentes e cadência da ferramenta

Curso do sistema transfer – eixo xx	925 mm
Curso do sistema transfer – eixo yy	330 mm
Curso do sistema transfer – eixo zz	245 mm
Curso do elemento móvel da ferramenta– Pisa	80 mm
Curso do elemento móvel da ferramenta – Cerra	140 mm
Curso do elemento móvel da ferramenta – Elevador	50 mm
Cadência da prensa	10 GPM
Tempo de ciclo	6 s

Toda estas variáveis foram definidas de forma empírica dentro dos valores permitidos pelos sistemas transfer e prensa.

Como forma de tornar possível a validação cinemática do sistema transfer numa licença básica do software DS CATIA V5, recorreu-se à criação de um ‘template’ onde, uma vez que o comportamento cinemático de uma prensa e de um sistema transfer triaxial é conhecido e pode ser descrito como o que se encontra ilustrado na figura 20, essas curvas cinemáticas foram introduzidas no software. As curvas cinemáticas são desenhadas parametricamente no software com elementos de linha/curva de forma a criar gráficos do tipo deslocamento vs. ângulo da cambota da prensa, tal como o ilustrado na figura 20. Sendo o principal objetivo deste ‘template’ a análise cinemática e uma vez que as curvas cinemáticas são conhecidas, foram posteriormente criados constrangimentos ao modelo CAD apresentado de forma a que os valores que definem esse constrangimento sejam dependentes do valor do ângulo da prensa. Por exemplo utilizou-se um constrangimento do tipo distância que ao estar associado à curva do movimento sinusoidal da prensa, irá definir a altura a que a parte superior da ferramenta se encontra em relação à base da prensa. Em relação ao sistema transfer, o acionamento dos movimentos em cada eixo é dependente do ângulo a que cambota da prensa se encontra, tendo sido considerado para o estudo deste TFM que as variáveis de projeto seriam definidas pelos valores dos ângulos de início e fim do acionamento do movimento, tal como demonstrado na Figura 47.

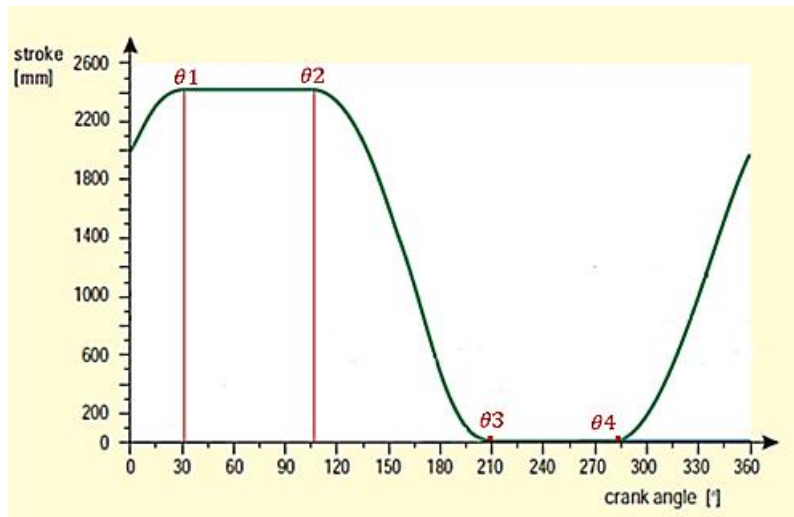


Figura 47 - Esquema demonstrativo dos ângulos de atuação dos movimentos (Adaptado de [4]). θ_1 – Ângulo final do avanço; θ_2 - Ângulo inicial do recuo; θ_3 – Ângulo final do recuo; θ_4 – Ângulo inicial do avanço.

Assim pode-se desde já definir claramente que a simulação cinemática conta com 12 variáveis de projeto, quatro que controlam o movimento de avanço e recuo de cada um dos três eixos xx, yy e zz. Este movimento tridimensional genérico do sistema transfer pode ser dividido num conjunto de 2 movimentos, um de avanço e outro de recuo, em cada um dos 3 eixos, tal como demonstrado na Figura 48.

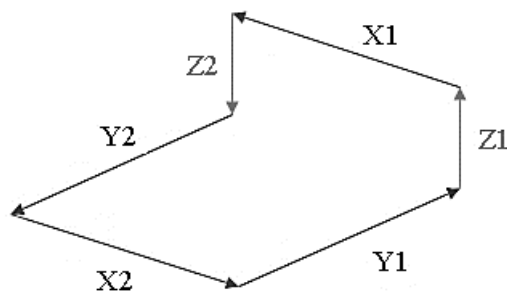


Figura 48 - Esquema tridimensional dos movimentos individuais do sistema transfer nos eixos xx, yy e zz.

Este movimento tridimensional está representado de forma simplificada nas tabelas 4 e 5 para posições de início e fim.

Tabela 4 - Esquema cinemático dos movimentos Y1, Z1 e X1 do modelo simplificado

	Início	Fim
Y1		
Z1		
X1		

Tabela 5 - Esquema cinemático dos movimentos Z2, Y2 e X2 do modelo simplificado

	Início	Fim
Z2		
Y2		
X2		

Assim no software Ds CATIA V5 foram criadas estas 12 variáveis que ficaram ligadas à construção da curva cinemática que define o movimento de cada um dos eixos, e o modelo de otimização foi criado para que para cada solução estes valores sejam atualizados nas respectivas curvas.

Tal como referido anteriormente recorreu-se à utilização de parâmetros internos do software Ds CATIA em conjunto com constrangimentos físicos para que com a variação do valor do ângulo da prensa, sejam atualizados os parâmetros dos constrangimentos e em resultado as respectivas posições no espaço tridimensional dos vários elementos da ferramenta.

Capítulo 4 – Metodologia de validação cinemática virtual de uma ferramenta transfer

No decorrer deste capítulo serão apresentadas as várias abordagens realizadas ao desenvolvimento da metodologia de validação e otimização cinemática da ferramenta transfer. O principal objetivo é o desenvolvimento de uma metodologia que permita a validação cinemática virtual numa licença base do software DS CATIA V5 assim como o desenvolvimento de um algoritmo de otimização que garanta um caminho eficiente realizado pelo sistema transfer.

As abordagens à metodologia a desenvolver, correspondem a estudos de implementação de algoritmos que permitam obter diferentes configurações do modelo cinemático do sistema transfer, cujas variáveis são os ângulos de início e de fim que definem o deslocamento do sistema transfer em função do ângulo da cambota da prensa. Estas configurações são avaliadas de acordo com a validação cinemática do sistema transfer tal que não ocorram colisões no mesmo e que permitam obter uma trajetória do sistema transfer tal que a ferramenta trabalhe na maior cadência possível. Como forma de expor de forma simples todas as abordagens realizadas, serão apresentados os principais objetivos de cada abordagem, a análise dos resultados obtidos com a respetiva abordagem e a definição dos próximos passos com vista ao melhoramento da metodologia a desenvolver.

4.1. Primeira Abordagem / Algoritmo I

Nesta primeira abordagem pretendia-se utilizar a plataforma de comunicação entre softwares, utilizando a interface Component Object Model (COM), como ponte de comunicação entre os softwares de simulação Ds CATIA V5 e o software que iria ficar encarregue da simulação, o Microsoft Excel, tal como ilustrado na Figura 49. Definiu-se como principal objetivo o estudo da capacidade de comunicação entre os softwares.

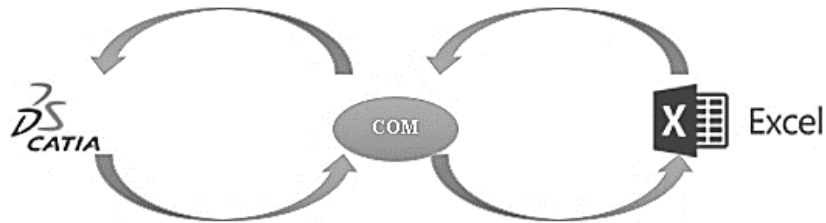


Figura 49 – Esquema da interação utilizada entre softwares

Uma vez definido qual o objetivo, foi desenvolvido um modelo de otimização onde se considerou como objeto de estudo o movimento no eixo dos yy conforme representado na Figura 47 e considerou-se que o deslocamento entre os ângulos θ_4 e θ_1 , assim como entre θ_2 e θ_3 , representados na Figura 46, é constante, sendo que apesar de se desprezar o fenómeno de aceleração e desaceleração, estes pontos irão determinar com exatidão a posição inicial e final do mecanismo. Assim, com base neste pressuposto foi desenvolvido um modelo de otimização onde se pretendia maximizar a diferença entre ângulo inicial e o ângulo final ($\Delta\theta$), pois a maximização desta variável é diretamente responsável por uma diminuição da velocidade de deslocamento e que, por sua vez, irá traduzir-se num aumento da cadência. Como variáveis de projeto foram inicialmente definidos os ângulos θ_1 , θ_2 , θ_3 e θ_4 , porém, nesta metodologia apenas foi considerado como variável de projeto o ângulo θ_1 e os restantes ângulos considerados constantes. Foi também considerado, definir um constrangimento físico, tal que o ângulo θ_1 , no eixo yy , não poderia ser inferior a 180° , pois é a altura em que a prensa se encontra completamente encerrada e como tal o sistema transfer não pode estar a desenvolver o movimento de entrada.

$$\Delta\theta = (\theta_3 - \theta_2) + ((360 - \theta_4) + \theta_1) \quad (3)$$

Na equação 3 encontra-se a equação que traduz a função objetivo e que será utilizada no problema de otimização.

Assim definiu-se o seguinte modelo de otimização:

Função Objetivo:	Maximizar $\Delta\theta$
Variáveis de Projeto:	θ_1
Constrangimentos:	Velocidade máxima de 3 m/s Ausência de Colisão

Com esta abordagem pretendia-se que o *Ds CATIA V5*, enviasse informação relativa aos ângulos para o *Excel*, onde seria executada uma rotina de otimização interna do *Excel* e seriam retornados os resultados ao *Ds CATIA V5*. Contudo da aplicação desta metodologia resultou o seguinte problema: como só existia comunicação com o *Ds CATIA V5* após ser executada a rotina de otimização, como a função objetivo é a maximização de $\Delta\theta$ resultava em que o valor de θ_1 convergia sempre para o seu valor mínimo, 180º neste caso.

Após esta a implementação e teste desta metodologia foram levantadas as seguintes questões:

- a) Como garantir que o software *Ds CATIA V5* verifica se ocorre colisão na ferramenta impedindo o valor de θ_1 convergir para o valor mínimo?
- b) Ao considerar os 3 eixos de movimento *xx*, *yy* e *zz*, como considerar as variáveis de projeto dos três eixos numa só função objetivo?

Assim como forma de responder a estas questões e primariamente, à questão b), o movimento do sistema transfer tridimensional representado na Figura 47, foi analisado de forma independente, chegando-se à conclusão que o movimento do sistema de transfer se poderia agregar da seguinte forma cumulativa, tal como se representa na Figura 50.

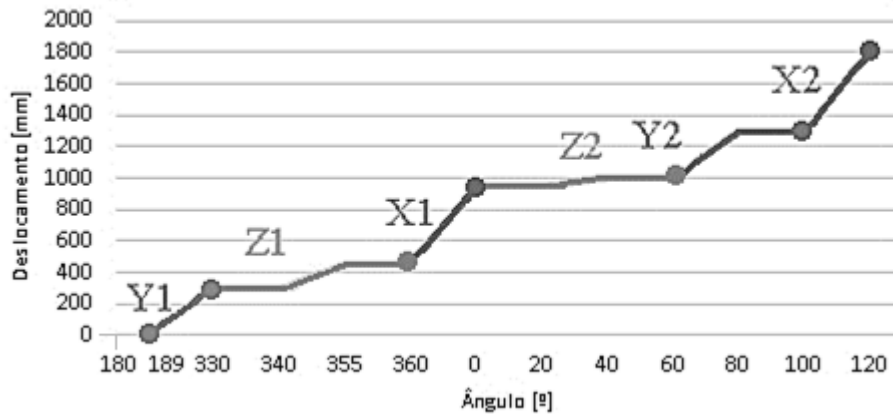


Figura 50 – Deslocamento de um sistema transfer em função do ângulo da cambota da prensa durante um ciclo

Na Figura 50, pode-se verificar que o movimento se inicia com o deslocamento Y1, ou seja, após os 180° da prensa pois considera-se que é este o movimento que inicia o ciclo de transferencia da peça. Assim analisando a figura, e pensando em termos de tempo, ao revés de velocidade, pode-se concluir que o tempo que o transfer demora a realizar o seu percurso, desde a sua posição casa Y1 (posição inicial) até que retorne novamente à posição casa é no máximo os 360°, que corresponde ao tempo que a prensa demora a realizar um ciclo. Com esta conclusão, a questão b), de como considerar as variáveis de projeto dos três eixos numa só função objetivo fica solucionada, ficando a função objetivo explorada na primeira abordagem obsoleta. A nova função objetivo tem em conta o tempo de todos os troços (Y1, Z1, X1, Z2, Y2 e X2) agregando assim todas as 12 variáveis de projeto. Desta forma, o tempo total do ciclo de transfer é definido por:

$$t_{ciclo_{transfer}} = \sum_{i=1}^2 t_{xi,yi,zi} \quad (4)$$

Onde $t_{xi,yi,zi}$ é o tempo de cada troço i de movimento individual nos eixos xx, yy e zz, tal que i=1 ou 2 para avanço ou retorno, respetivamente.

A nova função objetivo passa a ser definida por:

$$t_{ciclo_{transfer}} - t_{ciclo_{prensa}} \geq 0 \quad (5)$$

Onde $t_{ciclo_{prensa}}$ é o tempo de ciclo da prensa, definido em 6 s de acordo com a tabela 3.

Em conclusão desta abordagem, foi criada a necessidade de uma nova abordagem que explore a possibilidade da implementação desta nova função objetivo e que permita a detecção de colisões ao longo da simulação do ciclo da prensa.

4.2. Segunda Abordagem / Algoritmo II

Após a realização da primeira abordagem, foi considerado o estudo de algumas referências bibliográficas tais como Garcia-Sedano et al. [15] e Svensson et al.[16] que recorrendo a algoritmos do tipo genético calculam os valores ótimos para as variáveis de projeto. Assim como forma de utilizar as conclusões aprendidas com a primeira abordagem e como forma de permitir a medição da colisão durante a simulação considerando o deslocamento do mecanismo transfer em função do ângulo da prensa tal como demonstrado na Figura 50, foi abandonada a ideia da utilização da rotina de otimização do Excel e foi explorada a utilização de algoritmos na linguagem de programação Visual Basic for Applications (VBA) em ambiente DS Catia V5, que tenham o seu comportamento similar a algoritmos genéticos.

Tal como indicado anteriormente, o principal objetivo desta abordagem é a implementação da nova função objetivo, descrita pela equação (4) e que permita a detecção de possíveis colisões e que ao mesmo tempo garanta que a velocidade máxima durante o deslocamento seja inferior a 3m/s.

Após a análise da Figura 47 é possível definir que cada um dos movimentos representados irá possuir um ângulo que controla o início do movimento e o seu fim podendo ser transcrito como o esquema da Figura 51 sugere.

Y1 [°]		Z1 [°]		X1 [°]		Z2 [°]		Y2 [°]		X2 [°]	
Início	Fim	Início	Fim	Início	Fim	Início	Fim	Início	Fim	Início	Fim

Figura 51 - Esquema das variáveis de projeto utilizadas

Desta forma são consideradas 12 variáveis de projeto definidas por $Y1_{Início}$, $Y1_{Fim}$, $Z1_{Início}$, $Z1_{Fim}$, $X1_{Início}$, $X1_{Fim}$, $Z2_{Início}$, $Z2_{Fim}$, $Y2_{Início}$, $Y2_{Fim}$, $X2_{Início}$ e $X2_{Fim}$. Assim, indo de encontro ao comportamento que se pretende simular com algoritmos genéticos esta abordagem inicia-se com um algoritmo que permita a criação das 12 variáveis de projeto de forma aleatória. Inicialmente foi também considerado que apenas o ângulo que define o início de Y1 seria criado de forma aleatória, tendo sido considerado que o valor de Y1 estaria dentro dos limites $Y1_{Início} > 180^\circ$ e $Y1_{Início} \leq 240^\circ$, representado em destaque a sombreado na Tabela 6. Para as outras variáveis de projeto foram utilizados valores resultantes de uma simulação anterior onde os constrangimentos são cumpridos, apresentados na Tabela 6 em analogia à Figura 51.

Tabela 6 - Valores considerados para as variáveis de projeto

Y1 [°]		Z1 [°]		X1 [°]		Z2 [°]		Y2 [°]		X2 [°]	
180<	245	240	310	290	1	350	32	33	80	90	180

Este algoritmo foi desenvolvido de forma a que inicialmente seja criada uma solução, que contenha as 12 variáveis de projeto que é posteriormente avaliada em termos de variação de ângulo (diferença entre os ângulos de início e fim), velocidade e tempo de execução do movimento. Ao avaliar a velocidade da solução criada é posteriormente verificado se esta solução cumpre com o constrangimento de velocidade máxima, sendo a solução excluída se esse pressuposto não se verificar.

Uma vez verificado o critério de velocidade máxima as variáveis de projeto são atualizadas no software Ds CATIA V5, fazendo com que todas as curvas cinemáticas sejam atualizadas de acordo com os novos parâmetros. Seguidamente é realizada uma análise de colisões no software Ds CATIA V5, onde se simula todas as posições dos componentes ao longo de um ciclo da prensa e caso exista colisão a solução é automaticamente excluída iniciando o ciclo novamente. Caso o ciclo se conclua sem que exista colisão essa solução considera-se como uma solução válida, sendo apresentado o valor da sua função objetivo. Este algoritmo encontra-se descrito no fluxograma da Figura 52.

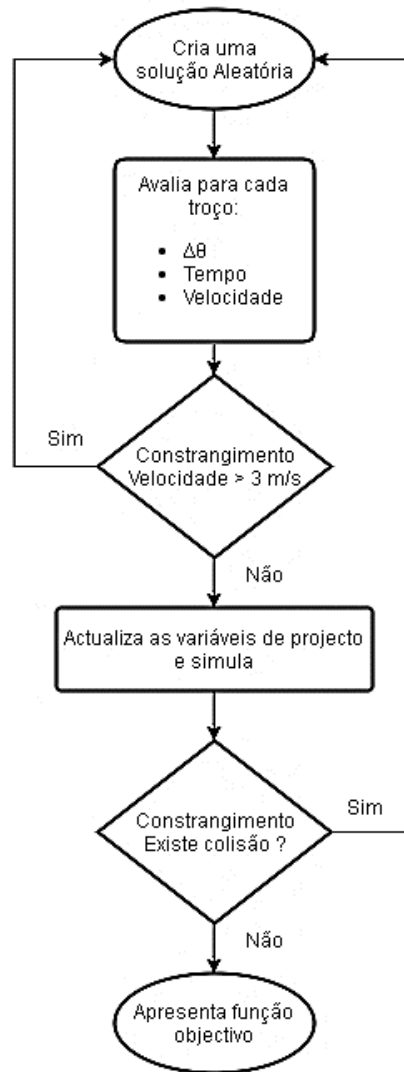


Figura 52 - Fluxograma do algoritmo II.

Com a implementação do algoritmo descrito nesta metodologia foi possível atingir o principal objetivo desta metodologia, sendo que o resultado é apresentado na Tabela 7, na qual se apresenta em destaque a sombreado o valor de $Y1_{\text{Início}}$ obtido e o respetivo valor da função objetivo.

Tabela 7- Tabela 7 - Resultados do algoritmo II.

Y1 [°]		Z1 [°]		X1 [°]		Z2 [°]		Y2 [°]		X2 [°]		F. Objetivo[s]
232	245	240	310	290	1	350	32	33	80	90	180	0.50

Em conclusão, tendo-se conseguido atingir um dos principais objetivos com esta abordagem, para o próximo passo foi definido o desenvolvimento de uma nova abordagem com um algoritmo que com um funcionamento similar ao apresentado nesta abordagem, produza em vez de uma, várias soluções que respeitem os constrangimentos.

4.3. Terceira Abordagem / Algoritmo III

Nesta abordagem o principal objetivo foi o desenvolvimento de um algoritmo que permitisse a criação de 12 variáveis de projeto, N vezes até que sejam calculadas 20 soluções de 12 variáveis que cumpram com os constrangimentos. Assim, tendo como base a abordagem anterior, foi realizada uma operação de repetição do algoritmo anterior até que fossem obtidas as 20 soluções. Após a realização desta alteração o algoritmo foi testado tendo-se obtido os resultados apresentados na Tabela 8, na qual se apresentam em destaque a sombreado os valores de $Y1_{Início}$ obtidos e os respetivos valores da função objetivo.

Tabela 8 - Resultados da implementação do algoritmo III.

Nº população	y1 [°]		z1 [°]		x1 [°]		z2 [°]		y2 [°]		x2 [°]		F. Objetivo[s]
1	238	245	240	310	311	1	2	32	33	80	90	180	1,03
2	237	245	240	310	311	1	2	32	33	80	90	180	1,02
3	229	245	240	310	311	1	2	32	33	80	90	180	0,88
4	232	245	240	310	311	1	2	32	33	80	90	180	0,93
5	237	245	240	310	311	1	2	32	33	80	90	180	1,02
6	234	245	240	310	311	1	2	32	33	80	90	180	0,97
7	229	245	240	310	311	1	2	32	33	80	90	180	0,88
8	234	245	240	310	311	1	2	32	33	80	90	180	0,97
9	229	245	240	310	311	1	2	32	33	80	90	180	0,88
10	230	245	240	310	311	1	2	32	33	80	90	180	0,90
11	236	245	240	310	311	1	2	32	33	80	90	180	1,00
12	236	245	240	310	311	1	2	32	33	80	90	180	1,00
13	231	245	240	310	311	1	2	32	33	80	90	180	0,92
14	231	245	240	310	311	1	2	32	33	80	90	180	0,92
15	230	245	240	310	311	1	2	32	33	80	90	180	0,90
16	235	245	240	310	311	1	2	32	33	80	90	180	0,98
17	231	245	240	310	311	1	2	32	33	80	90	180	0,92
18	237	245	240	310	311	1	2	32	33	80	90	180	1,02
19	234	245	240	310	311	1	2	32	33	80	90	180	0,97
20	234	245	240	310	311	1	2	32	33	80	90	180	0,97

A análise dos resultados obtidos permite verificar que nas 20 soluções obtidas existem soluções repetidas, comprovadas também por idêntico valor da função objetivo. Durante o teste realizado a este algoritmo observou-se que o tempo necessário para a criação das vinte soluções era bastante elevado, cerca de 40 minutos, simulado num computador de características Core i7-4710 HQ com 8 Gb de memória. Em conclusão, o algoritmo implementado satisfaz com sucesso o propósito da criação de 20 soluções válidas, mas o tempo de computação poderá tornar-se crítico quando se introduzirem operadores genéticos no algoritmo. Assim, existe a necessidade de explorar uma nova abordagem que permita a diminuição do tempo computacional necessário para obtenção da primeira população.

4.4. Quarta Abordagem / Algoritmo IV

Como forma de tentar diminuir o tempo computacional necessário para a criação das vinte soluções, foi criado um algoritmo que cria uma primeira solução que valida ambos os constrangimentos, sendo que as soluções posteriores são criadas tendo por base a primeira solução com uma tolerância associada. Tal como anteriormente, apenas se iria variar a variável $Y_{1_{Início}}$ e após a primeira solução obtida, as restantes soluções irão ter valores que estão compreendidos entre os valores $y_{1_{Início}} \pm 20^\circ$. Após esta verificação das soluções, tal como acontecia na abordagem anterior, são enviadas para o software Ds CATIA V5 que atualiza as variáveis de projeto, fazendo com que todas as curvas cinemáticas sejam atualizadas de acordo com os novos parâmetros. Seguidamente, é realizada uma análise de colisões no software Ds CATIA V5, e caso exista colisão existem novamente dois cenários possíveis, o primeiro cenário em que a solução que apresenta colisão é a primeira solução e caso aconteça essa solução é excluída e inicia-se novamente o ciclo com a criação de uma solução de forma aleatória, e um segundo cenário em que a solução que apresenta colisão não é a primeira. Neste segundo cenário, tal como foi considerado no caso de a solução não verificar o constrangimento de velocidade máxima e não ser a primeira solução, a solução que apresenta colisão é excluída e cria-se nova solução com base na primeira solução, considerando-se também, que o valor da variável de projeto $Y_{1_{Início}}$ seria criado com base no valor da primeira solução associada a uma variação de cerca

de 20 graus. Este ciclo foi repetido até o tamanho da população ser de 20 soluções. Este algoritmo encontra-se descrito no fluxograma da Figura 53.

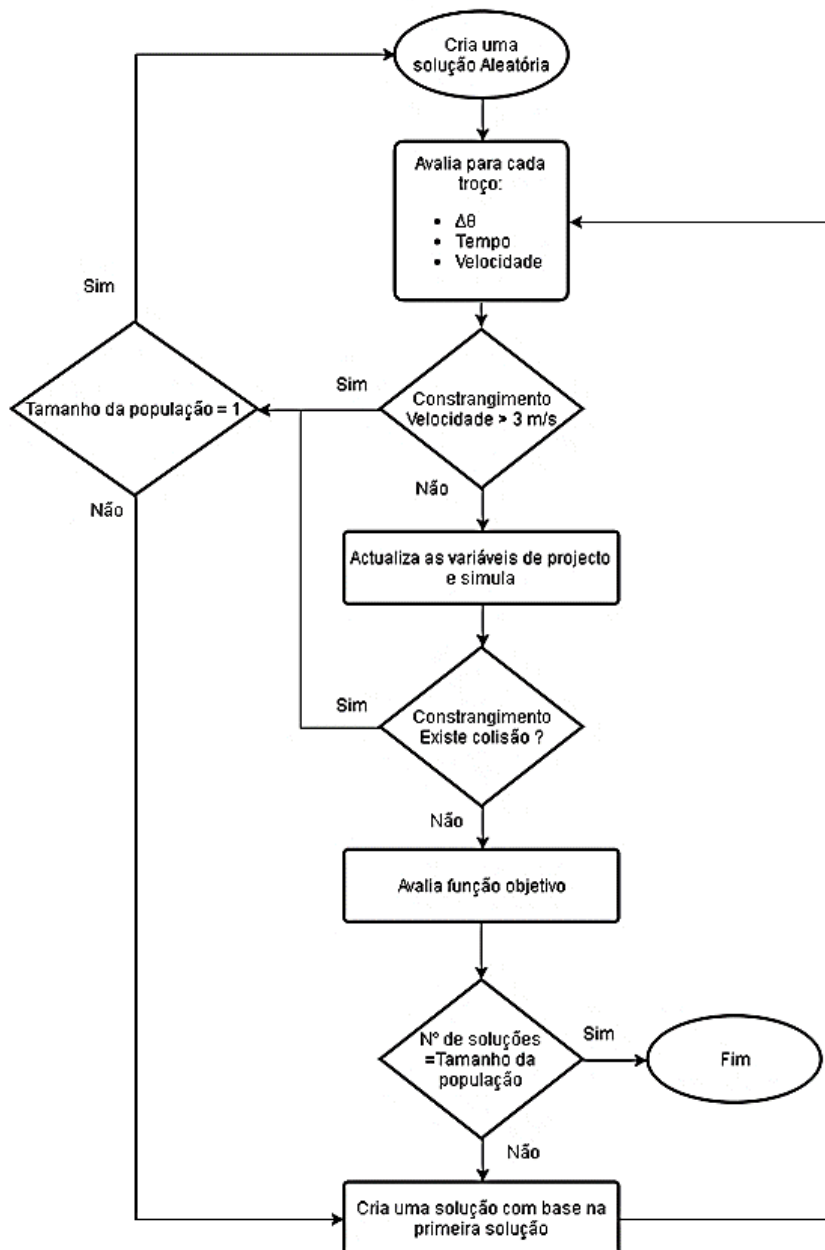


Figura 53 - Fluxograma do algoritmo IV.

Após a implementação e teste do algoritmo apresentado foram obtidos os resultados apresentados na tabela 9, na qual se destacam a sombreado os valores de $Y_{1\text{Início}}$ obtidos e os respetivos valores da função objetivo.

Tabela 9 - Resultados da implementação do algoritmo IV

Nº população	y1 [°]		z1 [°]		x1 [°]		z2 [°]		y2 [°]		x2 [°]		F. Objetivo[s]
<u>1</u>	<u>230</u>	<u>245</u>	<u>240</u>	<u>310</u>	<u>311</u>	<u>1</u>	<u>2</u>	<u>32</u>	<u>33</u>	<u>80</u>	<u>90</u>	<u>180</u>	<u>0,90</u>
2	250	245	240	310	311	1	2	32	33	80	90	180	1,23
3	274	245	240	310	311	1	2	32	33	80	90	180	1,63
4	275	245	240	310	311	1	2	32	33	80	90	180	1,65
5	269	245	240	310	311	1	2	32	33	80	90	180	1,55
6	267	245	240	310	311	1	2	32	33	80	90	180	1,52
7	271	245	240	310	311	1	2	32	33	80	90	180	1,58
8	251	245	240	310	311	1	2	32	33	80	90	180	1,25
9	270	245	240	310	311	1	2	32	33	80	90	180	1,57
10	250	245	240	310	311	1	2	32	33	80	90	180	1,23
<u>11</u>	<u>236</u>	<u>245</u>	<u>240</u>	<u>310</u>	<u>311</u>	<u>1</u>	<u>2</u>	<u>32</u>	<u>33</u>	<u>80</u>	<u>90</u>	<u>180</u>	<u>1,00</u>
<u>12</u>	<u>238</u>	<u>245</u>	<u>240</u>	<u>310</u>	<u>311</u>	<u>1</u>	<u>2</u>	<u>32</u>	<u>33</u>	<u>80</u>	<u>90</u>	<u>180</u>	<u>1,03</u>
13	258	245	240	310	311	1	2	32	33	80	90	180	1,37
14	269	245	240	310	311	1	2	32	33	80	90	180	1,55
15	261	245	240	310	311	1	2	32	33	80	90	180	1,42
16	265	245	240	310	311	1	2	32	33	80	90	180	1,48
17	263	245	240	310	311	1	2	32	33	80	90	180	1,45
18	271	245	240	310	311	1	2	32	33	80	90	180	1,58
19	270	245	240	310	311	1	2	32	33	80	90	180	1,57
20	265	245	240	310	311	1	2	32	33	80	90	180	1,48

Analisando os resultados apresentados na Tabela 9, pode-se verificar que como não foi criado nenhum constrangimento que defina que Y1 inicial tem que ser sempre inferior ao valor de Y1 final, só a 1ª a 11ª e a 12ª solução são válidas, que se apresentam em destaque a sublinhado na mesma tabela. Contudo o método considerado para criação das novas soluções em caso do não cumprimento dos constrangimentos, cria as soluções com base na primeira solução obtida com uma diferença percentual em torno da mesma, o que leva a uma introdução de elitismo o que poderá condicionar a convergência do algoritmo. Uma vez que não se verificaram vantagens nesta abordagem vai-se considerar a mesma como obsoleta e para a próxima abordagem vai novamente ser considerada a quarta abordagem para a criação de uma população de 20 soluções.

4.5. Quinta Abordagem / Algoritmo V

Nesta abordagem o principal objetivo era aproximar o algoritmo IV representado na Figura 53 considerando a criação de uma população inicial de 20 soluções válidas, conforme obtidas na terceira abordagem / algoritmo III, acrescentando operações genéticas tais como a seleção, o crossover e a mutação.

Nesta abordagem começou-se por criar uma população de 20 soluções, ou cromossomas, tal como descrito na abordagem anterior, sendo seguidamente cada solução avaliada em termos de variação de ângulo, velocidade e tempo de execução do movimento assim como o cumprimento dos constrangimentos, tal como acontecia nas abordagens apresentadas anteriormente. Porém nesta abordagem foi introduzida a penalização da função objetivo caso um dos constrangimentos não seja cumprido. A função objetivo será multiplicada por 10 sempre que ocorra uma penalização por não cumprimento dos constrangimentos, podendo ocorrer o caso de dupla penalização caso a solução não cumpra com os dois constrangimentos de velocidade máxima e ausência de colisão. Quando todas as soluções se encontrem devidamente avaliadas é iniciado o processo seleção, tendo-se considerado nesta metodologia um processo de seleção com base no método da roleta, que tem em conta a probabilidade de seleção de uma determinada solução com base no ajustamento da função objetivo.

De seguida é aplicado o processo de crossover, em que foi considerada uma probabilidade de crossover de 85%, tendo-se mantido o ponto de crossover flutuante e aleatório podendo ocorrer em qualquer uma das variáveis de projeto. A última operação aplicada é a mutação onde se considerou uma probabilidade de mutação de 1%, sendo que sempre que o valor seja inferior a este limite serão selecionados $0.1 \times (\text{número total de genes})$ para sofrer a mutação. A posição desses genes também é selecionada de forma aleatória.

Uma vez realizados os processos genéticos de seleção, crossover e mutação a população resultante é vulgarmente chamada de 'offsring' ou descendência. De seguida o número da geração será automaticamente aumentado e será esta solução que irá realizar novamente o processo de avaliação e penalização, caso os constrangimentos não sejam cumpridos, bem como a repetição dos processos de

seleção, crossover e mutação até que o critério de paragem seja atingido ou o número da geração seja igual a 10. Este algoritmo encontra-se descrito no fluxograma da Figura 54.

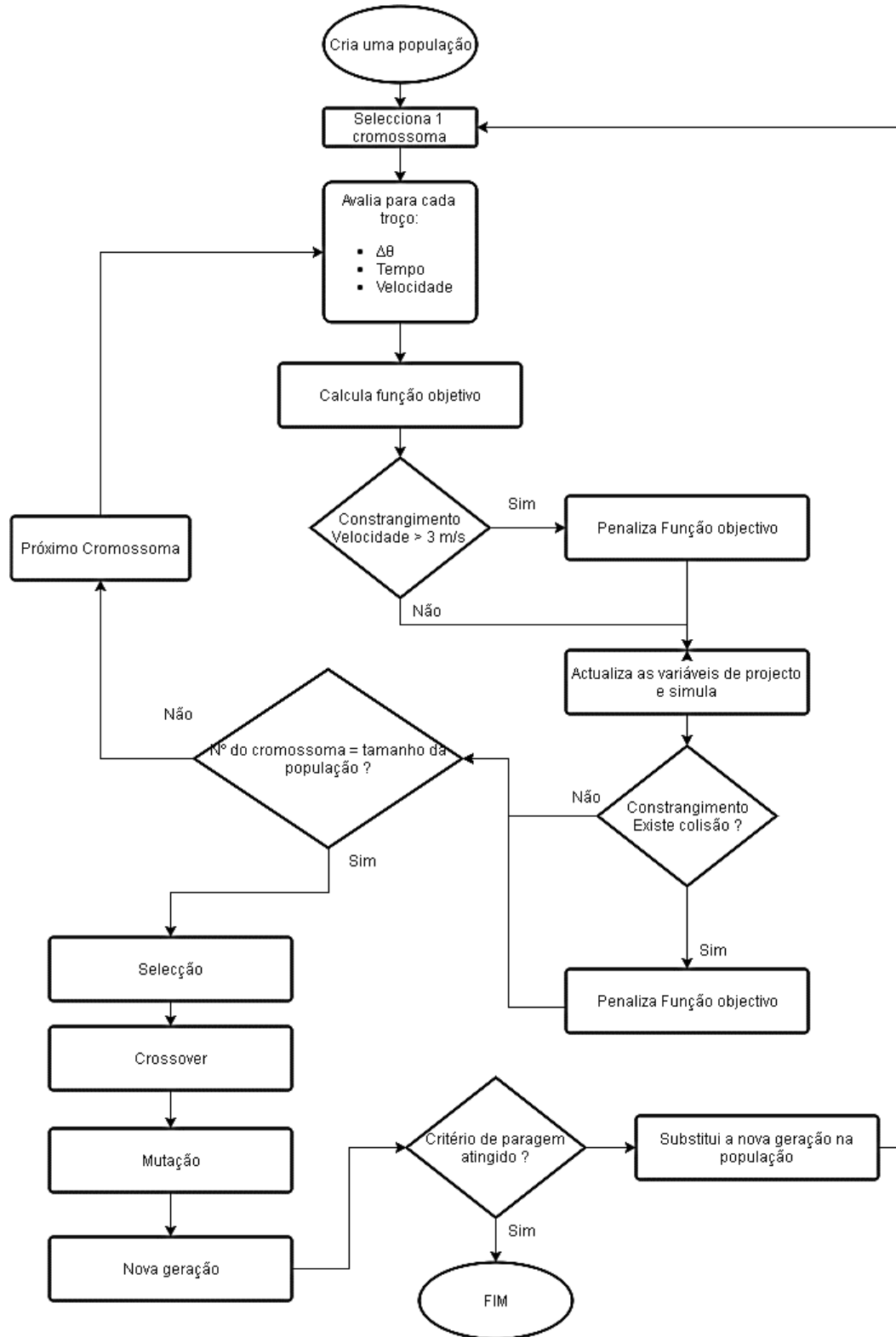


Figura 54 - Fluxograma do algoritmo V.

Com a implementação e teste deste algoritmo foram obtidos os seguintes resultados para a primeira população, conforme se apresenta na Tabela 9 na qual se destacam a sombreado os valores de $Y1_{Início}$ obtidos e os respectivos valores da função objetivo.

Tabela 10- Valores das variáveis de projeto na primeira população

Nº população	y1 [°]		z1[°]		x1[°]		z2[°]		y2[°]		x2[°]		F. Objetivo[s]
1	240	290	291	310	311	1	2	32	33	80	90	180	1,23
2	226	290	291	310	311	1	2	32	33	80	90	180	1,00
3	224	290	291	310	311	1	2	32	33	80	90	180	0,97
4	230	290	291	310	311	1	2	32	33	80	90	180	1,07
5	232	290	291	310	311	1	2	32	33	80	90	180	1,10
6	239	290	291	310	311	1	2	32	33	80	90	180	1,22
7	237	290	291	310	311	1	2	32	33	80	90	180	1,18
8	225	290	291	310	311	1	2	32	33	80	90	180	0,98
9	240	290	291	310	311	1	2	32	33	80	90	180	1,23
10	222	290	291	310	311	1	2	32	33	80	90	180	0,93
11	220	290	291	310	311	1	2	32	33	80	90	180	0,90
12	229	290	291	310	311	1	2	32	33	80	90	180	1,05
13	228	290	291	310	311	1	2	32	33	80	90	180	1,03
14	234	290	291	310	311	1	2	32	33	80	90	180	1,13
15	225	290	291	310	311	1	2	32	33	80	90	180	0,98
16	224	290	291	310	311	1	2	32	33	80	90	180	0,97
17	233	290	291	310	311	1	2	32	33	80	90	180	1,12
18	233	290	291	310	311	1	2	32	33	80	90	180	1,12
19	219	290	291	310	311	1	2	32	33	80	90	180	0,88
20	218	290	291	310	311	1	2	32	33	80	90	180	0,87

Na tabela 10 estão os resultados das 20 soluções criadas, sendo que estas 20 soluções são criadas de acordo com a terceira abordagem / algoritmo III, e como tal todas as soluções satisfazem os dois constrangimentos de velocidade máxima e ausência de colisão.

Após a 5ª geração foram obtidos os resultados apresentados na tabela 11.

Tabela 11– Valores das variáveis de projeto após a 5 geração.

Nº população	y1 [°]		z1[°]		x1[°]		z2[°]		y2[°]		x2[°]		F. Objetivo[s]
1	240	290	291	310	311	1	2	32	33	80	90	180	1,23
2	222	290	291	310	311	1	2	32	33	80	90	180	0,93
3	240	290	291	310	311	1	2	32	33	80	90	180	1,23
4	225	290	291	310	311	1	2	32	33	80	90	180	0,98
5	224	290	291	310	311	1	2	32	33	80	90	180	0,97
6	224	290	291	310	311	1	2	32	33	80	90	180	0,97
7	240	290	291	310	311	1	2	32	33	80	90	180	1,23
8	222	290	291	310	311	1	2	32	33	80	90	180	0,97
9	225	290	291	310	311	1	2	32	33	80	90	180	1,22
10	224	290	291	310	311	1	2	32	33	80	90	180	0,97
11	225	290	291	310	311	1	2	32	33	80	90	180	0,90
12	224	290	291	310	311	1	2	32	33	80	90	180	0,97
13	239	290	291	310	311	1	2	32	33	80	90	180	1,22
14	224	290	291	310	311	1	2	32	33	80	90	180	0,97
15	220	290	291	310	311	1	2	32	33	80	90	180	0,90
16	224	290	291	310	311	1	2	32	33	80	90	180	0,97
17	224	290	291	310	311	1	2	32	33	80	90	180	0,97
18	224	290	291	310	311	1	2	32	33	80	90	180	0,97
19	222	290	291	310	311	1	2	32	33	80	90	180	0,93
20	240	290	291	310	311	1	2	32	33	80	90	180	1,23

Durante a realização dos testes ao algoritmo apresentado, foram detetados alguns défices de memória necessária para a realização do número de gerações inicialmente definida de 10 gerações. Este fenómeno, deve-se essencialmente ao facto do software só libertar a memória quando termina o algoritmo, ou seja, quanto maior for o número de avaliações maior irá ser o valor da memória consumida. Tendo isso em conta foi considerado para esta abordagem que irão ser realizadas apenas 5 gerações. Analisando as tabelas 10 e 11, pode-se identificar que o gene dominante após a quinta geração é o que apresenta a função objetivo de 0.97 s conseguindo 8 soluções em 20. Assim poderá extrapolar-se que caso se continuasse a simular este seria o gene que iria prevalecer.

Sendo que a implementação deste algoritmo tinha como principal objetivo, a aproximação a um algoritmo genético pode-se dizer que essa foi bem-sucedida, apesar

do valor da função objetivo não ser bastante expressivo, pois com esta abordagem apenas se faz variar o valor de Y1 inicial o que não causa um grande impacto no valor da função objetivo. No entanto foi possível verificar que mesmo com apenas 5 gerações já existe uma solução que domina completamente as restantes. Infelizmente o tempo de computação continua a ser bastante elevado, cerca de 1h50 m, simulado num computador de características Core i7-4710 HQ e 8 Gb de memória. Assim numa próxima abordagem deverá procurar-se uma solução que permita a criação das 12 variáveis de projeto, garantindo a precedência entre elas, isto é, que à sequência das variáveis correspondam ângulos exequíveis que sejam cumulativamente superiores aos ângulos das variáveis anteriores. Deverá igualmente tentar realizar-se a redução do tempo computacional de simulação necessário para a obtenção de resultados.

4.6. Sexta Abordagem / Algoritmo VI

Tentando resolver os problemas levantados durante os testes da quinta abordagem procedeu-se ao desenvolvimento de um novo algoritmo onde se definiu como principal objetivo a criação das 12 variáveis de projeto, garantindo a precedência entre elas. Para isso foi necessário definir balizamentos que foram definidos com base nos constrangimentos físicos existentes na prensa, sendo que na tabela 12 estão descritos os constrangimentos considerados. Foi também considerado que os movimentos Z1, X1, Z2, Y2 e X2 se iniciam uma unidade de ângulo após o movimento imediatamente anterior, por forma a garantir uma precedência entre as variáveis.

Tabela 12 - Constrangimentos aplicados às variáveis de projeto

Y1	Início	$\geq 180 \wedge \leq 356$
	Fim	$> Y1_{\text{Início}} \wedge \leq 357$
Z1	Início	$= Y1_{\text{Fim}} + 1$
	Fim	$> Z1_{\text{Início}} \wedge \leq 359$
X1	Início	$= Z1_{\text{Fim}} + 1$
	Fim	$> 0 \wedge \leq 120$
Z2	Início	$= X1_{\text{Fim}} + 1$
	Fim	$> Z2_{\text{Início}} \wedge \leq 120$
Y2	Início	$= Z2_{\text{Fim}} + 1$
	Fim	$> Y2_{\text{Início}} \wedge \leq 120$
X2	Início	$= Y2_{\text{Fim}} + 1$
	Fim	$> X2_{\text{Início}} \wedge \leq 180$

O algoritmo apresentado nesta abordagem é baseado na quinta abordagem, considerando a criação de forma aleatória de uma população de 20 soluções, que são avaliadas, para que se possa aferir se respeitam os constrangimentos, sendo penalizadas caso não cumpram os constrangimentos. Todo o restante comportamento do algoritmo manteve-se idêntico ao descrito no fluxograma da Figura 53. Uma vez que com esta abordagem apenas se realiza a avaliação de 20 soluções, em vez de varias avaliações tal como acontecia na terceira abordagem, é expetável que o tempo necessário para a criação da população será reduzido de forma considerável e com isso todo o tempo de computação seja também diminuído. Durante o teste desta abordagem verificou-se que o tempo necessário para a criação e avaliação da população inicial passou de cerca de 40 minutos na terceira abordagem, para cerca de 20 minutos com esta abordagem, indo de encontro com o que se desejava.

Foram também obtidos os seguintes resultados para a população inicial, conforme se apresenta na Tabela 13.

Tabela 13 – População Inicial criada com a sexta abordagem / algoritmo VI.

Nº população	y1 [°]		z1 [°]		x1 [°]		z2 [°]		y2 [°]		x2 [°]		F. Objetivo[s]
1	215	232	233	346	347	34	35	44	45	115	116	129	15,17
2	249	256	257	263	264	95	96	109	110	111	112	146	1,80
3	311	312	313	344	345	116	117	118	119	120	121	139	29,50
4	193	198	199	345	346	31	32	50	51	96	97	102	16,00
5	274	328	329	347	348	93	94	115	116	117	118	152	21,17
6	283	340	341	359	360	100	101	109	110	118	119	124	2,73
7	263	342	343	355	356	62	63	71	72	87	88	118	2,50
8	230	279	280	288	289	112	113	114	115	119	120	176	9,83
9	342	346	347	351	352	41	42	57	58	100	101	177	2,83
10	325	342	343	346	347	25	26	82	83	89	90	112	3,63
11	270	332	333	336	337	44	45	55	56	112	113	139	2,27
12	324	351	352	355	356	114	115	116	117	118	119	127	3,37
13	289	298	299	320	321	26	27	80	81	102	103	176	1,97
14	246	341	342	354	355	73	74	112	113	115	116	145	1,77
15	334	349	350	358	359	32	33	48	49	102	103	162	2,95
16	250	335	336	357	358	6	7	101	102	111	112	154	1,68
17	292	332	333	352	353	17	18	46	47	78	79	179	1,97
18	224	256	257	313	314	72	73	104	105	113	114	179	8,33
19	229	247	248	341	342	57	58	90	91	118	119	148	14,33
20	279	344	345	356	357	50	51	98	99	103	104	118	2,77

Analisado a Tabela 13, que corresponde à população inicial criada, pode-se verificar que 5 soluções que se apresentam em destaque a sombreado na tabela possuem penalização pois possuem um valor na ordem de grandeza das dezenas, porem será expectável que com o aumento do número de gerações a quantidade de soluções que violem um ou ambos os constrangimentos, sejam reduzidas e até mitigadas. Na tabela 14, são apresentados os resultados do teste deste algoritmo após a 10ª geração.

Tabela 14 – População após a 10ª geração

Nº população	y1 [°]		z1 [°]		x1 [°]		z2 [°]		y2 [°]		x2 [°]		F. Objetivo[s]
1	334	349	350	358	359	32	18	46	47	78	79	176	2,42
2	334	349	350	358	359	32	18	46	47	78	79	179	2,42
3	292	349	350	358	359	32	18	46	47	78	79	179	2,02
4	334	349	350	358	359	32	18	46	47	78	79	179	2,42
5	249	256	257	352	353	17	18	46	47	78	79	176	1,25
6	292	332	333	352	353	17	18	46	47	78	79	176	2,02
7	249	349	350	358	359	32	18	46	47	78	79	179	13,83
8	334	349	350	358	359	32	18	46	47	78	79	179	2,42
9	334	349	350	358	359	32	18	46	47	78	79	179	2,42
10	292	349	350	358	359	32	18	46	47	78	79	179	1,72
11	292	349	350	358	359	32	18	46	47	78	79	179	1,72
12	249	256	257	263	264	95	96	109	115	119	120	176	13,83
13	334	349	350	358	359	32	18	46	47	78	79	179	2,42
14	292	349	350	358	359	32	18	46	47	78	79	179	1,72
15	292	332	333	352	353	17	18	46	49	102	103	176	1,97
16	334	349	350	358	359	32	18	46	47	78	79	179	2,42
17	292	349	350	358	359	32	18	46	47	78	79	179	1,72
18	292	332	333	352	353	17	33	48	49	102	103	176	2,27
19	292	349	350	358	359	32	18	46	47	78	79	179	1,72
20	334	349	350	358	359	32	18	46	47	78	79	179	2,42

Analisando os resultados da Tabela 14, pode-se verificar que ocorreu uma clara homogeneização dos valores das variáveis de projeto sendo que o valor da função objetivo acompanhou essa tendência e ficou perto dos 2 segundos. No entanto verificou-se que existem ainda 2 soluções, que se apresentam em destaque a sombreado na tabela, que possuem penalização. Detetou-se também que existia a

necessidade de criar uma validação na operação de crossover, pois segundo a tabela 12, por exemplo o valor de Z2 inicial deve ser o valor de X1 final mais uma unidade, facto que é facilmente evidenciado nas soluções 1 e 20 entre outras. Este tópico torna-se, portanto, um dos pontos a melhorar para a próxima abordagem.

Com a implementação desta metodologia pretendia-se a criação das 12 variáveis de projeto, garantindo a precedência entre elas, utilizando para isso o balizamento apresentado na tabela 12 e ao mesmo tempo foi introduzida uma alteração à forma como a primeira população é criada com o intuito de diminuir o tempo total de computação.

De forma geral pode considerar-se que os objetivos para esta abordagem foram atingidos e foram também retirados alguns pontos de melhoria nomeadamente, a criação de uma validação na operação de crossover para que a precedência entre os valores das variáveis de projeto de mantenha e a criação de uma nova abordagem para a criação da população inicial com o intuito de minimizar o tempo de computação e ao mesmo tempo otimizar os recursos disponíveis.

Como forma de implementar as melhorias, primeiramente foi definido que antes da realização da operação de crossover, as variáveis de projeto à esquerda e à direita do ponto de crossover, das duas soluções que irão sofrer a operação de crossover, passam a ser verificadas e se as precedências da tabela 12 se verificarem, efetua a operação caso contrário não ocorre a operação de crossover. Com a introdução deste mecanismo é expectável que o número de gerações diminua até que se atinja a convergência. Assim e no seguimento desta alteração foi implementada uma alteração na forma como a população inicial é criada, passando-se agora a criar e avaliar as soluções em termos do constrangimento de velocidade até que se obtenha uma população de 20 soluções e só após as 20 soluções que cumpram o constrangimento de velocidade são simuladas as soluções para validar o constrangimento da colisão. Com esta abordagem evita-se que se esteja a consumir recursos, com simulações de soluções que não cumpram com o constrangimento da velocidade.

Além destes dois tópicos, para a próxima abordagem foi definido introduzir dois contadores, um para cada tipo de penalização para que se possa obter uma ideia do comportamento do algoritmo em termos de que penalização ocorre assim como o seu tipo. Com a introdução destes contadores, vai também ser possível observar com maior detalhe o comportamento do algoritmo.

4.7. Sétima Abordagem/ Algoritmo VII

A sétima abordagem pretende essencialmente resolver os problemas descritos anteriormente, na sexta abordagem. Para isso, nesta abordagem foi considerado, tal como na quinta metodologia que as soluções seriam criadas de forma aleatória com base nos constrangimentos de precedência da Tabela 12, de forma a criar uma população que contenha 20 soluções que respeitem o constrangimento de velocidade máxima. Com esta abordagem pretende-se diminuir consideravelmente o tempo necessário para a criação da primeira população, sendo que essa população é posteriormente avaliada em termos de eventuais colisões e devidamente penalizada caso estas existam, pelo que será expetável que ao longo das gerações o valor da função objetivo diminua.

Utilizando o algoritmo descrito na sétima abordagem, foi realizada uma simulação com uma população de 20 soluções, ou cromossomas, tendo corrido até à 10ª geração, considerando uma cadência de 10 GPM, ou seja, um tempo de ciclo de 6s. Tendo a simulação corrido em cerca de 5000s, o que equivale a cerca de 1h25m, simulado num computador de características Core i7-4710 HQ com 8 Gb de memória.

A população inicial criada dada a sua dimensão elevada, encontra-se no Anexo A. O critério utilizado foi o de criar de forma aleatória todas as variáveis de projeto, mas de forma a que não ocorra a violação do constrangimento de velocidade, assim só quando existirem vinte soluções que não violem o constrangimento de velocidade é que será chamada de população criada, demonstrada na Tabela 15, e que será posteriormente testado o constrangimento de colisão.

Tabela 15 - População criada

Nº população	y1[°]		z1[°]		x1[°]		z2[°]		y2[°]		x2[°]		F. Objetivo
1	220	289	290	301	302	56	57	70	71	86	87	160	1,08
2	247	309	310	347	348	76	77	86	87	105	106	171	1,35
3	206	224	225	254	255	77	78	110	111	120	121	151	1,00
4	193	203	204	262	263	52	53	86	87	99	100	165	0,55
5	222	291	292	322	323	13	14	53	54	111	112	170	0,95
6	224	234	235	278	279	63	64	76	77	86	87	175	0,90
7	338	346	347	355	356	96	97	105	106	117	118	140	3,38
8	236	283	284	334	335	10	11	23	24	52	53	114	2,12
9	338	348	349	356	357	80	81	106	107	117	118	147	3,27
10	260	343	344	350	351	57	58	104	105	120	121	142	2,05
11	222	285	286	344	345	9	10	63	64	78	79	123	1,73
12	203	346	347	354	355	83	84	109	110	117	118	166	0,70
13	181	211	212	354	355	67	68	92	93	107	108	141	0,75
14	203	348	349	359	360	97	98	105	106	119	120	148	1,00
15	251	289	290	307	308	69	70	96	97	120	121	160	1,60
16	205	283	284	335	336	55	56	61	62	119	120	179	0,52
17	322	340	341	349	350	23	24	80	81	97	98	117	3,50
18	228	295	296	346	347	51	52	103	104	112	113	151	1,37
19	301	310	311	325	326	23	24	64	65	92	93	159	2,45
20	228	299	300	344	345	17	18	59	60	70	71	168	1,08

Analisado, por exemplo a primeira solução do Anexo A, em destaque a sombreado na Figura 55 que apresenta um extrato das 5 primeiras populações do Anexo A, pode-se verificar que o valor da função objetivo é da ordem de grandeza das centenas, pois no movimento Y2 onde a diferença entre o valor inicial 115 graus e o valor final 118 graus irá levar a uma diferença de apenas 3 graus. Esta diferença por sua vez, uma vez que a cadência é de 10 GPM, correspondente a um tempo de ciclo de 6s, assim o tempo correspondente aos 3 graus seria de apenas 0.05s e correspondentemente uma velocidade de 6,6 m/s, valor que claramente viola o constrangimento de velocidade máxima de 3m/s.

Nº população	y1[°]		z1[°]		x1[°]		z2[°]		y2[°]		x2[°]		F. Objetivo
1	318	346	347	353	354	102	103	114	115	118	119	166	261,67
2	263	277	278	289	290	104	105	114	115	116	117	153	191,67
3	220	289	290	301	302	56	57	70	71	86	87	160	1,08
4	273	314	315	331	332	83	84	107	108	110	111	146	220,00
5	343	344	345	359	360	22	23	47	48	87	88	175	288,33

Figura 55 - Estrato da população testada, disponível no Anexo A.

Após a obtenção da população inicial, essas vinte soluções serão posteriormente validadas em relação ao constrangimento de colisão. Na Tabela 16, encontra-se as vinte soluções que formam a população inicial após a validação de ausência de colisão.

Tabela 16 - População inicial

Nº população	y1[°]		z1[°]		x1[°]		z2[°]		y2[°]		x2[°]		F. Objetivo
1	220	289	290	301	302	56	57	70	71	86	87	160	1,08
2	247	309	310	347	348	76	77	86	87	105	106	171	1,35
3	206	224	225	254	255	77	78	110	111	120	121	151	10,00
4	193	203	204	262	263	52	53	86	87	99	100	165	5,50
5	222	291	292	322	323	13	14	53	54	111	112	170	0,95
6	224	234	235	278	279	63	64	76	77	86	87	175	9,00
7	338	346	347	355	356	96	97	105	106	117	118	140	33,83
8	236	283	284	334	335	10	11	23	24	52	53	114	2,12
9	338	348	349	356	357	80	81	106	107	117	118	147	32,67
10	260	343	344	350	351	57	58	104	105	120	121	142	20,50
11	222	285	286	344	345	9	10	63	64	78	79	123	1,73
12	203	346	347	354	355	83	84	109	110	117	118	166	7,00
13	181	211	212	354	355	67	68	92	93	107	108	141	7,50
14	203	348	349	359	360	97	98	105	106	119	120	148	10,00
15	251	289	290	307	308	69	70	96	97	120	121	160	16,00
16	205	283	284	335	336	55	56	61	62	119	120	179	5,17
17	322	340	341	349	350	23	24	80	81	97	98	117	3,50
18	228	295	296	346	347	51	52	103	104	112	113	151	1,37
19	301	310	311	325	326	23	24	64	65	92	93	159	2,45
20	228	299	300	344	345	17	18	59	60	70	71	168	1,08

Analisando a Tabela16, pode-se verificar que as soluções 3, 4, 6, 7, 9, 10, 12 a 16 em destaque a sombreado na tabela, foram penalizadas por violarem a condição de colisão.

Na Tabela 17, encontra-se a população após dez gerações sendo que neste caso apesar da população inicial não possuir nenhuma solução que violasse o constrangimento de velocidade máxima na população da décima geração existem duas soluções, em destaque a sombreado na tabela, que claramente apresentam uma violação deste constrangimento, pois o valor da função objetivo é da ordem de grandeza das centenas.

Tabela 17 - População após 10 gerações

Nº população	y1[°]		z1[°]		x1[°]		z2[°]		y2[°]		x2[°]		F. Objetivo
1	220	309	310	347	348	76	77	86	87	105	106	171	0,90
2	260	293	310	347	351	57	58	63	64	78	79	171	1,88
3	260	293	310	347	351	57	58	63	64	105	106	171	1,88
4	220	309	310	347	351	57	58	63	87	105	106	171	1,33
5	220	309	310	347	348	76	77	86	87	105	106	171	0,90
6	247	309	310	347	351	57	58	86	87	105	106	171	1,40
7	220	309	310	347	348	76	77	86	87	105	106	171	0,90
8	220	309	310	347	351	57	58	86	87	105	106	171	0,95
9	220	309	310	347	348	76	77	86	87	105	106	171	0,90
10	220	309	310	347	351	57	58	86	87	105	106	123	1,75
11	220	309	310	347	351	57	58	63	65	95	106	171	113,33
12	301	309	310	348	351	57	58	63	64	78	79	123	3,08
13	220	309	310	347	351	57	58	63	87	105	106	171	1,33
14	247	309	310	347	351	57	58	63	64	78	79	123	2,20
15	220	309	310	347	351	57	58	63	87	105	106	171	1,33
16	220	309	310	347	351	57	58	63	64	78	79	123	1,75
17	247	309	310	347	351	57	58	86	87	105	106	171	1,40
18	220	309	310	347	348	76	77	86	87	105	106	171	0,90
19	220	309	310	347	351	57	58	86	87	105	106	123	175,00
20	220	309	310	347	351	57	58	63	64	78	79	123	1,75

Durante o desenvolvimento do algoritmo genético, foram adicionados dois elementos contadores, com o intuito de contabilizar o número de vezes que uma determinada população seria penalizada em cada geração, em relação ao critério de velocidade máxima e ao critério referente à ausência de colisão.

Na Figura 56, está representada a evolução do número de penalizações devido à violação do constrangimento de velocidade máxima, que ocorreram em cada solução da população ao longo das dez gerações. Pode-se observar que após a criação da primeira população de vinte soluções válidas para o constrangimento de velocidade, o número de penalizações mantém-se perto de zero. Na primeira geração foi necessário a criação e avaliação de 84 soluções (conforme anexo A) para se obter as vinte soluções que respeitem o constrangimento da velocidade, sendo as restantes 64 soluções descartadas por não respeitarem esse constrangimento. Com a utilização desta metodologia o número de penalizações permanece estável sendo que apenas

nas gerações 7 e 10 ocorre uma e duas penalizações, respetivamente, sendo que este fenómeno se deve essencialmente à operação de mutação.

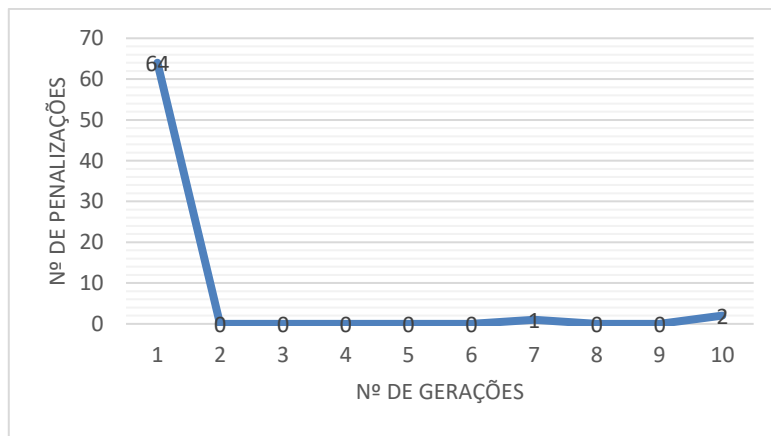


Figura 56 - Evolução do número de penalizações devido à violação do constrangimento da velocidade máxima

Na Figura 57, mostra-se a evolução do número de penalizações devido à violação do constrangimento de ausência de colisão, que ocorreram em cada solução da população ao longo das dez gerações. Aqui pode-se verificar que a primeira população criada apesar de respeitar completamente o constrangimento de velocidade máxima possui 11 soluções que não respeitam o constrangimento da ausência de colisão. Tal como seria espectável de um algoritmo de otimização o número de penalizações devido ao constrangimento da ausência de colisão vai diminuindo com o aumento da geração, vai convergindo.

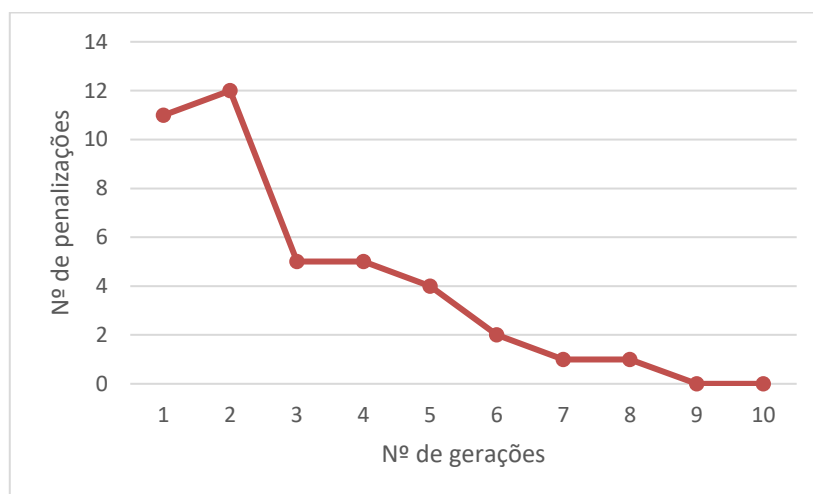


Figura 57 - Evolução do número de penalizações devido à violação do constrangimento da ausência de colisão

Analisando a tendência dos contadores de penalização apresentados nas Figuras 56 e 57, mas também o valor da função objetivo desde a população inicial, apresentada na Tabela 16 e a população da 10ª geração apresentada na Tabela 17, pode observar-se uma clara diminuição do valor da função objetivo assim como a diminuição clara das penalizações, tal como também se demonstra na Figura 57, onde se apresenta o valor da função objetivo da 10ª geração em comparação com os valores da função objetivo da população inicial. Analisando a Figura 57, pode verificar-se que tal como referido anteriormente existe uma melhoria bastante considerável do valor da função objetivo.

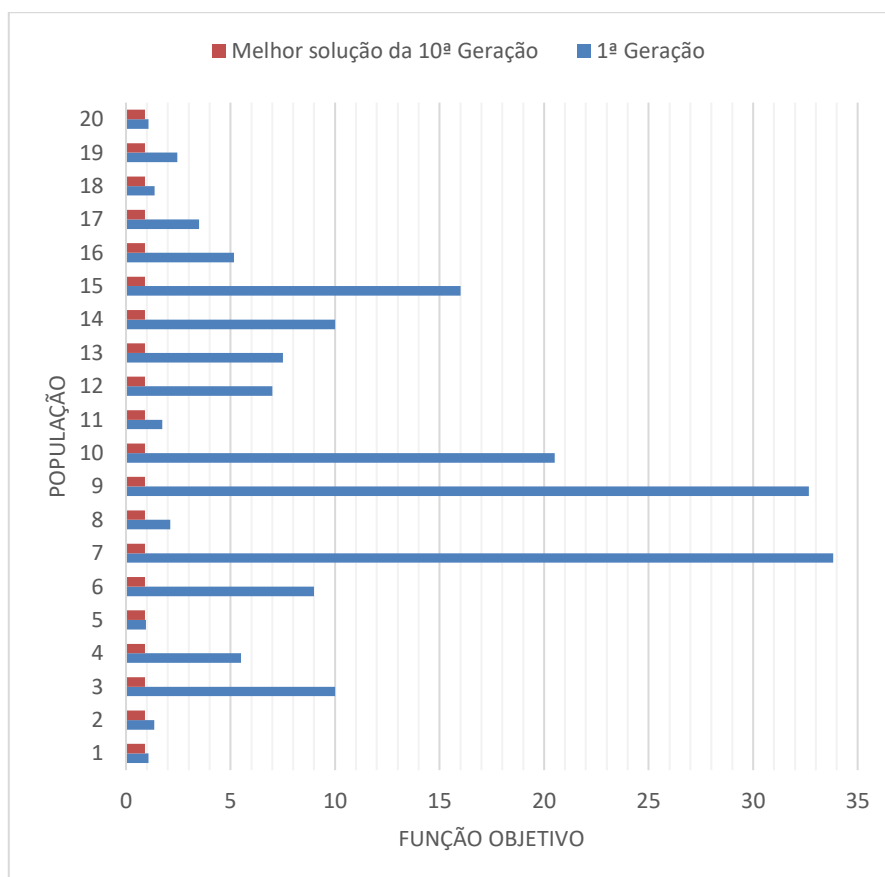


Figura 58 - Comparação do valor da função objetivo da melhor solução da 10ª geração em contraste com o as soluções da população inicial (1ª geração).

Em suma todos os resultados obtidos indicam que existiu uma melhoria considerável das soluções, ou seja, dos valores das variáveis de projeto em que os valores após o algoritmo correr durante 10 gerações cumprem os constrangimentos de

velocidade máxima e de ausência de colisão, à exceção de 2 soluções que não cumprem com o constrangimento de velocidade máxima.

Durante o desenvolvimento da metodologia de validação e otimização cinemática do sistema transfer, foram consideradas várias abordagens que culminaram no algoritmo VII. Este algoritmo cumpre os objetivos definidos para aplicação da metodologia em ambiente de produção de ferramentas transfer.

A implementação computacional desta metodologia em código VBA, está disponível no Anexo B.

Capítulo 5 – Conclusões

Neste TFM foi desenvolvida uma metodologia que, em conjunto com um modelo de CAD definido como modelo, tornou possível a simulação cinemática virtual de uma ferramenta transfer. Esta metodologia assenta essencialmente na utilização de algoritmos de otimização genéticos com interface ao modelo de simulação cinemática do sistema transfer implementado no software Ds CATIA. Com a metodologia desenvolvida, conseguiu-se realizar uma otimização de forma a reduzir o tempo de simulação cinemática cerca de 8 horas de trabalho manual (aproximadamente estimado 1 dia de trabalho) utilizando o módulo DMU kinematics para cerca de 1h25 com a licença base do software comercial Ds CATIA.

Como forma de desenvolver esta metodologia foram realizadas sete abordagens/algoritmos de forma a que se conseguisse obter o resultado pretendido.

Na primeira metodologia iniciou-se por se realizar o controlo de apenas uma das variáveis de projeto e utilizar a rotina interna solver do Microsoft Excel como fonte de otimização. A função objetivo pretendia o aumento do intervalo entre o ângulo de início (definido como variável de projeto) e o ângulo de chegada da curva de atuação da prensa para o movimento num dos eixos. Esta abordagem tornou-se imediatamente obsoleta pois não permitia iteração com o software para cada solução, ou seja, com esta metodologia não era possível determinar para cada solução o critério de ausência de colisão. Não era também possível interligar os movimentos em cada um dos 3 eixos xx, yy e zz, ou seja, esta função objetivo apenas conseguiria determinar a performance de uma solução por troços por eixo, sendo que esse não era o objetivo deste TFM.

Nas metodologias seguintes optou-se por utilizar a programação VBA, pois toda a informação se encontra dentro do software permitindo a avaliação de cada solução. Tendo-se partido da primeira metodologia, adota-se uma nova função objetivo onde se pretende a minimização do tempo de ciclo do transfer, em que a nova função objetivo tem em conta o tempo de todos os movimentos nos 3 eixos xx, yy e zz de avanço (1) e recuo (2), ao qual correspondem 6 troços (Y1, Z1, X1, Z2, Y2 e X2). Considerando para cada troço o ângulo de início e o ângulo de chegada da curva

de atuação da prensa, são definidas 12 variáveis de projeto. Com base neste princípio a segunda abordagem/metodologia tem como principal objetivo a implementação da nova função objetivo, que permita a detecção de possíveis colisões e que ao mesmo tempo garanta que a velocidade máxima durante o deslocamento seja inferior a 3 m/s considerando as características da prensa. Tendo-se conseguido atingir um dos principais objetivos com esta abordagem, foi, no entanto, obtida apenas uma solução válida que respeitava os constrangimentos. Foi definido o desenvolvimento de uma nova abordagem com um algoritmo que com um funcionamento similar ao apresentado nesta abordagem, produzisse em vez de uma, várias soluções que respeitem os constrangimentos. Na terceira abordagem/algoritmo o principal objetivo foi o desenvolvimento de um algoritmo que permitisse a criação das 12 variáveis de projeto, N vezes até que sejam criadas 20 soluções de 12 variáveis que cumpram com os constrangimentos que seria considerada a população inicial. Verificou-se que o tempo de computação poderia tornar-se crítico quando se introduzissem operadores genéticos no algoritmo, existindo a necessidade de explorar uma nova abordagem que permita a diminuição do tempo computacional necessário para obtenção da primeira população. Como forma de tentar diminuir o tempo computacional necessário para a criação das vinte soluções, foi criada uma quarta abordagem/algoritmo, onde se cria uma primeira solução que valida ambos os constrangimentos, sendo as soluções posteriores criadas tendo por base a primeira solução com uma tolerância associada. Contudo esta abordagem uma vez que cria as restantes soluções da população inicial com base na primeira solução obtida com uma diferença percentual em torno da mesma, leva a uma introdução de elitismo. Uma vez que não se verificaram quaisquer vantagens nesta abordagem foi considerada obsoleta e na quinta abordagem/algoritmo, a criação de uma população inicial de 20 soluções válidas, foi realizada conforme a terceira abordagem / algoritmo, acrescentando operações genéticas tais como a seleção, o crossover e a mutação. Foi concluído que seria necessário procurar uma abordagem que permita a criação das 12 variáveis de projeto, garantindo a precedência entre elas, isto é, que a sequência das variáveis correspondam ângulos exequíveis que sejam cumulativamente superiores aos ângulos das variáveis anteriores.

Assim, a sexta abordagem/ algoritmo definiu-se como principal objetivo a criação das 12 variáveis de projeto, garantindo a precedência entre elas, e para isso foi necessário definir balizamentos que foram definidos com base nos constrangimentos físicos existentes na prensa. Os objetivos para esta abordagem foram atingidos e foram também retirados alguns pontos de melhoria incluindo a necessária redução do tempo computacional de simulação para a obtenção de resultados.

Na sétima abordagem/ algoritmo foram adicionados alguns constrangimentos para verificar se as duas soluções que irão sofrer a operação de crossover, continuam a verificar as precedências e também uma nova forma de criar e avaliar a população inicial, definindo-se assim que primeiramente se criava 20 soluções que respeitem o constrangimento da velocidade e só após se obterem as 20 soluções é que estas eram avaliadas em termos de possíveis colisões. Esta abordagem permitiu uma diminuição considerável do tempo necessário para a realização de uma simulação assim como o tempo consumido para obter valores otimizados.

A análise de resultados permite verificar que após 10 gerações, o modelo já não apresenta penalizações do constrangimento da ausência de colisões, e apresenta apenas duas ocorrências da penalização do constrangimento da velocidade máxima. Apresenta também uma solução predominante com o valor de função objetivo mais baixo, 0.9 segundos, tendo sido por isso considerada como a solução ótima.

Com a utilização da metodologia apresentada neste TFM, foi possível simular todas as possibilidades dos vários valores possíveis para as 12 variáveis de projeto em cerca de 5000s, o que equivale a cerca de 1h25m. Assim foi possível definir quais os valores ótimos para as 12 variáveis de projeto que conseguem garantir um caminho livre de colisões, utilizando o menor tempo de ciclo garantindo uma elevada cadência, podendo assim considerar-se que o principal objetivo deste TFM foi atingido.

Capítulo 6 - Trabalhos Futuros

Uma vez que foram consideradas as velocidades nos três eixos como constante e com um valor máximo de 3 m/s, um trabalho futuro passará por criar um modelo matemático que permita descrever realmente qual é a velocidade real do sistema, incluindo os fenômenos de aceleração e desaceleração, bem como um constrangimento de aceleração máxima. Poderá também ser estudada a possibilidade de realizar uma otimização inicial dos valores ótimos para as variáveis de projeto que minimizam o Jerk, derivada de 1ª ordem da aceleração em ordem ao tempo. Com a minimização do Jerk, tal como vários autores descrevem, consegue-se obter uma movimentação mais linear e sem que ocorram fenômenos como a vibração residual que podem levar a um desgaste prematuro das ferramentas e a uma redução da qualidade das tolerâncias nos processos de fabricação.

Bibliografia

- [1] J. Rodrigues and P. Martins, *Tecnologia Mecânica - Tecnologia da deformação Plástica - Vol. II*, 2nd ed. Editora, Escolar, 2010.
- [2] K. Lange, "Handbook of Metal Forming." Society of Manufacturing Engineers, Michigan, 1994.
- [3] M. Samuel, "Influence of drawbead geometry on sheet metal forming," *J. Mater. Process. Technol.*, vol. 122, no. 1, pp. 94–103, 2002.
- [4] M. F. Handbook and S. B. Heidelberg, *Metal Forming Handbook*, vol. 3, no. c. 1998.
- [5] "<https://www.autoform.com/en/glossary/progressive-die-stamping/> - visitado a 13/02/2018." .
- [6] S. L. Semiatin, *ASM Handbook Vol. 14b: Metalworking: Sheet Forming*, vol. 14. 2006.
- [7] I. Suchy, *HandBook of Die Design*, 2ª Edição. McGraw-Hill, 2006.
- [8] J. April, M. Better, F. Glover, and J. Kelly, "New Advances and Applications for Marrying Simulation and Optimization," *Winter Simul.*, 2004.
- [9] R. Goor, "New coordination algorithms improve the throughput of a GM press line," *Int. Robot. Vis. Conf.*, 2003.
- [10] B. T. Lin and C. C. Kuo, "Application of an integrated CAD/CAE/CAM system for stamping dies for automobiles," *Int. J. Adv. Manuf. Technol.*, vol. 35, no. 9–10, pp. 1000–1013, 2008.
- [11] E. Emel and H. B. E. Tasci, "Optimization approach for scheduling of jobs on lines of press machines," *Ind. Eng. Res. - Conf. Proc.*, 1997.
- [12] S. A. Ozalp, "A genetic algorithm for scheduling of jobs on lines of press machines," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3743 LNCS, pp. 535–543, 2006.
- [13] S. Kuk, S. I. Soh, S. M. Lim, S. H. Joung, and S. Do Noh, "Construction and

Application of a Virtual Press Shop," *Imecs 2009 Int. Multi-Conference Eng. Comput. Sci. Vols I li*, pp. 1776–1781, 2009.

- [14] H. Hoffmann and M. Kohnhäuser, "Strategies to optimize the part transport in crossbar transfer presses," *CIRP Ann. - Manuf. Technol.*, vol. 51, no. 1, pp. 27–32, 2002.
- [15] J. a García-sedano and J. A. Bernardo, "Stamping line optimization using Genetic Algorithms and Virtual 3D Stamping line optimization using Genetic Algorithms and Virtual 3D line," pp. 319–326, 2010.
- [16] B. Svensson, F. Danielsson, and B. Lennartson, "An efficient algorithm for press line optimisation," pp. 1627–1638, 2013.
- [17] J. H. Holland, "Adaptation in Natural and Artificial Systems: An introductory Analysis with Applications to Biology, Control and Artificial Intelligence," *MIT Press*, p. 183, 1975.
- [18] K. Jebari and M. Madiafi, "Selection Methods for Genetic Algorithms," vol. 3, no. December, pp. 333–344, 2013.
- [19] D. E. Golberg, "Genetic Algorithms in Search Optimization & Machine Learning." p. 412, 1989.
- [20] D. A. Coley, "An Introduction to Genetic Algorithms for Scientists and Engineers."
- [21] D. Beasley, D. R. Bull, and R. R. Martin, "An overview of genetic algorithms : Part 1, fundamentals," *Univ. Comput.*, vol. 2, no. 15, pp. 1–16, 1993.
- [22] M. Gen and R. Cheng, "Genetic Algorithms and Engineering Design," pp. 1–65, 1997.
- [23] J. Carr, "An Introduction to Genetic Algorithms," pp. 1–40, 2014.
- [24] T. M. Mitchell, *Machine Learning* .
- [25] D. Hermawanto, "Genetic Algorithm for Solving Simple Mathematical Equality Problem," 2013.

Anexo A - População testada na 7ª Abordagem/algorithm

Nº população	y1		z1		x1		z2		y2		x2		F. Objetivo
1	318	346	347	353	354	102	103	114	115	118	119	166	261,67
2	263	277	278	289	290	104	105	114	115	116	117	153	191,67
3	220	289	290	301	302	56	57	70	71	86	87	160	1,08
4	273	314	315	331	332	83	84	107	108	110	111	146	220,00
5	343	344	345	359	360	22	23	47	48	87	88	175	288,33
6	247	309	310	347	348	76	77	86	87	105	106	171	1,35
7	216	297	298	327	328	31	32	107	108	109	110	114	178,33
8	277	329	330	341	342	27	28	61	62	118	119	131	251,67
9	195	260	261	307	308	61	62	83	84	115	116	125	125,00
10	355	357	358	359	360	45	46	103	104	110	111	140	366,67
11	217	275	276	344	345	9	10	100	101	112	113	119	171,67
12	260	303	304	358	359	72	73	116	117	120	121	166	165,00
13	297	308	309	329	330	32	33	86	87	106	107	108	323,33
14	254	347	348	357	358	78	79	114	115	117	118	149	183,33
15	259	280	281	312	313	99	100	114	115	117	118	140	206,67
16	247	332	333	356	357	4	5	99	100	109	110	153	165,00
17	206	224	225	254	255	77	78	110	111	120	121	151	1,00
18	193	203	204	262	263	52	53	86	87	99	100	165	0,55
19	350	356	357	359	360	20	21	42	43	90	91	105	416,67
20	222	291	292	322	323	13	14	53	54	111	112	170	0,95
21	331	332	333	340	341	93	94	99	100	116	117	133	338,33
22	355	356	357	359	360	25	26	35	36	74	75	89	451,67
23	279	299	300	327	328	67	68	93	94	113	114	118	276,67
24	332	333	334	335	336	26	27	89	90	108	109	119	363,33
25	283	299	300	320	321	82	83	113	114	118	119	135	255,00
26	210	237	238	257	258	29	30	115	116	117	118	132	138,33
27	193	290	291	331	332	103	104	114	115	119	120	155	71,67
28	351	355	356	357	358	10	11	90	91	106	107	173	305,00
29	224	234	235	278	279	63	64	76	77	86	87	175	0,90
30	338	346	347	355	356	96	97	105	106	117	118	140	3,38
31	284	353	354	355	356	109	110	118	119	120	121	129	266,67
32	285	288	289	340	341	108	109	110	111	120	121	133	261,67
33	343	357	358	359	360	45	46	87	88	89	90	149	331,67
34	313	316	317	346	347	93	94	99	100	107	108	117	335,00
35	309	340	341	353	354	72	73	106	107	113	114	134	300,00
36	328	341	342	358	359	77	78	117	118	119	120	126	345,00
37	261	352	353	356	357	11	12	96	97	114	115	146	200,00
38	335	351	352	356	357	91	92	118	119	120	121	159	301,67
39	322	341	342	344	345	58	59	103	104	105	106	154	288,33
40	355	357	358	359	360	35	36	42	43	45	46	166	323,33
41	318	347	348	350	351	68	69	72	73	82	83	93	383,33
42	236	283	284	334	335	10	11	23	24	52	53	114	2,12

42	236	283	284	334	335	10	11	23	24	52	53	114	2,12
43	219	328	329	332	333	53	54	108	109	112	113	120	173,33
44	338	348	349	356	357	80	81	106	107	117	118	147	3,27
45	320	355	356	358	359	90	91	112	113	114	115	124	335,00
46	213	217	218	290	291	93	94	110	111	115	116	117	168,33
47	260	343	344	350	351	57	58	104	105	120	121	142	2,05
48	189	325	326	344	345	8	9	64	65	91	92	94	166,67
49	213	289	290	308	309	39	40	105	106	111	112	149	115,00
50	256	336	337	354	355	20	21	78	79	82	83	171	150,00
51	222	285	286	344	345	9	10	63	64	78	79	123	1,73
52	325	349	350	353	354	38	39	115	116	118	119	146	306,67
53	203	346	347	354	355	83	84	109	110	117	118	166	0,70
54	193	227	228	273	274	108	109	111	112	114	115	146	86,67
55	275	324	325	340	341	107	108	111	112	113	114	164	193,33
56	346	357	358	359	360	109	110	111	112	115	116	130	368,33
57	337	340	341	353	354	75	76	83	84	96	97	144	330,00
58	271	348	349	354	355	112	113	115	116	120	121	139	228,33
59	272	307	308	347	348	51	52	110	111	117	118	165	186,67
60	252	288	289	329	330	103	104	107	108	112	113	136	201,67
61	273	356	357	359	360	53	54	86	87	91	92	113	275,00
62	181	211	212	354	355	67	68	92	93	107	108	141	0,75
63	266	285	286	311	312	55	56	106	107	111	112	132	231,67
64	203	348	349	359	360	97	98	105	106	119	120	148	1,00
65	331	345	346	350	351	80	81	100	101	108	109	151	308,33
66	214	338	339	357	358	71	72	101	102	105	106	128	151,67
67	275	282	283	318	319	7	8	66	67	114	115	122	263,33
68	251	289	290	307	308	69	70	96	97	120	121	160	1,60
69	321	354	355	359	360	8	9	52	53	81	82	162	273,33
70	225	283	284	351	352	62	63	69	70	90	91	93	228,33
71	205	283	284	335	336	55	56	61	62	119	120	179	0,52
72	333	357	358	359	360	63	64	101	102	107	108	164	290,00
73	339	343	344	349	350	40	41	55	56	99	100	117	378,33
74	322	340	341	349	350	23	24	80	81	97	98	117	3,50
75	350	352	353	359	360	96	97	110	111	114	115	169	310,00
76	228	295	296	346	347	51	52	103	104	112	113	151	1,37
77	344	354	355	358	359	62	63	112	113	119	120	136	355,00
78	301	310	311	325	326	23	24	64	65	92	93	159	2,45
79	294	337	338	357	358	6	7	113	114	118	119	168	218,33
80	211	294	295	355	356	96	97	111	112	114	115	141	125,00
81	325	351	352	359	360	70	71	93	94	110	111	124	343,33
82	256	299	300	343	344	93	94	115	116	119	120	131	216,67
83	356	357	358	359	360	24	25	48	49	95	96	112	415,00
84	228	299	300	344	345	17	18	59	60	70	71	168	1,08

Anexo B – Código VBA utilizado na 7ª Abordagem/algorithm

```
Public genNumber As Integer

Dim InitialPopulation(), Population(), delta(),
Newdelta(), OffSpring(), time(), Newtime(),
velocity(), Fobj(), fit(), newFobj(),
Chromosome(), fitness(), min(11), max(11) As
Double

Dim Probability(), Dist(2), CumulativeProb(),
roleteWheel(), SelectedPopulation(), Ranking(),
Parents(), C(), D(), r(), mutposi(), mutposj(), F(),
H() As Double

Dim A(), B(), folha As String

Dim popSize, GPM, cycleTime, Angulo,
ConflictsCount, CrossoverCount As Integer

Dim StartTime, SecondsElapsed, crossRate,
mutationRate, total, val, varpercent As Double

Dim Param(16) As Object

Dim documents1 As Documents

Dim partDocument1 As PartDocument

Dim part1 As Part

Dim ColisonResult() As Double

Dim Product As Products

Dim RootProd As Product

Dim parameters1 As Object

Dim cClashes As Clashes

Dim Clash1, Clash2, Clash3, Clash4, Clash5,
Clash6 As Object

Dim cConflicts(6) As Conflicts

Dim Conflits(5) As Double

Dim ClashNumber, ElitismNumber As Double

Dim ClashResult(6), Elistm() As Double

Dim Num, pop As Double

Dim oConflicts As Conflicts

Dim aStatus As CatConflictStatus

Dim ResultsArray(100, 360, 7)

Dim Colision As Boolean

Sub Main()

    StartTime = Timer

    Dim i, j, k As Integer

    genNumber = 1

    Call Values

    Call CreateInitialPopulation

    ReDim ColisonResult(100, popSize)

    Call ExportResults1

    Do Until genNumber = 10

        If genNumber = 1 Then

            Call Selection

            Call EraseParam

            Call ExportResults2

            Call Crossover

            'Call Mutation

            Call ExportResults3

            genNumber = genNumber + 1

        Else

            Call Values

            Call ReSelection

            Call EraseParam

            Call ReCrossover

            'Call Mutation

            genNumber = genNumber + 1

        End If

    End Do

End Sub
```

```

    'genNumber = genNumber + 1
Loop
Call ExportResults4
Call ExportResults5
SecondsElapsed = Round(Timer - StartTime,
2)
MsgBox "This code ran successfully in " &
SecondsElapsed & " Seconds", vbInformation
End Sub
Sub CreateInitialPopulation()
Dim i, j, k, w, C, max, maxx As Integer
w = 0
popSize = 20
ElitismNumber = Round(0.1 * popSize)
ReDim InitialPopulation(popSize, 11),
Fobj(popSize), fit(popSize), delta(5, popSize),
time(5, popSize), velocity(5, popSize),
Ranking(popSize), Parents(popSize),
OffSpring(popSize, 11)
CreateChromossome:
For i = 0 To popSize - 1
'y1_inicio
y1_inicio:
InitialPopulation(i, 0) =
FstChromossomeCreate(180, 356)
'Debug.Print "y1_inicio:" & Chromossomes(i,
0)
'y1_fim
min(1) = InitialPopulation(i, 0) + 1
InitialPopulation(i, 1) =
FstChromossomeCreate(min(1), 357)
'Debug.Print "y1_fim:" & Chromossomes(i, 1)
'z1_inicio
InitialPopulation(i, 2) = InitialPopulation(i, 1)
+ 1

```

```

'Debug.Print "z1_inicio:" & Chromossomes(i,
2)
'y1_fim
min(3) = InitialPopulation(i, 2) + 1
InitialPopulation(i, 3) =
FstChromossomeCreate(min(3), 359)
'Debug.Print "z1_fim:" & Chromossomes(i, 3)
'x1_inicio
InitialPopulation(i, 4) = InitialPopulation(i, 3)
+ 1
'Debug.Print "x1_inicio:" & Chromossomes(i,
4)
'x1_fim
min(4) = 1
InitialPopulation(i, 5) =
FstChromossomeCreate(min(4), 116)
'Debug.Print "x1_fim:" & Chromossomes(i, 5)
'z2_inicio
InitialPopulation(i, 6) = InitialPopulation(i, 5)
+ 1
'Debug.Print "z2_inicio:" & Chromossomes(i,
6)
'z2_fim
min(7) = InitialPopulation(i, 6) + 1
InitialPopulation(i, 7) =
FstChromossomeCreate(min(7), 118)
'Debug.Print "z2_fim:" & Chromossomes(i, 7)
'y2_inicio
InitialPopulation(i, 8) = InitialPopulation(i, 7)
+ 1
'Debug.Print "y2_inicio:" & Chromossomes(i,
8)
'y2_fim
min(9) = InitialPopulation(i, 8) + 1
InitialPopulation(i, 9) =
FstChromossomeCreate(min(9), 120)

```

```

'Debug.Print "y2_fim:" & Chromossomes(i, 9)

'x2_inicio

InitialPopulation(i, 10) = InitialPopulation(i, 9)
+ 1

'Debug.Print "x2_inicio:" & Chromossomes(i,
10)

'x2_fim

min(11) = InitialPopulation(i, 10) + 1

InitialPopulation(i, 11) =
FstChromossomeCreate(min(11), 179)

'Debug.Print "x2_fim:" & Chromossomes(i,
11)

'Debug.Print "-----
"

Next i

For C = 0 To popSize - 1

j = 0

For i = 0 To 5

k = j + 1

delta(i, C) = EvalfAngle(InitialPopulation(C,
j), InitialPopulation(C, k))

time(i, C) = EvalfTime(delta(i, C))

If i = 0 Or i = 4 Then

velocity(i, C) = 330 / time(i, C)

Elseif i = 1 Or i = 3 Then

velocity(i, C) = 245 / time(i, C)

Elseif i = 2 Or i = 5 Then

velocity(i, C) = 925 / time(i, C)

End If

'Debug.Print delta(i, c)

'Debug.Print "time(" & i & ", " & c & "):" &
time(i, c)

'Debug.Print velocity(i, c)

j = j + 2

Next i

```

```

For i = 0 To 5

If velocity(i, C) >= 3000 Then

Fobj(C) = EvalfObjective(C) * 10

Else

Fobj(C) = EvalfObjective(C)

End If

Next i

fit(C) = Abs(cycleTime - Fobj(C))

'If fit(C) < 0 Then

' fit(C) = Abs(fit(C) * 10)

'End If

'Debug.Print Fobj(c)

Next C

For C = 0 To popSize - 1

If fit(C) > max Then

max = fit(C)

Elseif fit(C) > maxx And fit(C) < max Then

maxx = fit(C)

End If

Next C

End Sub

Sub Selection()

ReDim Probability(popSize),
Elistm(ElitismNumber),
CumulativeProb(popSize),
roleteWheel(popSize),
SelectedPopulation(popSize, 11),
Ranking(popSize), Parents(popSize),
fitness(popSize)

Dim i, w, j, k, C, max, maxx As Integer

total = 0

For i = 0 To popSize - 1

fitness(i) = (1 / (1 + fit(i)))

total = total + fitness(i)

```

```

Next i
For i = 0 To ElitismNumber
    Next i
    For i = 0 To popSize - 1
        Probability(i) = Round((fitness(i) / total), 4)
        'Debug.Print "Probability(" & i & "):" & Probability(i)
    Next

    For i = 0 To popSize - 1
        If i = 0 Then
            CumulativeProb(i) = Round(Probability(i), 4)
        Else
            CumulativeProb(i) = Round(Probability(i), 4) + CumulativeProb(i - 1)
        End If
        'Debug.Print "CumulativeProb(" & i & "):" & CumulativeProb(i)
        roleteWheel(i) = Round(Rolete(0, 1), 4)
        'Debug.Print "roleteWheel (" & i & "):" & roleteWheel(i)
    Next i
    k = 0
    Dim Number As Double
    Do Until k = popSize
        Number = Round(Rolete(0, 1), 4)
        'Debug.Print Number
        For i = 0 To popSize - 1
            If CumulativeProb(i) < Number And Number < CumulativeProb(i + 1) Then
                'Debug.Print i
                Parents(k) = i
            Else CumulativeProb(0) > Number Then
                'Debug.Print 1
                Parents(k) = 1
            End If
        Next i
        k = k + 1
    Loop
    Debug.Print "-----"
    For i = 0 To popSize - 1
        Debug.Print Parents(i)
    Next i
    For i = 0 To popSize - 1
        For w = 0 To 11
            SelectedPopulation(i, w) = InitialPopulation(Parents(i), w)
        Next w
    Next i
    For C = 0 To popSize - 1
        j = 0
        For i = 0 To 5
            k = j + 1
            delta(i, C) = EvalfAngle(SelectedPopulation(C, j), SelectedPopulation(C, k))
            time(i, C) = EvalfTime(delta(i, C))
            If i = 0 Or i = 4 Then
                velocity(i, C) = 330 / time(i, C)
            Else i = 1 Or i = 3 Then
                velocity(i, C) = 245 / time(i, C)
            Else i = 2 Or i = 5 Then
                velocity(i, C) = 925 / time(i, C)
            End If
        Next i
    Next C

```

```

'Debug.Print delta(i, c)
'Debug.Print "time(" & i & ", " & c & "):" &
time(i, c)
'Debug.Print velocity(i, c)
j = j + 2
Next i
For i = 0 To 5
  If velocity(i, C) >= 3000 Then
    Fobj(C) = EvalfObjective(C) * 10
  Else
    Fobj(C) = EvalfObjective(C)
  End If
Next i
Dim RootProduct As Product
Set RootProduct = CATIA.ActiveDocument.Product
Angulo = 0
ConflictsCount = 0
Colision = False
Call AttributesActualization(C)
Do Until Angulo > 360 Or Colision = True
  Param(0).Value = Angulo
  'Debug.Print "Angulo:" & Angulo
  RootProduct.Update
  Call ClashCreate
  For k = 1 To 2
    If cConflicts(k).Count <> 0 Then
      ConflictsCount = ConflictsCount +
cConflicts(k).Count
    End If
  Next k
  If ConflictsCount <> 0 Then
    Colision = True
  End If
  Call ClashErase
  GoTo ClashObjective
Else
  Call ClashErase
End If
Angulo = Angulo + 10
Loop
ClashObjective:
If Colision = True Then
  Fobj(C) = EvalfObjective(C) * 10
Else
  Fobj(C) = EvalfObjective(C)
End If
fit(C) = Abs(cycleTime - Fobj(C))
'If fit(C) < 0 Then
'  fit(C) = Abs(fit(C) * 10)
'End If
'For i = 0 To 5
'  If velocity(i, C) = 3000 Then
'    fit(C) = Abs(fit(C)) * 10
'  End If
'Next i
'Debug.Print Fobj(c)
Next C
For C = 0 To popSize - 1
  If fit(C) > max Then
    max = fit(C)
  ElseIf fit(C) > maxx And fit(C) < max Then
    maxx = fit(C)
  End If
  If Colision = True Then

```

```

        ColisionResult(genNumber, C) = 1
    Else
        ColisionResult(genNumber, C) = 0
    End If
Next C
End Sub
Sub Crossover()
    Dim i, w, j, k, s, Num, A, B, D, E As Integer
    k = 0
    i = 0
    While k < popSize
        Ranking(k) = Rolete(0, 1)
        If Ranking(k) < crossRate Then
            Parents(Num) = k
            Num = Num + 1
            'Debug.Print k
        End If
        'Parents(Num) = k
        k = k + 1
    Wend
    'Debug.Print "-----"
    "
    'For i = 0 To Num - 1
        'Debug.Print Parents(i)
    'Next i
    'Debug.Print Num
    'Debug.Print "-----"
    "
    ReDim C(Num)
    For j = 0 To Num - 1
        C(j) = randomNumber(1, 12)
        'Debug.Print "Crossover point(" & j & ")" &
        " In the generation(" & genNumber & "):" & C(j)

```

```

    Next j
    CrossoverCount = 0
    For j = 0 To popSize - 1
        For i = 0 To 11
            OffSpring(j, i) = SelectedPopulation(j, i)
        Next i
    Next j
    For j = 0 To Num - 1
        A = Parents(j)
        B = C(j)
        D = C(j) - 1
        E = C(j) + 1
        If C(j) = 12 Then
            GoTo NextParent
        End If
        If SelectedPopulation(A, D) <
        SelectedPopulation(A + 1, B) Then
            CrossoverCount = CrossoverCount + 1
            For i = 0 To 11
                If j < Num - 1 Then
                    If i < C(j) Then
                        'Debug.Print "Chromossomes(" &
                        Parents(j) & "," & i & "):" &
                        Chromossomes(Parents(j), i)
                        OffSpring(Parents(j), i) =
                        SelectedPopulation(Parents(j), i)
                    Elseif i >= C(j) Then
                        'Debug.Print "Chromossomes(" &
                        Parents(j) & "," & i & "):" &
                        Chromossomes(Parents(j + 1), i)
                        OffSpring(Parents(j), i) =
                        SelectedPopulation(Parents(j + 1), i)
                    End If
                Elseif j = Num - 1 Then
                    If i < C(j) Then

```

```

        OffSpring(Parents(j), i) = SelectedPopulation(Parents(j), i)
    Elseif i >= C(j) Then
        OffSpring(Parents(j), i) = SelectedPopulation(Parents(0), i)
    End If
End If
Next i
End If
NextParent:
Next j
End Sub

Sub Mutation()
    Dim totalGene As Integer
    Dim i, j, k, C, max, maxx As Integer
    Dim mutationNum As Double
    Dim C1 As String
    totalGene = 12 * popSize
    mutationNum = Round(totalGene * mutationRate, 0)
    'Debug.Print "MUTATION RATE:" & mutationRate
    If mutationNum >= 1 Then
        ReDim r(mutationNum - 1), mutposi(mutationNum - 1), mutposj(mutationNum - 1), H(mutationNum - 1), F(mutationNum - 1)
        For i = 0 To mutationNum - 1
            r(i) = mutRandNum(1, totalGene)
            'Debug.Print " Valor de R(" & i & "):"; R(i)
        Next i
    End If
    MutationEvaluate:
    If i <> 0 Then
        If r(i) = r(i - 1) Then
            r(i) = mutRandNum(1, totalGene)
            GoTo MutationEvaluate
        End If
        H(i) = r(i) / 12
        mutposi(i) = CInt(r(i) / 12)
        'Debug.Print H(i) - mutposi(i)
        If H(i) - mutposi(i) > 0 Then
            mutposi(i) = CInt(r(i) / 12)
        Elseif H(i) - mutposi(i) = 0 Then
            mutposi(i) = CInt(r(i) / 12) - 1
        Elseif H(i) - mutposi(i) < 0 Then
            mutposi(i) = (CInt(r(i) / 12)) - 1
        End If
        'Debug.Print mutposi(i)
        F(i) = (12 * mutposi(i)) + 1
        mutposj(i) = r(i) - F(i)
        'Debug.Print mutposj(i)
    Next
    For i = 0 To mutationNum - 1
        'Debug.Print "OLD:" & Chromossomes(mutposi(i), mutposj(i))
        OffSpring(mutposi(i), mutposj(i)) = MutationChromoCreate(mutposi(i), mutposj(i))
        'Debug.Print "NEW:" & Chromossomes(mutposi(i), mutposj(i))
    Next i
    End If
    For C = 0 To popSize - 1
        j = 0
        For i = 0 To 5
            k = j + 1
            delta(i, C) = EvalfAngle(OffSpring(C, j), OffSpring(C, k))
            time(i, C) = EvalfTime(delta(i, C))
        Next i
    Next C
End Sub

```

```

If i = 0 Or i = 4 Then
    velocity(i, C) = 330 / time(i, C)
Elseif i = 1 Or i = 3 Then
    velocity(i, C) = 245 / time(i, C)
Elseif i = 2 Or i = 5 Then
    velocity(i, C) = 925 / time(i, C)
End If

'Debug.Print delta(i, c)

'Debug.Print "time(" & i & ", " & c & "):" &
time(i, c)

'Debug.Print velocity(i, c)

j = j + 2
Next i
Fobj(C) = EvalfObjective(C)
fit(C) = Abs(cycleTime - Fobj(C))

'If fit(C) < 0 Then
'  fit(C) = Abs(fit(C) * 10)
'End If

'For i = 0 To 5
'  If velocity(i, C) = 3000 Then
'    fit(C) = Abs(fit(C)) * 10
'  End If
'Next i

'Debug.Print Fobj(c)
Next C
For C = 0 To popSize - 1
    If fit(C) > max Then
        max = fit(C)
    Elseif fit(C) > maxx And fit(C) < max Then
        maxx = fit(C)
    End If
Next C

```

```

End Sub

Sub ClashCreate()
    '----- CLASH DEFINING -----

    Set RootProd =
CATIA.ActiveDocument.Product

    Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologic
alObject("Clashes")

    Set Clash1 = cClashes.Item("UPPER TOOL vs
TRANSFER")

    Set Clash2 = cClashes.Item("LOWER TOOL vs
TRANSFER")

    'Set Clash3 = cClashes.Item("TRANSFER vs
ESTRUTURA PRENSA")

    'Set Clash4 = cClashes.Item("TRANSFER vs
STRIP LAYOUT")

    'Set Clash5 = cClashes.Item("UPPER TOOL vs
STRIP LAYOUT")

    'Set Clash6 = cClashes.Item("STRIP LAYOUT
VS ESTUTURA PRENSA")

    '----- CLASH COMPUTING -----

    Clash1.Compute
    Clash2.Compute
    'Clash3.Compute
    'Clash4.Compute
    'Clash5.Compute
    'Clash6.Compute

    Set cConflicts(1) = Clash1.Conflicts
    Set cConflicts(2) = Clash2.Conflicts

    'Set cConflicts(3) = Clash3.Conflicts
    'Set cConflicts(4) = Clash4.Conflicts
    'Set cConflicts(5) = Clash5.Conflicts
    'Set cConflicts(6) = Clash6.Conflicts

End Sub

Sub ClashErase()

    Set RootProd = Nothing

```

```

Set cClashes = Nothing

Set Clash1 = Nothing

Set Clash2 = Nothing

Set cConflicts(1) = Nothing

Set cConflicts(2) = Nothing

End Sub

Sub ReSelection()

    ReDim                Probability(popSize),
Population(popSize,    11),
Elistm(ElitismNumber),
CumulativeProb(popSize),
roleteWheel(popSize), Ranking(popSize),
Parents(popSize), fitness(popSize)

    Dim i, w, j, k, C, max, maxx As Integer

    total = 0

    For j = 0 To popSize - 1

        For i = 0 To 11

            Population(j, i) = OffSpring(j, i)

        Next i

    Next j

    For i = 0 To popSize - 1

        fitness(i) = (1 / (1 + fit(i)))

        total = total + fitness(i)

    Next i

    For i = 0 To ElitismNumber

        Next i

    For i = 0 To popSize - 1

        Probability(i) = Round((fitness(i) / total), 4)

        'Debug.Print "Probability(" & i & "):" &
Probability(i)

    Next

    For i = 0 To popSize - 1

```

```

        If i = 0 Then

            CumulativeProb(i) = Round(Probability(i),
4)

        Else

            CumulativeProb(i) = Round(Probability(i),
4) + CumulativeProb(i - 1)

        End If

        'Debug.Print "CumulativeProb(" & i & "):"
& CumulativeProb(i)

        roleteWheel(i) = Round(Rolete(0, 1), 4)

        'Debug.Print "roleteWheel (" & i & "):" &
roleteWheel(i)

    Next i

    k = 0

    Dim Number As Double

    Do Until k = popSize

        Number = Round(Rolete(0, 1), 4)

        'Debug.Print Number

        For i = 0 To popSize - 1

            If CumulativeProb(i) < Number And
Number < CumulativeProb(i + 1) Then

                'Debug.Print i

                Parents(k) = i

            ElseIf CumulativeProb(0) > Number Then

                'Debug.Print 1

                Parents(k) = 1

            Exit For

        End If

    Next i

    k = k + 1

Loop

Debug.Print "-----"

For i = 0 To popSize - 1

    Debug.Print Parents(i)

```

```

Next i
For i = 0 To popSize - 1
    For w = 0 To 11
        SelectedPopulation(i, w) = Population(Parents(i), w)
    Next w
Next i
For C = 0 To popSize - 1
    j = 0
    For i = 0 To 5
        k = j + 1
        delta(i, C) = EvalfAngle(SelectedPopulation(C, SelectedPopulation(C, k)), j),
        time(i, C) = EvalfTime(delta(i, C))
        If i = 0 Or i = 4 Then
            velocity(i, C) = 330 / time(i, C)
        ElseIf i = 1 Or i = 3 Then
            velocity(i, C) = 245 / time(i, C)
        ElseIf i = 2 Or i = 5 Then
            velocity(i, C) = 925 / time(i, C)
        End If
        'Debug.Print delta(i, c)
        'Debug.Print "time(" & i & "," & c & "):" & time(i, c)
        'Debug.Print velocity(i, c)
        j = j + 2
    Next i
    For i = 0 To 5
        If velocity(i, C) >= 3000 Then
            Fobj(C) = EvalfObjective(C) * 10
        Else
            Fobj(C) = EvalfObjective(C)
        End If
    Next i
End If
Next i
Dim RootProduct As Product
Set RootProduct = CATIA.ActiveDocument.Product
Angulo = 0
ConflictsCount = 0
Colision = False
Call AttributesActualization(C)
Do Until Angulo > 360 Or Colision = True
    Param(0).Value = Angulo
    'Debug.Print "Angulo:" & Angulo
    RootProduct.Update
    Call ClashCreate
    For k = 1 To 2
        If cConflicts(k).Count <> 0 Then
            ConflictsCount = ConflictsCount + cConflicts(k).Count
        End If
    Next k
    If ConflictsCount <> 0 Then
        Colision = True
        Call ClashErase
        GoTo ClashObjective
    Else
        Call ClashErase
    End If
    Angulo = Angulo + 10
Loop
ClashObjective:
If Colision = True Then
    Fobj(C) = EvalfObjective(C) * 10

```

```

Else
    Fobj(C) = EvalfObjective(C)
End If
fit(C) = Abs(cycleTime - Fobj(C))
'If fit(C) < 0 Then
'  fit(C) = Abs(fit(C) * 10)
'End If
'For i = 0 To 5
'  If velocity(i, C) = 3000 Then
'    fit(C) = Abs(fit(C)) * 10
'  End If
'Next i
'Debug.Print Fobj(c)
If Colision = True Then
    ColisionResult(genNumber, C) = 1
Else
    ColisionResult(genNumber, C) = 0
End If
Next C
For C = 0 To popSize - 1
    If fit(C) > max Then
        max = fit(C)
    ElseIf fit(C) > maxx And fit(C) < max Then
        maxx = fit(C)
    End If
Next C
End Sub
Sub ReCrossover()
    Dim i, w, j, k, s, Num, A, B, D, E As Integer
    ReDim OffSpring(popSize, 11)
    k = 0
    i = 0
    While k < popSize
        Ranking(k) = Rolete(0, 1)
        If Ranking(k) < crossRate Then
            Parents(Num) = k
            Num = Num + 1
            'Debug.Print k
        End If
        'Parents(Num) = k
        k = k + 1
    Wend
    'Debug.Print "-----"
    "
    'For i = 0 To Num - 1
        'Debug.Print Parents(i)
    'Next i
    'Debug.Print Num
    'Debug.Print "-----"
    "
    ReDim C(Num)
    For j = 0 To Num - 1
        C(j) = randomNumber(1, 12)
        'Debug.Print "Crossover point(" & j & ")" &
        " In the generation(" & genNumber & "):" & C(j)
    Next j
    CrossoverCount = 0
    For j = 0 To popSize - 1
        For i = 0 To 11
            OffSpring(j, i) = SelectedPopulation(j, i)
        Next i
    Next j
    For j = 0 To Num - 1
        A = Parents(j)

```

```

B = C(j)

D = C(j) - 1

If C(j) = 12 Then
    GoTo NextParent
End If

If SelectedPopulation(A, D) <
SelectedPopulation(A + 1, B) Then

    CrossoverCount = CrossoverCount + 1

    For i = 0 To 11

        If j < Num - 1 Then

            If i < C(j) Then

                'Debug.Print "Chromossomes(" &
                Parents(j) & "," & i & "):" &
                Chromossomes(Parents(j), i)

                OffSpring(Parents(j), i) =
                SelectedPopulation(Parents(j), i)

                ElseIf i >= C(j) Then

                'Debug.Print "Chromossomes(" &
                Parents(j) & "," & i & "):" &
                Chromossomes(Parents(j + 1), i)

                OffSpring(Parents(j), i) =
                SelectedPopulation(Parents(j + 1), i)

            End If

            ElseIf j = Num - 1 Then

                If i < C(j) Then

                    OffSpring(Parents(j), i) =
                    SelectedPopulation(Parents(j), i)

                    ElseIf i >= C(j) Then

                    OffSpring(Parents(j), i) =
                    SelectedPopulation(Parents(0), i)

                End If

            End If

        Next i

    End If

NextParent:

Next j

```

```

End Sub

Sub ReMutation()

    Dim totalGene As Integer

    Dim i, j, k, C, max, maxx, genumbert2 As
    Integer

    Dim mutationNum As Double

    Dim C1 As String

    genumbert2 = genNumber

    totalGene = 12 * popSize

    mutationNum = Round(totalGene *
    mutationRate, 0)

    'Debug.Print "MUTATION RATE:" &
    mutationRate

    If mutationNum >= 1 Then

        ReDim r(mutationNum - 1),
        mutposi(mutationNum - 1),
        mutposj(mutationNum - 1), H(mutationNum -
        1), F(mutationNum - 1)

        For i = 0 To mutationNum - 1

            r(i) = mutRandNum(1, totalGene)

            'Debug.Print " Valor de R(" & i & "):"; R(i)

        ReMutationEvaluate:

            If i <> 0 Then

                If r(i) = r(i - 1) Then

                    r(i) = mutRandNum(1, totalGene)

                    GoTo ReMutationEvaluate

                End If

            End If

            H(i) = r(i) / 12

            mutposi(i) = CInt(r(i) / 12)

            'Debug.Print H(i) - mutposi(i)

            If H(i) - mutposi(i) > 0 Then

                mutposi(i) = CInt(r(i) / 12)

            ElseIf H(i) - mutposi(i) = 0 Then

```

```

    mutposi(i) = Clnt(r(i) / 12) - 1
Elseif H(i) - mutposi(i) < 0 Then
    mutposi(i) = (Clnt(r(i) / 12)) - 1
End If

'Debug.Print mutposi(i)
F(i) = (12 * mutposi(i)) + 1
mutposj(i) = r(i) - F(i)
'Debug.Print mutposj(i)

Next
For i = 0 To mutationNum - 1
    'Debug.Print "OLD:" &
    Chromossomes(mutposi(i), mutposj(i))

    OffSpring(mutposi(i), mutposj(i)) =
    MutationChromoCreate(mutposi(i), mutposj(i))

    'Debug.Print "NEW:" &
    Chromossomes(mutposi(i), mutposj(i))

    Next i

End If

For C = 0 To popSize - 1
    j = 0
    For i = 0 To 5
        k = j + 1
        delta(i, C) = EvalfAngle(OffSpring(C, j),
        OffSpring(C, k))
        time(i, C) = EvalfTime(delta(i, C))
        If i = 0 Or i = 4 Then
            velocity(i, C) = 330 / time(i, C)
        Elseif i = 1 Or i = 3 Then
            velocity(i, C) = 245 / time(i, C)
        Elseif i = 2 Or i = 5 Then
            velocity(i, C) = 925 / time(i, C)
        End If
        'Debug.Print delta(i, c)

```

```

'Debug.Print "time(" & i & "," & c & "):" &
time(i, c)
'Debug.Print velocity(i, c)
j = j + 2
Next i
Fobj(C) = EvalfObjective(C)
fit(C) = Abs(cycleTime - Fobj(C))
'If fit(C) < 0 Then
' fit(C) = Abs(fit(C) * 10)
'End If
'For i = 0 To 5
' If velocity(i, C) = 3000 Then
' fit(C) = Abs(fit(C)) * 10
' End If
'Next i
'Debug.Print Fobj(c)
Next C
For C = 0 To popSize - 1
    If fit(C) > max Then
        max = fit(C)
    Elseif fit(C) > maxx And fit(C) < max Then
        maxx = fit(C)
    End If
Next C
End Sub

Function FstChromossomeCreate(ByVal min As Integer, ByVal max As Integer)
    Randomize
    FstChromossomeCreate = Int((max - min + 1)
    * Rnd + min)
End Function

Function EvalfAngle(ByVal Angle1 As Integer,
ByVal Angle2 As Integer) As Double

```

```

Dim erro As String
If Angle2 > Angle1 Then
    EvalfAngle = Angle2 - Angle1
Elseif Angle1 > Angle2 Then
    EvalfAngle = (360 - Angle1) + Angle2
End If
If Angle1 = Angle2 Then
    EvalfAngle = 1
End If
End Function

Function EvalfTime(ByVal deltaAngle1 As Double) As Double
    EvalfTime = (deltaAngle1 * cycleTime) / 360
End Function

Function EvalfVelocity(ByVal time As Double, ByVal i As Integer) As Double
    If i = 0 Or i = 4 Then
        EvalfVelocity = 330 / time
    Elseif i = 1 Or i = 3 Then
        EvalfVelocity = 245 / time
    Elseif i = 2 Or i = 5 Then
        EvalfVelocity = 925 / time
    End If
    If EvalfVelocity > 3000 Then
        'GoTo CreateChromossome
        EvalfVelocity = 3000
    End If
End Function

Function EvalfObjective(ByVal w As Integer)
    EvalfObjective = time(0, w) + time(1, w) +
time(2, w) + time(3, w) + time(4, w) + time(5, w)
End Function

```

```

Function mutRandNum(ByVal min As Double, ByVal max As Double)
    Randomize
    mutRandNum = Int((max - min + 1) * Rnd + min)
End Function

Function Rolete(ByVal min As Double, ByVal max As Double) As Double
    Randomize
    Rolete = (max - min) * Rnd() + min
End Function

Function randomNumber(ByVal min As Double, ByVal max As Double) As Double
    Randomize
    randomNumber = Int((max - min + 1) * Rnd + min)
End Function

Function MutationChromoCreate1(ByVal i As Double, ByVal j As Double)
    Dim Mut1, Mut2 As Double
    Dim k As Integer
    If j = 0 Then
        Mut1 = 180
        Mut2 = OffSpring(i, 1) - 1
    Elseif j = 1 Then
        Mut1 = OffSpring(i, 2) - 1
        Mut2 = OffSpring(i, 2) - 1
    Elseif j = 2 Then
        Mut1 = OffSpring(i, j - 1) + 1
        Mut2 = OffSpring(i, j - 1) + 1
    Elseif j = 3 Then
        Mut1 = OffSpring(i, j - 1) + 1
        Mut2 = OffSpring(i, j + 1) - 1
    Elseif j = 4 Then

```

```

If OffSpring(i, j - 1) <= 360 Then
    Mut1 = 1
Elseif OffSpring(i, j - 1) = 1 Then
    Mut1 = 2
End If

Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 5 Then
    If OffSpring(i, j - 1) <= 360 Then
        Mut1 = 1
    Elseif OffSpring(i, j - 1) = 1 Then
        Mut1 = 2
    End If
    Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 6 Then
    Mut1 = OffSpring(i, j - 1) + 1
    Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 7 Then
    Mut1 = OffSpring(i, j - 1) + 1
    Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 8 Then
    Mut1 = OffSpring(i, j - 1) + 1
    Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 9 Then
    Mut1 = OffSpring(i, j - 1) + 1
    Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 10 Then
    Mut1 = OffSpring(i, j - 1) + 1
    Mut2 = OffSpring(i, j + 1) - 1
Elseif j = 11 Then
    Mut1 = OffSpring(i, j - 1) + 1
    Mut2 = 179
End If

MutationChromoCreate =
FstChromossomeCreate(Mut1, Mut2)
End Function

Function MutationChromoCreate(ByVal i As
Double, ByVal j As Double)
    Dim Mut1, Mut2 As Double
    Dim k As Integer
    If j = 0 Then
        Mut1 = 180
        Mut2 = OffSpring(i, 1) - 1
    Elseif j = 1 Then
        Mut1 = OffSpring(i, 0) + 1
        Mut2 = OffSpring(i, 2) - 1
    Elseif j = 2 Then
        Mut1 = OffSpring(i, 1) + 1
        Mut2 = OffSpring(i, 1) + 1
    Elseif j = 3 Then
        Mut1 = OffSpring(i, 2) + 1
        Mut2 = OffSpring(i, 4) - 1
    Elseif j = 4 Then
        Mut1 = OffSpring(i, 3) + 1
        Mut2 = OffSpring(i, 3) + 1
    Elseif j = 5 Then
        Mut1 = 1
        Mut2 = OffSpring(i, 6) - 1
    Elseif j = 6 Then
        Mut1 = OffSpring(i, 5) + 1
        Mut2 = OffSpring(i, 5) + 1
    Elseif j = 7 Then
        Mut1 = OffSpring(i, 6) + 1
        Mut2 = OffSpring(i, 8) - 1
    End If
End Function

```

```

Elseif j = 8 Then
    Mut1 = OffSpring(i, 7) + 1
    Mut2 = OffSpring(i, 7) + 1
Elseif j = 9 Then
    Mut1 = OffSpring(i, 8) + 1
    Mut2 = OffSpring(i, 10) - 1
Elseif j = 10 Then
    Mut1 = OffSpring(i, 9) + 1
    Mut2 = OffSpring(i, 9) + 1
Elseif j = 11 Then
    Mut1 = OffSpring(i, 10) + 1
    Mut2 = 179
End If

MutationChromoCreate = FstChromossomeCreate(Mut1, Mut2)

'Debug.Print "MutationChromoCreate:" & MutationChromoCreate

'If j > 0 Then
'    Debug.Print "Chromossomes(i, j - 1):" & Chromossomes(i, j - 1)
'End If

'If j < 11 Then
'Debug.Print "Chromossomes(i, j + 1):" & Chromossomes(i, j + 1)
'End If

End Function

Function AttributesActualization(ByVal j As Double)

    Set documents1 = CATIA.Documents

    Set RootProd = CATIA.ActiveDocument.Product

    Set partDocument1 = documents1.Item("_0__laws__000-0000.CATPart")

    Set part1 = partDocument1.Part

    Set parameters1 = part1.Parameters

    Param(4).Value = SelectedPopulation(j, 0) 'y1_inicio
    Param(5).Value = SelectedPopulation(j, 1) 'y1_fim
    Param(6).Value = SelectedPopulation(j, 2) 'z1__inicio
    Param(7).Value = SelectedPopulation(j, 3) 'z1_fim
    Param(8).Value = SelectedPopulation(j, 4) 'x1__inicio
    Param(9).Value = SelectedPopulation(j, 5) 'x1_fim
    Param(10).Value = SelectedPopulation(j, 6) 'z2__inicio
    Param(11).Value = SelectedPopulation(j, 7) 'z2_fim
    Param(12).Value = SelectedPopulation(j, 8) 'y2__inicio
    Param(13).Value = SelectedPopulation(j, 9) 'y2_fim
    Param(14).Value = SelectedPopulation(j, 10) 'x2__inicio
    Param(15).Value = SelectedPopulation(j, 11) 'x2_fim

    Set documents1 = Nothing
    Set RootProd = Nothing
    Set partDocument1 = Nothing
    Set part1 = Nothing
    Set parameters1 = Nothing
End Function

Private Sub Values()

    Set documents1 = CATIA.Documents

    Set RootProd = CATIA.ActiveDocument.Product

    Set partDocument1 = documents1.Item("_0__laws__000-0000.CATPart")

```

```

Set part1 = partDocument1.Part
Set parameters1 = part1.Parameters

Set Param(0) = parameters1.Item("Angulo")

Set Param(1) = parameters1.Item("Curso_xx")
Set Param(2) = parameters1.Item("Curso_yy")
Set Param(3) = parameters1.Item("Curso_zz")
Set Param(4) = parameters1.Item("Y1_ENTRAR_INICIO")
Debug.Print Param(4).Value
Set Param(5) = parameters1.Item("Y1_ENTRAR_FIM")
Set Param(6) = parameters1.Item("Z1_SUBIR_INICIO")
Set Param(7) = parameters1.Item("Z1_SUBIR_FIM")
Set Param(8) = parameters1.Item("X1_AVANCO_INICIO")
Set Param(9) = parameters1.Item("X1_AVANCO_FIM")
Set Param(10) = parameters1.Item("Z2_DESCER_INICIO")
Set Param(11) = parameters1.Item("Z2_DESCER_FIM")
Set Param(12) = parameters1.Item("Y2_SAIR_INICIO")
Set Param(13) = parameters1.Item("Y2_SAIR_FIM")
Set Param(14) = parameters1.Item("X2_RECUEO_INICIO")

Set Param(15) = parameters1.Item("X2_RECUEO_FIM")
Set Param(16) = parameters1.Item("GPM")

Dist(0) = 925 'Curso_xx
Dist(1) = 330 'Curso_yy
Dist(2) = 245 'Curso_zz

mutationRate = 0.01

crossRate = 0.6 'A taxa de crossover deve ser em geral alta, cerca de 80% - 95%. Entretanto, alguns resultados mostram que para alguns tipos de problemas, uma taxa de crossover de cerca de 60% é o melhor.

GPM = 10
cycleTime = 60 / GPM

Set documents1 = Nothing
Set RootProd = Nothing
Set partDocument1 = Nothing
Set part1 = Nothing
Set parameters1 = Nothing

'mutationRate = 0.01 'As melhores taxas parecem estar na faixa de 0.5% - 1%

End Sub

Sub EraseParam()
Dim i As Integer
For i = 0 To 14
Set Param(i) = Nothing
Next i
End Sub

```

