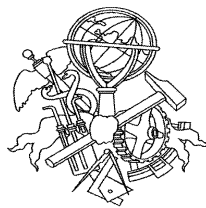


BALANCEAMENTO DE LINHAS DE PRODUÇÃO USANDO ALGORITMOS GENÉTICOS

Ricardo Guedes Pimenta



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

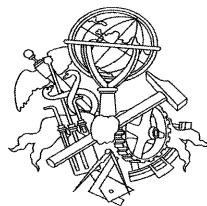
2011

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Ricardo Guedes Pimenta, Nº 1050799, 1050799@isep.ipp.pt

Orientação científica: Manuel Santos Silva, mss@isep.ipp.pt

Co-Orientação científica: Cecília Reis, cmr@isep.ipp.pt



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

21 de Novembro de 2011

Agradecimentos

Gostaria de expressar a minha sincera gratidão às seguintes pessoas que contribuíram de forma directa ou indirecta para a realização deste trabalho:

- Professores Manuel Silva e Cecília Reis pela sua disponibilidade, orientação e dedicação.
- Um agradecimento especial aos meus pais, que sem o seu apoio e esforço, este trabalho nunca teria sido realizado.

Resumo

Aquando da definição de um *layout* por fluxo de produto, ou linha de produção, é necessário proceder-se à melhor selecção de combinações de tarefas a serem executadas em cada estação / posto de trabalho para que o trabalho seja executado numa sequência exequível e sejam necessárias quantidades de tempo aproximadamente iguais em cada estação / posto de trabalho. Este processo é chamado de balanceamento da linha de produção.

Verifica-se que as estações de trabalho e equipamentos podem ser combinados de muitas maneiras diferentes; daí que a necessidade de efectuar o balanceamento das linhas de produção implique a distribuição de actividades sequenciais por postos de trabalho de modo a permitir uma elevada utilização de trabalho e de equipamentos e a minimizar o tempo de vazio. Os problemas de balanceamento de linhas são tipicamente problemas complexos de tratar, devido ao elevado número de combinações possíveis. Entre os métodos utilizados para resolver estes problemas encontram-se métodos de tentativa e erro, métodos heurísticos, métodos computacionais de avaliação de diferentes opções até se encontrar uma boa solução e métodos de optimização.

O objectivo deste trabalho passou pelo desenvolvimento de uma ferramenta computacional para efectuar o balanceamento de linhas de produção recorrendo a algoritmos genéticos. Foi desenvolvida uma aplicação que implementa dois algoritmos genéticos, um primeiro que obtém soluções para o problema e um segundo que optimiza essas soluções, associada a uma interface gráfica em C# que permite a inserção do problema e a visualização de resultados. Obtiveram-se resultados exequíveis demonstrando vantagens em relação aos métodos heurísticos, pois é possível obter-se mais do que uma solução. Além disso, para problemas complexos torna-se mais prático o uso da aplicação desenvolvida. No entanto, esta aplicação permite no máximo seis precedências por cada operação e resultados com o máximo de nove estações de trabalho.

Palavras-Chave

Linhas de produção, balanceamento de linhas de produção, métodos heurísticos, algoritmos genéticos.

Abstract

When defining a layout for product flow, or production line, it is necessary to select the best combination of tasks to be performed at each workstation, so that the work is performed in a feasible sequence and equal amounts of time are needed at each workstation. This process is called assembly line balancing.

It is verified that the workstations and equipment can be combined in many different ways. Therefore, the need for balancing of assembly lines involves the distribution of sequential activities for jobs in order to allow a high utilization of labor and equipment and minimize the idle time. Line balancing problems are typically complex to treat due to the high number of possible combinations, and among the methods used to solve these problems it is possible to find trial and error methods, heuristics, computational methods for evaluating different options to find a good solution and optimization methods.

The aim of this work was the development of a computational tool to perform the balancing of production lines using genetic algorithms. Following this idea, it was developed an application that implements two genetic algorithms (the first one obtains solutions to the problem and the second optimizes those solutions), associated with the development of a graphical interface in C # for the input of the problem data and visualization of results. Achieved results demonstrate advantages over heuristic methods as it is possible to obtain more than one solution. Moreover, for complex problems it is more practical to use the developed application. However, this application allows a maximum of six precedence relations for each operation and results with a maximum of nine workstations.

Keywords

Assembly lines, assembly line balancing, heuristic methods, genetic algorithms.

Índice

AGRADECIMENTOS	III
RESUMO	V
ABSTRACT	VII
ÍNDICE	IX
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XIV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	2
1.2. OBJECTIVOS.....	2
1.3. CALENDARIZAÇÃO	3
1.4. ORGANIZAÇÃO DO RELATÓRIO	4
2. LAYOUTS FABRIS	5
2.1. LAYOUT.....	6
2.1.1. Layout de posição fixa.....	7
2.1.2. Layout de processos intermitentes	8
2.1.3. Layout em linha.....	8
2.1.4. Layout em fluxo contínuo	10
3. BALANCEAMENTO DE LINHAS DE PRODUÇÃO	11
3.1. DEFINIÇÕES E CONCEITOS DO DE BALANCEAMENTO DE LINHAS DE PRODUÇÃO	12
3.2. OBJECTIVOS DO BALANCEAMENTO DAS LINHAS	13
3.3. MÉTODOS DE BALANCEAMENTO DE LINHAS DE PRODUÇÃO	13
3.3.1. Métodos heurísticos	14
3.3.1.1. Heurística de helgeson-birnie	14
3.3.1.2. Heurística de kilbridge e wester.....	15
3.3.2. Aplicações de software para balanceamento de linhas.....	15
3.3.2.1. ProBalance.....	16
3.3.2.2. AviX® Balance.....	18
3.4. PROJECTOS DE BALANCEAMENTO DE LINHAS USANDO AG	20
3.4.1. Algoritmo genético para o balanceamento de linhas de produção.....	20
3.4.2. Balanceamento de linhas de produção de modelo misto com algoritmo genético híbrido ...	21
4. ALGORITMOS GENÉTICOS	23

4.1.	DEFINIÇÃO E FUNDAMENTOS BIOLÓGICOS DOS ALGORITMOS GENÉTICOS	23
4.2.	ESPAÇO DE SOLUÇÕES	24
4.3.	ESTRUTURA DE UM ALGORITMO GENÉTICO	25
4.4.	MECANISMOS DE LIGAÇÃO DE UM ALGORITMO GENÉTICO AO PROBLEMA	26
4.4.1.	<i>Codificação do problema</i>	26
4.4.1.1.	<i>Codificação binária</i>	27
4.4.1.2.	<i>Codificação por permutação</i>	27
4.4.2.	<i>Função de avaliação</i>	27
4.5.	OPERADORES DO ALGORITMO GENÉTICO	28
4.5.1.	<i>Cruzamento</i>	28
4.5.2.	<i>Mutação</i>	28
4.5.3.	<i>Seleção</i>	29
4.6.	CRITÉRIOS DE PARAGEM.....	32
5.	ARQUITECTURA DA APLICAÇÃO A IMPLEMENTAR.....	33
5.1.	ALGORITMO GENÉTICO DESENVOLVIDO.....	33
5.1.1.	<i>Estrutura do Algoritmo Genético desenvolvido</i>	34
5.1.2.	<i>Codificação</i>	35
5.1.3.	<i>Geração da população inicial</i>	35
5.1.4.	<i>Seleção</i>	37
5.1.5.	<i>Função de avaliação</i>	37
5.1.6.	<i>Cruzamento e mutação</i>	38
5.1.7.	<i>Critério de paragem</i>	40
5.2.	ALGORITMO GENÉTICO DE MELHORAMENTO DE RESULTADOS	41
5.2.1.	<i>Estrutura geral do algoritmo</i>	41
5.2.2.	<i>Seleção</i>	42
5.2.3.	<i>Mutação</i>	42
5.2.4.	<i>Função de avaliação</i>	42
5.2.5.	<i>Critério de paragem</i>	42
5.3.	INTERFACE DESENVOLVIDA.....	43
6.	TESTES DA APLICAÇÃO DESENVOLVIDA	49
6.1.	PRIMEIRO TESTE	49
6.2.	SEGUNDO TESTE	52
6.3.	TERCEIRO TESTE.....	55
6.4.	QUARTO TESTE.....	57
7.	CONCLUSÕES E PERSPECTIVAS DE DESENVOLVIMENTOS FUTUROS.....	63
	REFERÊNCIAS DOCUMENTAIS.....	65

Índice de Figuras

Figura 1	Exemplo de linhas de produção ou montagem.....	5
Figura 2	Estaleiro de construção de navios.	7
Figura 3	Linha de montagem com <i>layout</i> em linha	9
Figura 4	Exemplo de um <i>layout</i> de fluxo contínuo	10
Figura 5	Interface do <i>software</i> ProBalance.....	16
Figura 6	Gráfico de precedências	17
Figura 7	Gráfico de balanceamento.....	17
Figura 8	Interface da aplicação AviX® Balance	19
Figura 9	Opção do gráfico do balanceamento	19
Figura 10	Relatório de balanceamento	20
Figura 11	Exemplo de espaço de soluções	25
Figura 12	Exemplo da operação de cruzamento de dois cromossomas.....	28
Figura 13	Exemplo da realização de uma mutação de cromossomas.....	29
Figura 14	Probabilidade de selecção dos cromossomas utilizando o método de selecção aleatória por roleta	30
Figura 15	Probabilidade de selecção antes de classificação (esquerda) e probabilidade depois da classificação (direita).....	31
Figura 16	Fluxograma do Algoritmo Genético desenvolvido	34
Figura 17	Estrutura do algoritmo de melhoramento.....	41
Figura 18	Interface gráfica da aplicação.....	43
Figura 19	Exemplo de introdução de dados.....	44
Figura 20	Seleccção de tipo de problema.....	44
Figura 21	Seleccção do número de soluções.....	45
Figura 22	Interface gráfica da aplicação depois de executado o primeiro AG.....	46
Figura 23	Janela principal da interface gráfica com mensagem de aviso.....	46
Figura 24	Menu para correcção de dados do problema.....	47
Figura 25	Janela de análise de resultados	47
Figura 26	Análise dos resultados do segundo algoritmo	48
Figura 27	Solução obtida para o problema 1	51
Figura 28	Solução obtida para o segundo problema de teste da aplicação.....	54
Figura 29	Solução obtida para o terceiro problema.	56
Figura 30	Janela principal depois do primeiro Algoritmo executado no quarto teste	59
Figura 31	Análise de um cromossoma da população do quarto teste	60
Figura 32	Solução obtida para o quarto teste.....	60

Índice de Tabelas

Tabela 1	Calendarização do trabalho	3
Tabela 2	Cromossomas exemplo com aptidão e probabilidade de selecção	30
Tabela 3	População exemplo de cromossomas, com respectiva aptidão e classificação	31
Tabela 4	Exemplo de codificação das tarefas.	35
Tabela 5	Processo executado para baralhar o cromossoma	37
Tabela 6	Exemplo de aplicação do primeiro método de mutação testado.	39
Tabela 7	Exemplo de aplicação do segundo método de mutação testado.	39
Tabela 8	Exemplo de aplicação do terceiro método de mutação testado.	39
Tabela 9	Tabela de operações, tempos de execução e precedência do problema	50
Tabela 10	Tabela de comparação de resultados para o problema 1	52
Tabela 11	Tabela de operações, tempos de execução e precedência do problema 2	52
Tabela 12	Tabela de comparação de resultados para o problema 2	54
Tabela 13	Tabela de operações, tempos de execução e precedência do problema 3	55
Tabela 14	Tabela de comparação de resultados para o problema 3	57
Tabela 15	Tabela de operações, tempos de execução e precedência do problema 4	58
Tabela 16	Tabela de comparação de resultados para o problema 4	61

Acrónimos

AG – Algoritmo Genético

ISEP – Instituto Superior de Engenharia do Porto

DEE – Departamento de Engenharia Electrotécnica

BL – Balanceamento de Linha

1. INTRODUÇÃO

O primeiro conceito de linha de produção foi apresentado em 1901 por Ransom E. Olds [1]. Este apresenta um conceito básico da linha de produção que lhe permite iniciar a indústria automóvel em Detroit. Através deste conceito, Olds consegue quadruplicar a produção automóvel na sua fábrica, passando de 425 carros em 1901 para 2500 carros em 1902. O conceito de linhas de produção que é conhecido nos dias de hoje foi apresentado por Henry Ford em 1908, na produção do famoso Modelo T da Ford. Esta linha de produção foi a primeira a apresentar um sistema de transporte que permitia ao produto (neste caso o carro) circular pelas diferentes estações de montagem chegando ao fim da linha no estado final. Isto permitiu que a Ford se tornasse no maior produtor automóvel do mundo em 1913.

Hoje em dia, devido ao aumento da concorrência no mercado, o problema de balanceamento de linhas, especialmente o balanceamento de linhas de produção, tem um papel importante para as indústrias obterem alta qualidade nos produtos e o menor custo possível.

O balanceamento de uma linha de produção é um método de análise que tem como objectivo distribuir de igual forma o trabalho a ser executado pelas diferentes estações de trabalho ou postos de trabalho de tal forma que o número de trabalhadores ou estações de trabalho na linha de produção sejam minimizados e rentabilizados de forma equiparada.

1.1. CONTEXTUALIZAÇÃO

Este trabalho surgiu no seguimento da proposta apresentada pelos docentes Manuel Silva e Cecília Reis, a qual envolvia a interligação de duas áreas bastante interessantes, nomeadamente linhas de produção e algoritmos genéticos (AG). Este tema cativou o meu interesse, permitindo-me explorar e aprofundar conhecimentos em duas áreas distintas.

Sendo as linhas de produção parte essencial das indústrias actuais e o problema de balanceamento de linhas de produção um problema que surge no dia-a-dia, a resolução rápida e eficaz deste problema torna-se fundamental. Propôs-se então, com este projecto, o desenvolvimento de uma aplicação baseada numa abordagem actual que produza resultados rápidos e com elevada eficiência.

1.2. OBJECTIVOS

O objectivo deste projecto passava pelo desenvolvimento de uma aplicação computacional, tendo por base um algoritmo genético, que permitisse a resolução de problemas de balanceamento de linhas.

Esta aplicação deveria permitir a introdução dos dados dos problemas a solucionar de forma fácil e amigável para o utilizador, resolver o problema do balanceamento da linha de produção recorrendo a um algoritmo genético e permitir ao utilizador analisar os resultados obtidos, mais uma vez, de uma forma fácil e amigável.

Dentro do possível, a aplicação deveria gerar uma solução (ou um conjunto de soluções) exequível e com uma eficiência igual ou superior às soluções que podem ser encontradas recorrendo aos métodos heurísticos.

Para atingir este objectivo global, foi necessário:

- Uma abordagem profunda dos métodos heurísticos em todos os seus aspectos;
- Desenvolvimento do conhecimento de programação C#;
- Aprendizagem e estudo dos algoritmos genéticos.

1.3. CALENDARIZAÇÃO

Neste tópico é apresentada a calendarização da elaboração deste projecto, apresentando as várias fases do mesmo. Na Tabela 1 são apresentadas as várias fases do trabalho, bem como o tempo decorrido em cada fase do mesmo.

Tabela 1 Calendarização do trabalho

	2010			2011								
	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Marco	Abril	Maiο	Junho	Julho	Agosto	Setembro
1	█	█	█									
2		█	█	█	█	█	█	█	█	█	█	█
3			█	█	█	█	█	█	█	█	█	█
4				█	█	█	█	█	█	█	█	█
5			█	█	█	█	█	█	█	█	█	█
6		█	█	█	█	█	█	█	█	█	█	█
7							█	█	█	█	█	█
8						█	█	█	█	█	█	█
9									X			
10				█	█	█	█	█	█	█	█	█

De seguida é apresentada a legenda da tabela anteriormente apresentada, onde se descreve a fase do trabalho correspondente ao número apresentado na tabela.

- 1 - ESTUDO DO ESTADO DA ARTE
- 2 - ESTUDO DOS PROBLEMAS DE BALANCEAMENTO DE LINHAS
- 3 - DESENVOLVIMENTO DO ALGORITMO GENÉTICO
- 4 - MELHORIAS AO ALGORITMO GENÉTICO
- 5 - TESTES AO ALGORITMO GENÉTICO
- 6 - DESENVOLVIMENTO DA INTERFACE GRÁFICA
- 7 - TESTES DA APLICAÇÃO DESENVOLVIDA
- 8 - PREPARAÇÃO DE ARTIGO PARA CONFERÊNCIA
- 9 - APRESENTAÇÃO DE ARTIGO NA CONFERÊNCIA
- 10 - REDACÇÃO DA TESE/DISSERTAÇÃO

1.4. ORGANIZAÇÃO DO RELATÓRIO

Este relatório encontra-se estruturado de acordo com os capítulos seguintes:

- Capítulo 1: neste primeiro capítulo, efectua-se uma introdução ao trabalho desenvolvido;
- Capítulo 2: aqui são apresentados os conceitos de linhas de produção e os seus possíveis diferentes *layouts*;
- Capítulo 3: o terceiro capítulo introduz o balanceamento de linhas de produção e os diferentes métodos para efectuar este processo;
- Capítulo 4: no quarto capítulo são apresentados os Algoritmos Genéticos;
- Capítulo 5: é introduzida a arquitectura implementada, descrevendo-se o algoritmo desenvolvido e a interface gráfica implementada;
- Capítulo 6: neste capítulo são apresentados alguns dos testes efectuados à aplicação desenvolvida e os principais resultados obtidos;
- No Capítulo 7 são apresentadas as conclusões, sendo recapitulados os principais resultados obtidos e indicadas perspectivas para desenvolvimentos futuros.

2. *LAYOUTS FABRIS*

Uma linha de produção consiste num grupo de postos de trabalho, cuja posição é fixa e em que a sequência é imposta pela lógica das sucessivas operações a realizar e descritas na gama operatória. Um posto de trabalho tem uma determinada área física e pode ser composto por um único operador, ou por vários operadores, realizando operações manuais ou assistidas por ferramentas ou maquinaria ou ainda uma máquina autónoma, como um robô, que executam um determinado tipo de operações.

As imagens apresentadas na Figura 1 representam exemplos de linhas de produção ou montagem.



Figura 1 Exemplo de linhas de produção ou montagem

O objectivo da análise do balanceamento das linhas de produção é determinar o número de estações de trabalho necessárias e quais as tarefas ou operações associadas a cada uma delas. Através desta análise é possível obter a capacidade pretendida, como o número mínimo de operadores, máquinas e ferramentas, ou, por outro lado, consegue-se aumentar a capacidade de produção com os recursos disponíveis.

Neste capítulo são apresentados diversos *layouts* de linhas de produção, sendo apresentada uma breve descrição das suas vantagens e desvantagens.

2.1. LAYOUT

O *layout* é uma das áreas de estratégia que determinam a eficiência das operações [2]. O objectivo da estratégia do *layout* é o de desenvolver uma economia de *layout* que atinja os seguintes requisitos:

1. Projecto de produto e volume (estratégia de produto);
2. Equipamentos de processo e capacidade (estratégia de processo);
3. Qualidade de vida no trabalho (estratégia de recursos humanos);
4. Edifício e restrições do terreno (estratégia de localização).

O *layout* especifica o arranjo dos processos (tais como soldadura, fresagem e pintura), o equipamento relacionado e áreas de trabalho, incluindo serviço de clientes e armazéns. Um *layout* eficiente facilita o fluxo de material e pessoal dentro e entre áreas de trabalho. O objectivo da administração é arranjar o sistema (definir o *layout*) para que o mesmo esteja operacional no máximo de eficiência e eficácia. Decisões de *layout* incluem o melhor posicionamento de máquinas (configuração de produção), escritórios e secretárias (configuração de escritórios) ou centros de serviços (por exemplo, em configurações de hospitais) [2, 3].

Para atingir os objectivos desses *layouts*, foram desenvolvidas algumas estratégias. Dentro dessas estratégias são discutidas as seguintes:

1. *Layout* de posição fixa;
2. *Layouts* de processos intermitentes;

3. *Layouts* em linha;
4. *Layouts* em fluxo contínuo.

2.1.1. **LAYOUT DE POSIÇÃO FIXA**

Um *layout* de posição fixa é um *layout* no qual o produto se mantém estacionário e os operários e os equipamentos são deslocados para a área de trabalho, sendo um dos *layouts* mais “básicos”. Exemplos típicos deste tipo de *layouts* podem ser encontrados na construção de navios, auto-estradas, pontes e casas. Neste caso, os trabalhadores e os materiais são colocados no local onde vai ser realizado o trabalho [4]. Na Figura 2 é apresentado um exemplo de um estaleiro de construção naval.

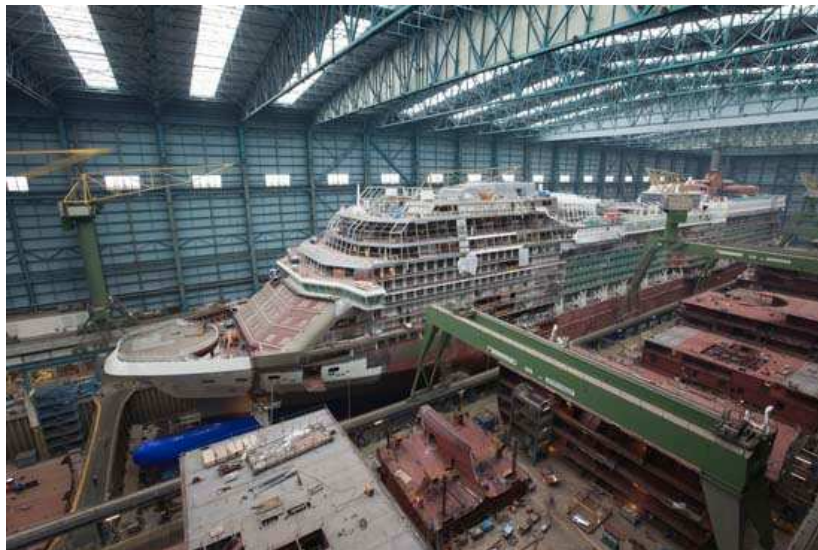


Figura 2 Estaleiro de construção de navios.

Neste tipo de *layout* os custos de material são normalmente muito grandes. Tenta-se colocar os materiais mais usados perto do local de construção/produção, sendo que os materiais que se usam menos são colocados mais longe. Outro factor importante na colocação de materiais é a sua precedência temporal. Este factor é particularmente importante sempre que o espaço é limitado [4].

De entre as principais vantagens deste tipo de *layout*, destaca-se o facto de ser possível efectuar um melhor planeamento e controlo do trabalho, dado que tudo está orientado para um único objectivo.

Por outro lado, este tipo de *layout* apresenta como principais desvantagens as seguintes:

- Os custos de deslocação de pessoal especializado podem ser elevados;
- Falta de estruturas de apoio, tais como energia eléctrica e água;
- Utilização de equipamento muito caro, que não é totalmente aproveitado.

2.1.2. **LAYOUT DE PROCESSOS INTERMITENTES**

Este tipo de *layout* permite manusear simultaneamente uma grande variedade de produtos ou serviços. Na verdade, é mais eficaz quando se encontra aplicado ao fabrico de diferentes produtos que têm diferentes requisitos ou quando se trata de clientes que têm diferentes necessidades. Neste ambiente cada produto, ou cada pequeno grupo de produtos, tem uma sequência diferente de operações. Um produto ou uma pequena encomenda é produzida movendo o produto de um departamento para outro, numa sequência necessária para o produto em questão.

Uma grande vantagem deste tipo de *layout* é a flexibilidade nos equipamentos e na atribuição de tarefas aos operários. Uma avaria numa máquina, por exemplo, não interrompe todo o processo de fabrico, pois o trabalho pode ser deslocado para outra máquina no mesmo, ou noutra, departamento.

As desvantagens deste *layout* advêm do uso geral dos equipamentos. As ordens de fabrico demoram mais tempo para se deslocar durante o processo, devido às dificuldades de agendamento, preparação dos equipamentos e manuseamento de materiais, levando a custos maiores. Adicionalmente, os operários necessitam de uma formação mais elevada e os inventários de estoques em processamento são mais elevados devido a grandes desequilíbrios no processo de produção. Também a maior habilidade dos operários requer maiores níveis de treino e experiência [2, 3].

2.1.3. **LAYOUT EM LINHA**

Sendo este projecto desenvolvido para linhas de produção com *layout* em linha, são então apresentadas as definições e conceitos deste tipo de implantação especificando as vantagens, assim como as desvantagens, deste *layout* para a produção.

Os *layouts* de linhas de produção são obtidos juntamente com as pessoas e o equipamento, de acordo com uma sequência pré-definida de operações a realizar num produto. Ao *layout* em linha costuma-se chamar linha de produção ou linha de montagem, porque normalmente são utilizados transportadores automáticos (geralmente, mas nem sempre, com a forma de uma linha recta) que minimizam o transporte de material entre os postos de trabalho. Os *layouts* em linha são utilizados, por exemplo, na montagem de automóveis, nas fábricas de produção de produtos alimentares, etc.[4]. Na Figura 3 é possível visualizar-se um destes exemplos.



Figura 3 Linha de montagem com *layout* em linha

Quando se define o *layout* para uma linha não se altera a direcção do fluxo do produto. No entanto, alteram-se as tarefas destinadas aos operários individuais alterando por consequência a eficiência da linha.

Este tipo de implantação das operações produtivas apresenta as características principais que se enumeram de seguida. Em termos de vantagens, são as seguintes:

- Resultados muito eficientes;
- Menores custos de manipulação de materiais;
- Operações muito simplificadas, que permitem a utilização de mão-de-obra pouco qualificada (barata);
- Pequenos estoques intermédios;
- Simplificação do controlo da produção.

Quanto às principais desvantagens, elas são as seguintes:

- Pouca flexibilidade;
- Efeitos colaterais graves em termos de aborrecimento dos operários (devido à monotonia associada à repetição das mesmas operações) e de absentismo (tendência de mudança de emprego);
- Elevada dependência entre diversas operações (uma máquina que deixe de funcionar, pode comprometer a produção);
- É muito importante que a linha esteja bem balanceada.

2.1.4. *LAYOUT EM FLUXO CONTÍNUO*

Os *layouts* em fluxo contínuo, são utilizados maioritariamente na indústria do processo, tal como a indústria dos cimentos, a produção de produtos químicos e a produção de electricidade. Neste caso, as fábricas representam um elevado investimento de capital, dado que são altamente automatizadas e projectadas de modo a funcionarem como uma unidade. Nestas situações o *layout* tem directamente a ver com o processo e está ligado à estrutura básica da fábrica.[4] Na Figura 4 é possível ver um exemplo de uma construção (uma barragem) cuja produção é um exemplo deste tipo de *layout*.



Figura 4 Exemplo de um *layout* de fluxo contínuo

3. BALANCEAMENTO DE LINHAS DE PRODUÇÃO

Uma linha de produção é constituída por uma série de estações de trabalho, cuja posição é fixa e cuja sequência é ditada pela lógica das sucessivas operações a realizar e descritas na gama operatória. Cada estação de trabalho pode ser constituída por um único operador ou por vários operadores realizando operações manuais eventualmente assistidas por ferramentas ou pequenos equipamentos.

É importante a existência de um fluxo contínuo de materiais na linha de produção. Para que esse fluxo contínuo seja alcançado é indispensável que haja tempos de operação semelhantes nas diferentes estações de trabalho. Este balanceamento das operações dentro do tempo disponível, de acordo com a taxa de produção desejada, é o principal problema do balanceamento de linhas de produção.

Neste capítulo é apresentado o balanceamento de linhas de produção, é introduzida a sua definição e objectivos, apresentados os diferentes métodos existentes para a sua realização e apresentadas (a título de exemplo) duas aplicações informáticas que permitem o balanceamento de linhas.

3.1. DEFINIÇÕES E CONCEITOS DO DE BALANCEAMENTO DE LINHAS DE PRODUÇÃO

Pode-se definir balanceamento de uma linha como sendo um método para melhorar a taxa de transferência das linhas de produção, reduzindo os requisitos de mão-de-obra e os custos associados. Alternativamente, o balanceamento de linhas de produção é um problema de distribuição de operações pelas diferentes estações de trabalho ao longo da linha de produção, de maneira a que a distribuição de operações seja óptima em algum sentido.

Numa linha de produção constituída por um grupo de operações tem-se:

- Tempo total necessário para a produção de uma peça, que corresponde ao somatório dos tempos de todas as operações, como se pode ver na expressão (1).

$$\sum_{i=1}^k t_i \quad (1)$$

- Tempo de ciclo, que consiste no tempo entre a saída de peças consecutivas, corresponde ainda ao máximo tempo necessário em cada estação de trabalho. A equação 2 corresponde à equação para o cálculo do tempo de ciclo teórico ou planeado (também conhecido por *Takt time*), que poderá ser diferente do tempo de ciclo efectivo.

$$\text{Tempo de ciclo} = c = \frac{\text{Tempo disponivel por periodo}}{\text{numero de peças pretendidas por periodo}} \quad (2)$$

- Número mínimo de estações necessárias, é o valor obtido pela aplicação da equação (3), que é sempre arredondado para o valor inteiro imediatamente superior.

$$n_{min} = \frac{\sum_{i=1}^k t_i}{c} \quad (3)$$

- Eficiência da solução adoptada:

$$e = \frac{\sum_{i=1}^k t_i}{c} - n \times c \quad (4)$$

- Tempo total de paragem de uma linha ou folga:

$$IT = n \times c - \sum_{i=1}^k t_i \quad (5)$$

Considerando:

- n - Número de estações de trabalho;
- c - Tempo de ciclo para a linha de produção;
- t_i - Tempo necessário para completar a tarefa i da linha de produção;
- k - Número total de operações a serem executadas na linha de produção.

3.2. OBJECTIVOS DO BALANCEAMENTO DAS LINHAS

Por norma, o objectivo do balanceamento é permitir (na medida do possível) um fluxo de materiais rápido e uniforme através da linha de produção e, ao mesmo tempo, obter a máxima utilização do potencial humano, máxima utilização de máquinas, equipamentos e ferramentas e um dispêndio mínimo de material ou tempo de processo.

Portanto, o balanceamento é usado para obter os seguintes resultados na linha de produção:

- Minimizar o número de estações de trabalho;
- Minimizar o tempo de ciclo.

3.3. MÉTODOS DE BALANCEAMENTO DE LINHAS DE PRODUÇÃO

Existem vários métodos para proceder ao balanceamento de uma linha de produção, dos quais se destacam os seguintes:

- Tentativa e erro;
- Heurísticos, significando que se baseiam na lógica, no conhecimento;
- Métodos computacionais que avaliam diferentes opções até encontrarem a melhor solução;
- Métodos de otimização, baseando-se em programação linear e/ou programação dinâmica;
- Computação evolutiva nomeadamente algoritmos genéticos.

Nenhum destes métodos garante um balanceamento óptimo, mas alguns deles retornam soluções boas e viáveis que se aproximam, em maior ou menor grau, da solução óptima.

3.3.1. MÉTODOS HEURÍSTICOS

Como a própria palavra indica, heurística consiste em descobrir e inventar autonomamente sobre determinada tarefa. Ora, no contexto deste trabalho, as heurísticas fornecem métodos que fazem uso do senso comum e da lógica para obter boas aproximações à solução de problemas de balanceamento de linhas de produção.

Nesta subsecção são descritos dois métodos heurísticos através dos quais o problema em questão neste trabalho é resolvido analiticamente, para posterior comparação com os resultados obtidos através do Algoritmo Genético desenvolvido.

3.3.1.1. HEURÍSTICA DE *HELGESON-BIRNIE*

O primeiro método heurístico que se considera, é conhecido como método de Helgeson-Birnie, sendo uma das heurísticas mais simples de implementar. Esta tem em consideração o tempo de ciclo, o tempo de cada operação e as relações de precedência entre operações. Conforme o tempo de operação, são atribuídas às estações de trabalho as operações por ordem decrescente deste tempo, ou seja, as tarefas mais demoradas são efectuadas nas primeiras estações, garantindo o cumprimento das relações de precedência entre as operações. O somatório do tempo ocupado pelas tarefas em cada estação não pode ultrapassar o tempo de ciclo, pois impossibilitaria que se cumprisse o tempo de ciclo à saída da linha. Este método pode ser aplicado em problemas de pouca complexidade e onde se respeitem as precedências.

Em resumo, as etapas para aplicar esta heurística são as seguintes:

- Ordenar as operações por ordem decrescente de tempo de operação;
- Atribuir operações a uma estação, até perfazer o tempo de ciclo, respeitando as precedências.

3.3.1.2. HEURÍSTICA DE KILBRIDGE E WESTER

O método heurístico conhecido como método Kilbridge e Wester tem semelhanças com a heurística anterior. A principal diferença é que em vez de se atribuir tanta importância ao tempo de cada operação esta foca-se mais nas relações de precedência entre estações. Em concordância com o diagrama de precedências, distribuem-se as operações em colunas, sendo que na primeira coluna se colocam as operações sem precedências, na segunda as operações com precedências relativamente às operações colocadas na primeira coluna e assim sucessivamente. Posteriormente as operações são afectas às estações de trabalho, começando pelas operações que se encontram na primeira coluna, depois na segunda, e assim sucessivamente. Mais uma vez, o somatório do tempo ocupado pelas tarefas em cada estação não pode ultrapassar o tempo de ciclo.

As etapas a cumprir para aplicar esta heurística são, resumidamente, as seguintes:

- Construir um diagrama de precedências de modo a que operações com idêntica precedência sejam colocadas verticalmente em colunas;
- Listar os elementos seguindo uma ordem crescente de colunas e listar também os tempos de operação e o somatório dos tempos de operação de cada coluna;
- Atribuir elementos a estações começando pelas operações das colunas de ordem inferior;
- Repetir este processo até atingir o tempo de ciclo.

3.3.2. APLICAÇÕES DE SOFTWARE PARA BALANCEAMENTO DE LINHAS

Existem várias aplicações informáticas que oferecem soluções de optimização para linhas de montagem. De entre elas, referem-se neste trabalho as aplicações Probalance e AviX® Balance.

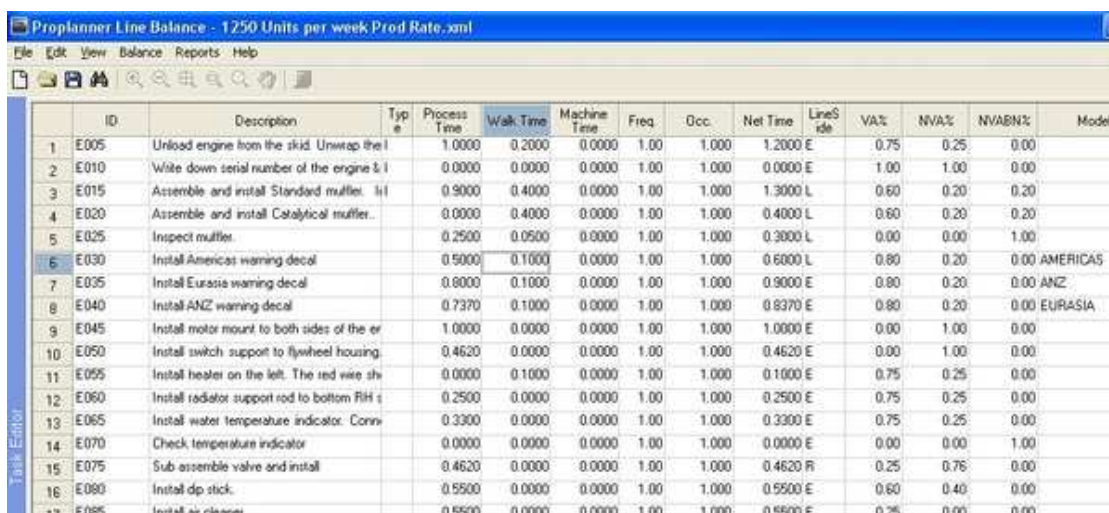
3.3.2.1. PROBALANCE

A Proplanner é uma empresa de software que dispõe de vários programas ligados ao processo de produção dentro da engenharia e gestão, sendo um desses programas, o ProBalance, direccionado ao balanceamento de linhas [5].

Esta aplicação permite obter diferentes soluções de balanceamento de linhas através dos dados do processo em questão. Existem dois métodos de balanceamento de linhas disponíveis no software ProBalance. Os utilizadores podem seleccionar minimizar o número de estações de trabalho (balanceamento tipo 1) ou minimizar o tempo de ciclo das operações (balanceamento tipo 2). O BL tipo 1 requer que o utilizador introduza o tempo de ciclo, a partir do qual a aplicação calcula quantas estações de trabalho são necessárias. O BL tipo 2 requer que o utilizador introduza o número de estações de trabalho disponíveis, ou desejadas, e através desse valor é calculado o tempo de ciclo.

São necessárias duas tarefas para completar o BL usando o ProBalance. Primeiro, o utilizador necessita de especificar os dados do problema (exemplo: ID, tempo de processo, precedências). Depois de introduzidos os dados, a segunda tarefa é criar a linha balanceada (manualmente ou gerada pelo ProBalance) baseada nos dados introduzidos pelo utilizador.

Na Figura 5 é possível ver a interface que permite a introdução das tarefas e das suas características, tais como o tempo de operação e respectivas precedências.



The screenshot shows the 'Task Editor' window of the ProBalance software. The window title is 'Proplanner Line Balance - 1250 Units per week Prod Rate.xml'. The menu bar includes 'File', 'Edit', 'View', 'Balance', 'Reports', and 'Help'. Below the menu bar is a toolbar with various icons. The main area contains a table with the following columns: ID, Description, Type, Process Time, Walk Time, Machine Time, Freq, Occ, Net Time, Line Side, VA%, NVA%, NVABN%, and Models. The table lists 17 tasks (E005 to E085) with their respective descriptions and values for each column.

ID	Description	Type	Process Time	Walk Time	Machine Time	Freq	Occ	Net Time	Line Side	VA%	NVA%	NVABN%	Models
1	E005 Unload engine from the skid. Unwrap the l		1.0000	0.2000	0.0000	1.00	1.000	1.2000	E	0.75	0.25	0.00	
2	E010 Write down serial number of the engine & l		0.0000	0.0000	0.0000	1.00	1.000	0.0000	E	1.00	1.00	0.00	
3	E015 Assemble and install Standard muffler. In		0.9000	0.4000	0.0000	1.00	1.000	1.3000	L	0.60	0.20	0.20	
4	E020 Assemble and install Catalytic muffler...		0.0000	0.4000	0.0000	1.00	1.000	0.4000	L	0.60	0.20	0.20	
5	E025 Inspect muffler.		0.2500	0.0500	0.0000	1.00	1.000	0.3000	L	0.00	0.00	1.00	
6	E030 Install Americas warning decal		0.5000	0.1000	0.0000	1.00	1.000	0.6000	L	0.80	0.20	0.00	AMERICAS
7	E035 Install Eurasia warning decal		0.8000	0.1000	0.0000	1.00	1.000	0.9000	E	0.80	0.20	0.00	ANZ
8	E040 Install ANZ warning decal		0.7370	0.1000	0.0000	1.00	1.000	0.8370	E	0.80	0.20	0.00	EURASIA
9	E045 Install motor mount to both sides of the er		1.0000	0.0000	0.0000	1.00	1.000	1.0000	E	0.00	1.00	0.00	
10	E050 Install switch support to flywheel housing.		0.4620	0.0000	0.0000	1.00	1.000	0.4620	E	0.00	1.00	0.00	
11	E055 Install heater on the left. The red wire sh		0.0000	0.1000	0.0000	1.00	1.000	0.1000	E	0.75	0.25	0.00	
12	E060 Install radiator support rod to bottom RH s		0.2500	0.0000	0.0000	1.00	1.000	0.2500	E	0.75	0.25	0.00	
13	E065 Install water temperature indicator. Conn		0.3300	0.0000	0.0000	1.00	1.000	0.3300	E	0.75	0.25	0.00	
14	E070 Check temperature indicator		0.0000	0.0000	0.0000	1.00	1.000	0.0000	E	0.00	0.00	1.00	
15	E075 Sub assemble valve and install		0.4620	0.0000	0.0000	1.00	1.000	0.4620	R	0.25	0.75	0.00	
16	E080 Install dip stick.		0.5500	0.0000	0.0000	1.00	1.000	0.5500	E	0.60	0.40	0.00	
17	E085 Install air cleaner		0.5500	0.0000	0.0000	1.00	1.000	0.5500	E	0.25	0.00	0.00	

Figura 5 Interface do software ProBalance

Na Figura 6 e Figura 7 é possível ver o gráfico de precedências das operações e o gráfico com os resultados do balanceamento da linha.

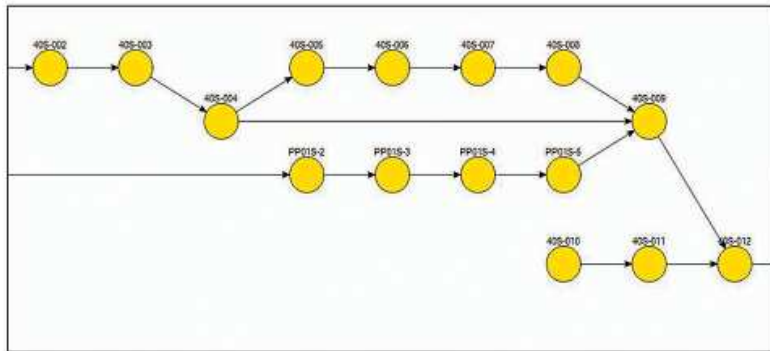


Figura 6 Gráfico de precedências

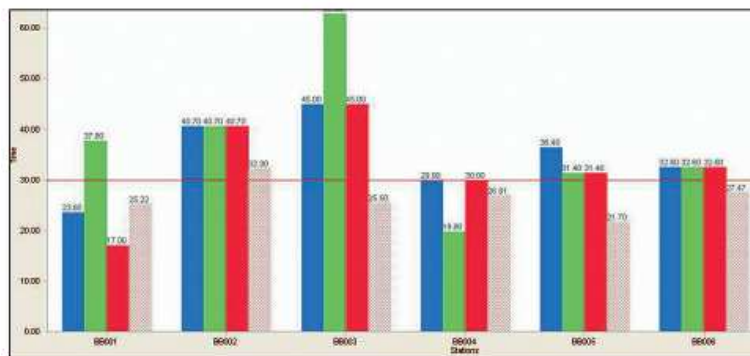


Figura 7 Gráfico de balanceamento

Através de algoritmos de atribuição automática de tarefas e recursos, a aplicação gera uma solução ao problema introduzido pelo utilizador. Estes algoritmos funcionam através de métodos de optimização. Quando a aplicação gera de forma automática uma solução esta lida com as seguintes restrições:

- Área de trabalho - lado da linha na qual essa tarefa deve ser atribuída.
- Precedências - ordem pela qual tarefas devem ser executadas.
- Agrupamento - tarefas que precisam de se mover juntas para a mesma estação, ou mesmo operador.
- Recursos - tarefas que requerem um recurso e, portanto, devem ser atribuídas a uma estação com esse recurso.
- Operadores - são suportados múltiplos operadores por estação, e cada operador pode ser atribuído a uma, ou mais, estações de trabalho.

Quando o balanceamento é feito manualmente pelo utilizador o ProBalance exhibe todas as violações efectuadas durante o processo, baseando-se novamente nas restrições previamente apresentadas.

3.3.2.2. AViX® BALANCE

O AviX® Balance é uma aplicação de balanceamento de linhas de produção, que ajuda a otimizar a linha de produção. Com os dados da aplicação é possível criar e simular novas linhas de produção para os produtos [6].

A aplicação permite ao utilizador:

- Ver os resultados do balanceamento directamente no ecrã;
- Balancear vários produtos em simultâneo;
- Facilidade na optimização da produção e aumento da eficiência da produção;
- Obter *layouts* sem perdas;
- Balancear com o controlo dos métodos de trabalho, objectos, componentes e ferramentas.

Esta aplicação permite efectuar o balanceamento manualmente através da deslocação de tarefas entre estações. Depois de cada passo tomado no balanceamento, a aplicação mostra a sua influência na produtividade e na eficiência da linha de produção.

Na Figura 8 é possível ver a janela principal desta aplicação: no canto superior esquerdo encontram-se as estações de trabalho e na parte inferior direita o gráfico da linha com os tempos correspondentes de cada estação.



Figura 8 Interface da aplicação AviX® Balance

A aplicação permite ainda modificar o gráfico e visualizar as operações correspondentes em cada coluna, ou seja cada estação de trabalho. É possível ver essa opção na Figura 9.

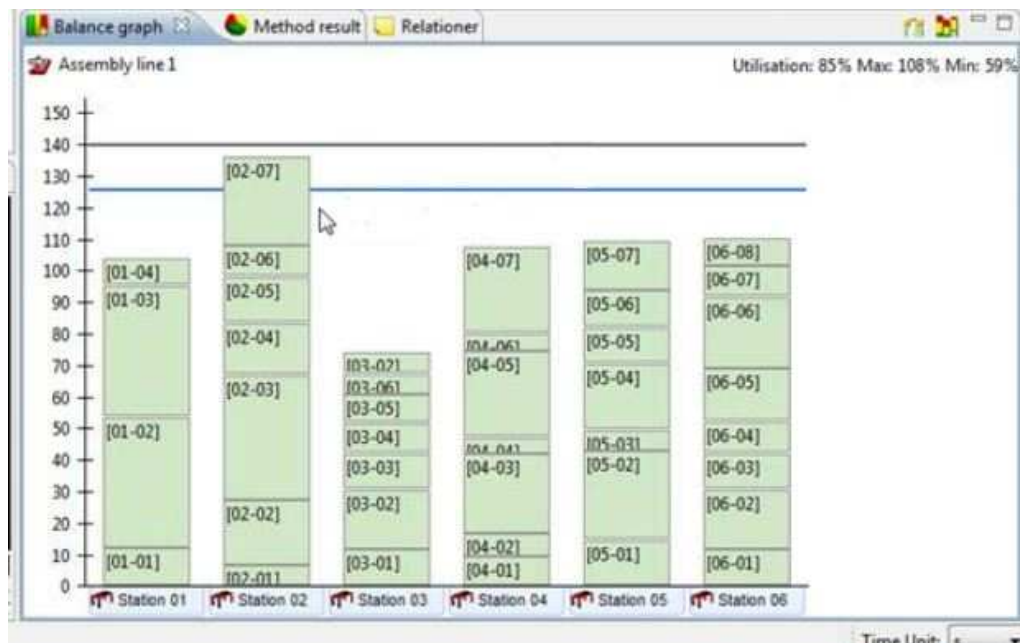


Figura 9 Opção do gráfico do balanceamento

Depois da linha balanceada é possível obter vários relatórios sobre a linha. Na Figura 10 mostra-se um exemplo de um deles onde é possível ver os tempos de operação nas estações.

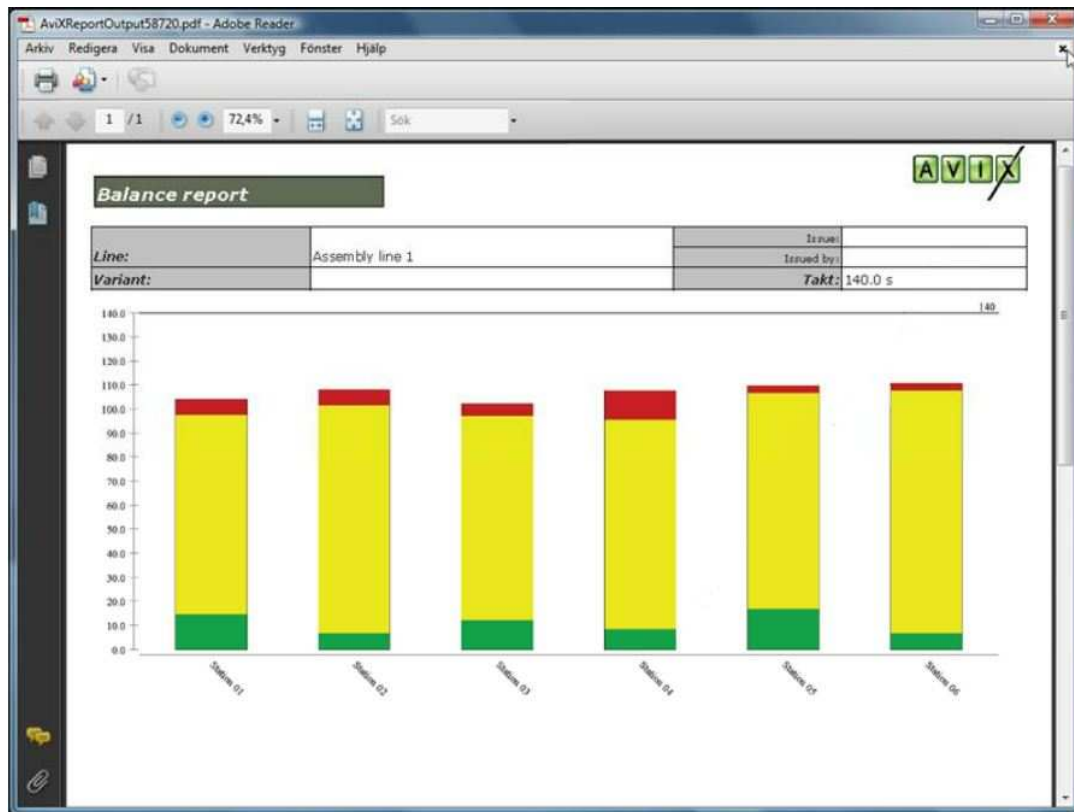


Figura 10 Relatório de balanceamento

3.4. PROJECTOS DE BALANCEAMENTO DE LINHAS USANDO AG

Nesta secção são descritos dois trabalhos desenvolvidos para balanceamento de linhas usando Algoritmos Genéticos.

3.4.1. ALGORITMO GENÉTICO PARA O BALANCEAMENTO DE LINHAS DE PRODUÇÃO PARA PRODUTO ÚNICO

Neste projecto, desenvolvido por Mayerle e Santos é discutido o problema do balanceamento de linhas de produção simples (produto único), o qual consiste na obtenção de uma alocação de um conjunto de tarefas aos postos (ou estações) de trabalho, respeitando a relação de precedência na execução destas tarefas. São consideradas duas situações distintas: reduzir o número de postos de trabalho, considerando o tempo de ciclo que satisfaça a procura, ou reduzir o tempo de ciclo, dado o número de postos de trabalho.

Embora, os autores não terem realizado testes exaustivos com o algoritmo proposto, o seu desempenho ficou próximo de ótimo, por outro lado sugerem que seja ampliado o número de testes realizados com o algoritmo, incluindo problemas que tenham soluções por outros métodos heurísticos encontrados na literatura [7].

3.4.2. **BALANCEAMENTO DE LINHAS DE PRODUÇÃO DE MODELO MISTO COM ALGORITMO GENÉTICO HÍBRIDO**

Tendo em conta o problema existente do modelo misto de balanceamento de linhas de produção, no qual consiste numa linha de produção onde existem vários produtos de semelhantes características e os quais são produzidos conforme a necessidade do cliente.

É proposto por Ying, Hongshun e Liao um modelo matemático baseado em dois factores, que são integrados, incluindo o número de estações de trabalho e a eficiência da linha de montagem.

De seguida, é desenvolvido um novo algoritmo genético híbrido para encontrar a solução ótima dos problemas. Para evitar que o problema se encaminhe para uma solução de forma prematura e melhorar a optimização, o GA é combinado com um simulador de algoritmos de reconhecimento (*simulated annealing algorithms*). Os resultados da simulação indicam que, desta forma, o algoritmo híbrido tem uma melhor eficiência de optimização e de desempenho [8].

4. ALGORITMOS GENÉTICOS

Neste capítulo são introduzidos os Algoritmos Genéticos, sendo descrito o seu aparecimento e desenvolvimento histórico, definido o que é um Algoritmo Genético, esclarecida a sua estrutura e o seu funcionamento.

Os AG foram inventados por John Holland, na década de 1960, e desenvolvidos pelo próprio e pelos seus colegas e estudantes, na Universidade de Michigan, nos anos 60 e 70. O objectivo de Holland não era projectar algoritmos para resolver problemas específicos, mas sim estudar formalmente o fenómeno da adaptação, tal como ocorre na natureza, e desenvolver maneiras através das quais os mecanismos de adaptação da natureza pudessem ser importados para sistemas computacionais. O livro de Holland de 1975 *Adaptation in Natural and artificial Systems* apresentou o algoritmo genético como uma abstracção da evolução biológica e deu um panorama teórico para a adaptação nos AG [9, 10].

4.1. DEFINIÇÃO E FUNDAMENTOS BIOLÓGICOS DOS ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos são uma família de modelos computacionais inspirados na genética e na evolução natural das espécies. Estes algoritmos modelam uma solução para um problema específico numa estrutura de dados, como a de um cromossoma, e aplicam operadores que recombina estas estruturas preservando informações críticas.

Como principais aspectos de um algoritmo genético podem-se referir os cromossomas e a reprodução. Os primeiros são a representação das soluções para o problema que se pretende resolver e o segundo permite a evolução das soluções no sentido de atingirem a solução final.

- **Cromossomas**

Todos os organismos vivos são constituídos por células. Em cada célula há um mesmo conjunto de **cromossomas**. Estes cromossomas são cadeias de ADN e servem como modelo para todo o organismo. O cromossoma é constituído por **genes**, que são blocos de ADN. Cada gene codifica uma determinada proteína. Basicamente, pode-se dizer que cada gene codifica uma determinada **feição** como, por exemplo, cor dos olhos. Conjuntos de genes relacionados com determinada feição (por exemplo, olhos azuis, castanhos) são chamados **alelos**. Cada gene tem uma posição própria dentro do cromossoma. Essa posição é chamada **locus**.

- **Reprodução**

Durante a reprodução, ocorre inicialmente **recombinação** (ou **cruzamento**). Os genes dos pais são combinados para formar um novo cromossoma. Essa descendência recém-criada pode sofrer uma mutação. **Mutação** significa que os elementos do ADN são ligeiramente modificados. Essas mudanças são causadas principalmente por erros na cópia dos genes dos pais.

A **adaptação** de um organismo é medida pelo sucesso do organismo na vida (sobrevivência).

4.2. ESPAÇO DE SOLUÇÕES

Quando se está a resolver um problema, geralmente procura-se alguma solução que seja melhor do que outras. O espaço de todas as soluções possíveis (que é o conjunto no qual se encontra a solução procurada) é chamado espaço das soluções (ou espaço de Estados). Cada ponto desse estado representa uma solução possível. Cada uma das possíveis soluções pode ser "marcada" pelo seu valor (ou adequação) para o problema. Com os AG procura-se a melhor solução dentro de um número de possíveis soluções – representadas por pontos no Espaço de Soluções.

Procurar uma solução resume-se a procurar algum valor extremo (um mínimo ou máximo) no espaço de soluções. Na Figura 11 é visível um exemplo de um espaço de estados. Por vezes o espaço de soluções pode estar bem definido, mas usualmente conhecem-se apenas poucos pontos do espaço de soluções. Com os AG o processo de encontrar soluções gera outros pontos (possíveis soluções) à medida que a evolução ocorre.

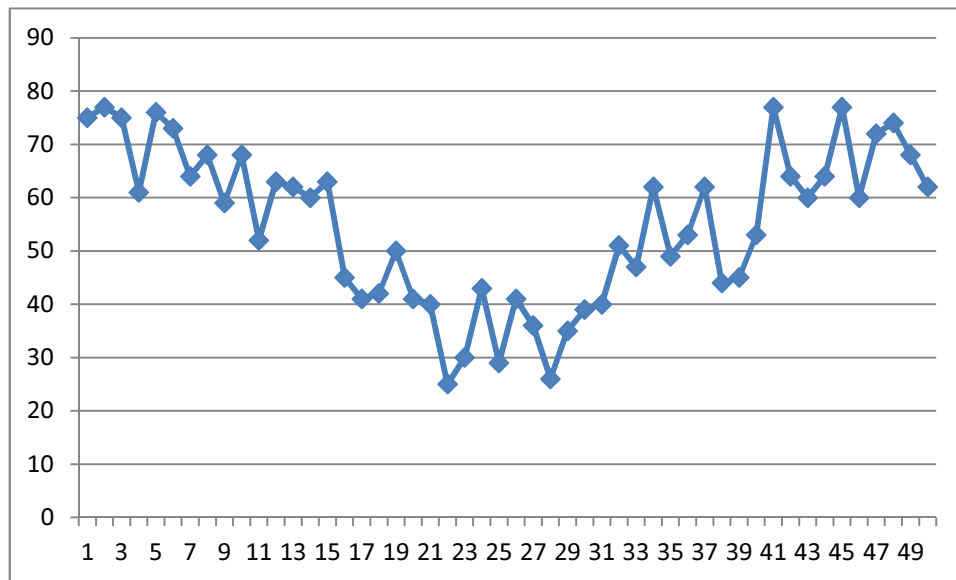


Figura 11 Exemplo de espaço de soluções

O problema é que cada pesquisa pode ser muito complicada. Por exemplo, pode não se saber por onde procurar uma solução, ou por onde começar. Há muitos métodos que podem ser usados para encontrar uma solução adequada, mas esses métodos não fornecerão necessariamente a melhor solução.

4.3. ESTRUTURA DE UM ALGORITMO GENÉTICO

O algoritmo começa por gerar um conjunto inicial de soluções (representadas por cromossomas) chamadas de população inicial. Soluções de uma população são utilizadas para formar uma nova população. Isto é motivado pela esperança que a nova população (descendência) seja melhor do que a anterior (ascendência). As soluções que são seleccionadas para formar novas gerações de soluções são seleccionadas de acordo com a sua adequação – quanto melhores, mais hipóteses de reprodução terão. Esse processo é repetido até que alguma condição seja satisfeita (por exemplo, o número de gerações ou o aperfeiçoamento da melhor solução).

O funcionamento básico de um Algoritmo Genético pode ser descrito pelo seguinte algoritmo:

1. **[Início]** Gerar uma população aleatória de n cromossomas (soluções adequadas para o problema);
2. **[Adequação]** Avaliar a adequação $f(x)$ de cada cromossoma x da população;
3. **[Nova população]** Criar uma nova população repetindo os passos seguintes até que a nova população esteja completa:
 1. **[Seleção]** Seleccionar, de acordo com sua adequação (melhor adequação, mais chances de ser seleccionado), dois cromossomas para serem os pais;
 2. **[Cruzamento]** Com a probabilidade de cruzamento definida, cruzar os pais para formar a nova geração. Se não realizar cruzamento, a nova geração será uma cópia exacta dos pais;
 3. **[Mutação]** Com a probabilidade de mutação definida, alterar os cromossomas da nova geração nos Locus (posição nos cromossomas);
 4. **[Aceitação]** Colocar a nova descendência na nova população.
4. **[Substituir]** Utilizar a nova população gerada para a próxima iteração do algoritmo;
5. **[Teste]** Se a condição final foi atingida, **parar**, e retornar a melhor solução da população actual;
6. **[Repetir]** Ir para o passo 2.

4.4. MECANISMOS DE LIGAÇÃO DE UM ALGORITMO GENÉTICO AO PROBLEMA

Existem dois mecanismos que ligam o Algoritmo Genético ao problema que este pretende resolver: o primeiro mecanismo é a codificação do problema em cromossomas; o segundo mecanismo é a função de avaliação que calcula o valor de cada cromossoma no contexto do problema.

4.4.1. CODIFICAÇÃO DO PROBLEMA

O primeiro mecanismo, a codificação do problema, varia de problema para problema e entre algoritmos aplicados. A codificação é feita em vectores e pode ser feita em vários formatos: codificação binária, codificação por permutação, codificação de valores e codificação em árvore. A escolha da codificação é efectuada com a análise do problema, sendo escolhida a mais adequada ao problema. De seguida são ilustradas duas das principais codificações.

4.4.1.1. CODIFICAÇÃO BINÁRIA

A codificação binária é a mais comum, principalmente porque foi a que os primeiros investigadores de AG usaram e devido à sua relativa simplicidade.

Na codificação binária cada cromossoma é uma série de bits – 0 ou 1. A título de exemplo, apresentam-se de seguida dois possíveis cromossomas codificados utilizando esta técnica:

- Cromossoma A – 101100101100101011100101
- Cromossoma B – 111111100000110000011111

A codificação binária permite muitos possíveis cromossomas, mesmo com pequenos números de alelos. Por outro lado, esta codificação não é natural para muitos problemas e algumas vezes é necessário fazer correcções antes dos cruzamentos e/ou mutações.

4.4.1.2. CODIFICAÇÃO POR PERMUTAÇÃO

A codificação por permutação pode ser usada em problemas que envolvem ordenação, tal como o "Problema do Caixeiro-Viajante", ou problemas de ordenação de tarefas.

Na codificação por permutação cada cromossoma é uma série de números que representam uma posição numa sequência. A título de exemplo, apresentam-se de seguida dois possíveis cromossomas codificados utilizando esta técnica:

- Cromossoma A – 1 5 3 2 6 4 7 9 8
- Cromossoma B – 8 5 6 7 2 3 1 4 9

A codificação por permutação é útil para a solução de problemas de ordenação. Para alguns tipos de cruzamentos e mutações são necessárias correcções para que os cromossomas fiquem consistentes (isto é, contenham sequências reais) para alguns problemas.

4.4.2. FUNÇÃO DE AVALIAÇÃO

O segundo mecanismo, a função de avaliação, é a ligação entre o problema a ser resolvido e o Algoritmo Genético. A função de avaliação recebe como entrada o cromossoma e retorna um valor, ou uma lista de valores, que medem o desempenho do cromossoma no

problema a ser resolvido. Esse valor, ou lista de valores, é a aptidão do cromossoma, que o algoritmo usa para avaliar os cromossomas e para a reprodução de novos cromossomas. A função de avaliação é definida de acordo com a solução desejada do problema, sendo esta função que faz desenvolver o algoritmo até uma solução óptima.

4.5. OPERADORES DO ALGORITMO GENÉTICO

Como vimos anteriormente, na construção básica do Algoritmo Genético o cruzamento e a mutação são os operadores mais importantes, sendo o seu desempenho influenciado principalmente por esses dois operadores.

4.5.1. CRUZAMENTO

Depois de se decidir que codificação usar, pode-se proceder à operação de cruzamento. O cruzamento opera em determinados genes dos cromossomas dos pais e cria novas descendências. A maneira mais simples de fazer isso é escolher aleatoriamente alguns pontos de cruzamento e copiar para o filho todos os genes que se encontram antes desse ponto de um dos pais e todos os genes que vêm depois desse ponto do outro pai.

A Figura 12 ilustra o processo de cruzamento (“|” representa o ponto de cruzamento):

Cromossoma 1	11011 00100110110
Cromossoma 2	10111 11000011110
Descendência 1	11011 11000011110
Descendência 2	10111 00100110110

Figura 12 Exemplo da operação de cruzamento de dois cromossomas

Existem outras maneiras de efectuar o cruzamento. Por exemplo, podem-se escolher mais pontos de cruzamento. O cruzamento pode ser um pouco complicado e depender principalmente da codificação dos cromossomas. De referir que cruzamentos específicos, desenvolvidos para problemas específicos, podem melhorar o desempenho dos algoritmos genéticos.

4.5.2. MUTAÇÃO

Depois de realizar um cruzamento procede-se à mutação. A mutação tem a intenção de prevenir que todas as soluções do problema dessa população caíam num ponto óptimo

local. A operação de mutação muda aleatoriamente a descendência criada pelo cruzamento. No caso de uma codificação binária, pode-se mudar aleatoriamente algum *bit* escolhido de 1 para 0 e vice-versa. Apresenta-se na Figura 13 um exemplo da realização de uma operação de mutação em dois cromossomas:

Descendência Original 1	1101111000011110
Descendência Original 2	1101100100110110
Descendência Mutada 1	1100111000011110
Descendência Mutada 2	1101101100110110

Figura 13 Exemplo da realização de uma mutação de cromossomas

A técnica da mutação (da mesma forma que a do cruzamento) depende muito da codificação dos cromossomas. Por exemplo, quando se codificam permutações, as mutações podem ser realizadas como uma troca de dois genes.

4.5.3. SELECÇÃO

A selecção é outro operador importante, e também com elevada influência no resultado, pois é o processo através do qual os cromossomas da população são seleccionados para depois se proceder ao seu cruzamento. Existem vários métodos de selecção, nomeadamente: selecção aleatória por roleta, selecção por classificação, selecção por torneio, selecção Boltzman, selecção por estado estacionário, entre outras.

Seguidamente são listados os principais processos de selecção do ciclo do Algoritmo Genético, sendo descrito o funcionamento dos mesmos.

- **Seleção aleatória por roleta**

Este método é um dos mais usados em Algoritmos Genéticos. Neste tipo de selecção é atribuída aos indivíduos (cromossomas) uma probabilidade de serem seleccionados, que é directamente proporcional à sua aptidão. O problema deste método é se existir uma grande discrepância na probabilidade, o que faz com que existam cromossomas com probabilidades de sair muito baixas e outros com probabilidades elevadas, sendo então maioritariamente seleccionados sempre os mesmos cromossomas.

A Tabela 2 mostra um exemplo deste tipo de selecção, assumindo que existem quatro cromossomas, com aptidões diferentes, é atribuída uma probabilidade a cada um dos cromossomas.

Tabela 2 Cromossomas exemplo com aptidão e probabilidade de selecção

Cromossoma	Aptidão	Probabilidade de selecção
A	4	3,74 \approx 4%
B	52	48,60 \approx 48%
C	34	31,78 \approx 32%
D	17	15,89 \approx 16%

A Figura 14 ilustra um gráfico que representa a probabilidade de selecção de cada cromossoma, sendo possível verificar que o cromossoma com maior aptidão tem uma probabilidade muito superior de ser seleccionado, comparativamente ao que apresenta uma aptidão inferior.

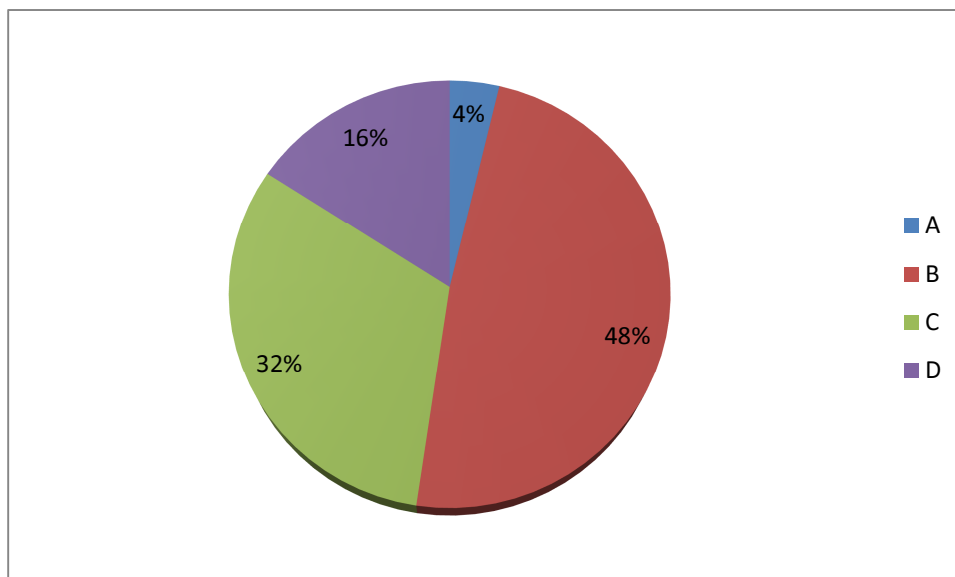


Figura 14 Probabilidade de selecção dos cromossomas utilizando o método de selecção aleatória por roleta

- **Seleccção por classificação**

A selecção por classificação classifica primeiro a população e, de seguida, atribui a cada cromossoma um valor de adequação determinado pela sua classificação. O cromossoma pior classificado terá adequação igual a 1, o segundo pior igual a 2, etc., de forma que o melhor terá adequação igual a N (número de cromossomas na população), permitindo que todos os cromossomas tenham hipótese de serem seleccionados. No entanto, este método pode resultar em menor convergência, pois os melhores cromossomas não têm grande distinção em relação aos outros.

Para melhor compreensão é apresentado um exemplo deste tipo de selecção, usando uma população equivalente a usada no exemplo anterior mas atribuindo uma classificação a cada cromossoma, tal como representado na Tabela 3.

Tabela 3 População exemplo de cromossomas, com respectiva aptidão e classificação

Cromossoma	Aptidão	Classificação	Probabilidade de selecção
A	5	1	10%
B	50	4	40%
C	30	3	30%
D	15	2	20%

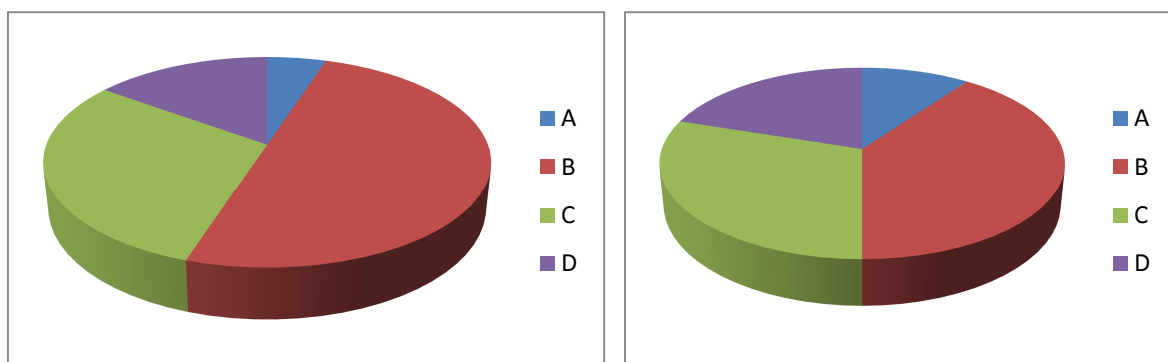


Figura 15 Probabilidade de selecção antes de classificação (esquerda) e probabilidade depois da classificação (direita).

Como é possível confirmar através dos gráficos apresentados na Figura 15, a probabilidade de selecção dos cromossomas com aptidões inferiores é superior depois de feita a classificação, tornando a selecção destes mais provável do que anteriormente. No entanto, como referido, este método pode resultar em menor convergência, porque os melhores indivíduos não se distinguem muito dos outros.

- **Seleção por Torneio**

Na selecção por torneio, são inicialmente seleccionados três cromossomas aleatoriamente; depois destes seleccionados, é verificado qual dos três contém a aptidão mais elevada, sendo esse seleccionado para o cruzamento. Posteriormente é executado o mesmo processo para se obter o segundo cromossoma para o cruzamento.

4.6. CRITÉRIOS DE PARAGEM

O algoritmo pára quando é alcançado (ou aproximado) um objectivo definido para o problema. De entre os diferentes objectivos usualmente considerados como critério de paragem, contam-se os seguintes:

- **Máximo de gerações** – O AG pára quando atinge o número máximo de gerações definido para o algoritmo.
- **Tempo decorrido** – O AG pára quando o algoritmo atingir o tempo de execução estipulado.
- **Limite da aptidão** – O AG pára quando o valor de aptidão do melhor indivíduo (cromossoma) da população actual for inferior ou igual ao limite da aptidão, no caso de se estar a proceder a uma minimização.
- **Geração *stall*** – O AG pára se não existir melhoria na função objectivo numa sequência de gerações seguidas (gerações de comprimento *stall*).

5. ARQUITECTURA DA APLICAÇÃO A IMPLEMENTAR

Neste capítulo é descrito o Algoritmo Genético desenvolvido em todos os seus aspectos, e são explicadas todas as configurações do mesmo. É ainda apresentada a interface gráfica desenvolvida de raiz para servir de ligação do problema ao algoritmo, simplificando a introdução dos problemas e dados dos mesmos, assim como para apresentar os resultados ao utilizador.

5.1. ALGORITMO GENÉTICO DESENVOLVIDO

O primeiro passo do desenvolvimento de um Algoritmo Genético é a codificação do problema a cuja resolução o algoritmo se destina. Segue-se a geração da população inicial, o processo de selecção e as operações de cruzamento e mutação. Por último, há a considerar ainda a função de avaliação e a escolha do critério de paragem.

5.1.1. ESTRUTURA DO ALGORITMO GENÉTICO DESENVOLVIDO

Na Figura 16 é apresentado um fluxograma da estrutura geral do Algoritmo Genético desenvolvido. Neste fluxograma são apresentadas as fases do AG, fases estas que são explicadas nas subsecções seguintes.

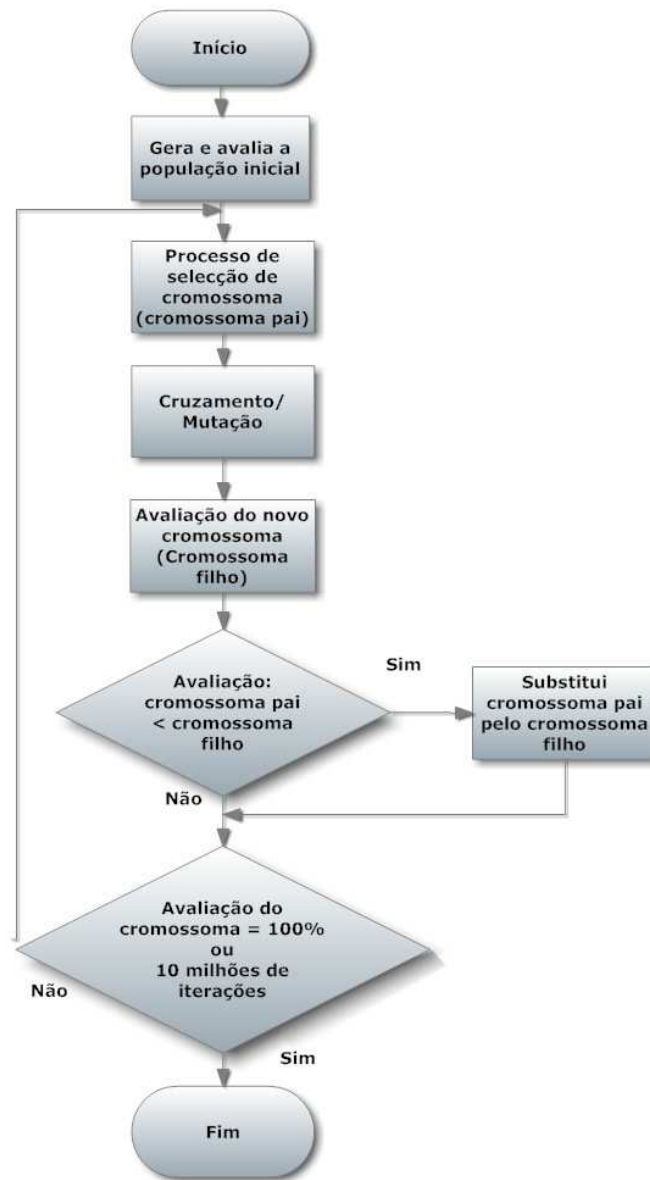


Figura 16 Fluxograma do Algoritmo Genético desenvolvido

5.1.2. CODIFICAÇÃO

Para a codificação dos problemas de balanceamento de linhas de produção, a codificação seleccionada foi aquela que permite uma simples associação do cromossoma ao sequenciamento da linha de montagem, possibilitando uma rápida avaliação do cromossoma como possível solução do problema.

Para tal foi associado a cada tarefa um valor. Considere-se, por exemplo, a Tabela 4 que representa a codificação de uma linha de produção com 5 tarefas.

Tabela 4 Exemplo de codificação das tarefas.

Tarefa	Valor de codificação
A	1
B	2
C	3
D	4
E	5

5.1.3. GERAÇÃO DA POPULAÇÃO INICIAL

A geração da população inicial é feita aleatoriamente.

Inicialmente foi pensado no seguinte processo para gerar a população. Este primeiro método gera um número aleatório entre 0 e o número de operações e este valor é atribuído ao primeiro gene do cromossoma. De seguida é gerado outro número aleatório, cumprindo os requisitos especificados anteriormente. É então verificado se este número já foi atribuído a algum dos genes do cromossoma. Caso tenha sido, é feita nova geração de um número aleatório, este é testado novamente, e este processo repete-se até ser encontrado um valor ainda não atribuído. O processo descrito anteriormente repete-se até o cromossoma estar completo, e é repetido para toda a população.

Depois de implementado este método, verificou-se que este processo é lento, o que leva a que o algoritmo demore a gerar a população inicial.

Devido a este problema foi então desenvolvido e aplicado outro método, que se passa a descrever.

Neste segundo método o cromossoma é preenchido completamente, começando por se atribuir o valor de um ao primeiro gene, e incrementando o valor em cada gene até ser preenchido o cromossoma. Assim, se o problema consistir em cinco operações, o cromossoma será preenchido de forma organizada, com os valores inteiros de 1 a 5. Depois do algoritmo ter preenchido o cromossoma, este vai ser “baralhado”. Este processo é efectuado através da troca de genes, seleccionando posições aleatórias. Para efectuar esta troca, são seleccionadas duas posições aleatórias e os genes correspondentes a essas posições são trocados. Este processo é executado x vezes, sendo que o valor de x é calculado usando o número de operações do problema e uma constante definida, que permitiu variar o valor de x para testar qual seria o valor mais eficaz e simultaneamente mais rápido. Chegou-se ao valor de 3 para a constante; isto quer dizer que então x é igual ao número de operações multiplicado por 3, o que permite que o número de vezes que o cromossoma seja baralhado seja directamente ligado ao número de operações do problema. Na Tabela 5 é possível ver um pequeno exemplo de um cromossoma a ser baralhado, sendo apenas representadas quatro das quinze trocas que, de acordo com o método implementado, são executadas. Este método foi então implementado e é o executado para gerar a população inicial.

Tabela 5 Processo executado para baralhar o cromossoma

Cromossoma Gerado	1	2	3	4	5
1ª Troca	1	4	3	2	5
2ª Troca	5	4	3	2	1
3ª Troca	4	5	3	2	1
4ª Troca	4	5	1	2	3

5.1.4. SELECÇÃO

O processo de selecção implementado no AG desenvolvido é o método de selecção por torneio, previamente explicado. São então seleccionados três cromossomas da população, de forma aleatória, e é comparada a aptidão de cada um deles, sendo seleccionado o cromossoma que contém a aptidão superior.

5.1.5. FUNÇÃO DE AVALIAÇÃO

A função de avaliação tem o propósito de, tal como o nome indica, avaliar cada cromossoma em função do objectivo do problema.

No balanceamento de uma linha de montagem, deve-se garantir que as precedências entre as operações sejam cumpridas e que o tempo de ciclo seja garantido, o que por si só faz com que o número de estações necessárias seja estabelecido. Tomando por base estes conhecimentos, foi desenvolvida uma função de avaliação que avalia os cromossomas através da verificação do cumprimento das precedências e do número de estações que cada possível solução requer.

Inicialmente é feita uma contagem do número de precedências existentes no problema. Este valor serve de referência para verificar qual a percentagem de inviabilidades. Depois, ao cromossoma ao qual é aplicada a função de avaliação, é verificado se as precedências são cumpridas em cada gene do cromossoma; por cada precedência não cumprida é

incrementada uma variável de penalização (a variável *penalidade*). Durante esse processo é somada a duração temporal de cada tarefa que está a ser verificada, e quanto o valor acumulado de tempo excede o valor de tempo de ciclo pretendido, sabe-se que temos uma estação. No final é verificado em quantas estações a solução analisada excede o número desejado de estações, e por cada estação a mais é incrementada a variável de penalização.

$$Aptidão = \frac{Max_{penalidade} - penalidade}{Max_{penalidade}} \times 100 \quad (6)$$

Na equação 6 é apresentada a expressão de cálculo da aptidão, que contém a referida variável $Max_{penalidade}$. Esta variável representa o máximo de penalidade que o problema contém. Por sua vez a variável *penalidade*, é a variável incrementada por cada precedência não cumprida ou estação a mais.

5.1.6. CRUZAMENTO E MUTAÇÃO

Neste problema específico, e devido à codificação utilizada, as operações de cruzamento não podem ser efectuadas, pois iriam gerar soluções inviáveis. Como cada solução, ou seja cromossoma, contém valores inteiros entre 1 e o número de operações do problema em questão, nenhum valor se repete dentro do cromossoma. Utilizando o cruzamento, só é possível garantir que não se repitam as mesmas operações dentro do cromossoma no caso dos cromossomas pai serem exactamente iguais; em qualquer outro caso iriam surgir cromossomas filhos cujas soluções seriam inviáveis.

Quanto à mutação, foram testados três métodos de mutação tendo sido aplicado o que garantiu melhores resultados.

O primeiro método consiste na troca dos dois genes seleccionados, através da selecção de duas posições aleatórias no cromossoma.

Como se pode observar na Tabela 6 foram seleccionadas as posições 2 e 5 do cromossoma e os genes correspondentes a essas posições são trocados.

Tabela 6 Exemplo de aplicação do primeiro método de mutação testado.

Antes	1	3	4	2	5
Depois	1	5	4	2	3

No segundo método implementado é apenas seleccionada uma posição, novamente de forma aleatória, e é efectuada a troca entre o gene da posição seleccionada e o gene consecutivo à posição seleccionada (gene que se encontra na posição à direita da posição aleatória, tal como se pode observar na Tabela 7.

Tabela 7 Exemplo de aplicação do segundo método de mutação testado.

Antes	1	3	4	2	5
Depois	1	4	3	2	5

Por último, apresenta-se o terceiro método implementado e que é o utilizado na aplicação, pois foi o que apresentou melhores resultados. Este método consiste novamente em seleccionar duas posições do cromossoma de forma aleatória, e o gene que se encontra na primeira posição seleccionada é movido para a segunda posição seleccionada, e todas as operações entre estas duas posições são deslocados de uma posição no sentido de deslocação entre a segunda posição e a primeira posição. Este processo é exemplificado na Tabela 8, onde é possível verificar a troca efectuada.

Tabela 8 Exemplo de aplicação do terceiro método de mutação testado.

Antes	1	3	4	2	5
Depois	1	4	2	3	5

Depois de seleccionado o terceiro método de mutação para implementação, pois apresentava os melhores resultados, foi ainda aplicada uma melhoria ao sistema de mutação, que é a mutação dinâmica. Esta consiste em proceder ao processo de mutação y vezes, sendo que o valor de y é calculado e dependente do número de operações que a linha contém e do número total de precedências existentes, ou seja quanto mais operações e precedências existirem no problema mais mutações são efectuadas. O valor de y é calculado usando a equação apresentada na equação 7.

$$y = (\text{número de operações}/4) + (\text{número de precedências}/4) \quad (7)$$

Tal como é apresentado na equação 7 o número de mutações é igual $\frac{1}{4}$ do número de operações, mais $\frac{1}{4}$ do número de precedências das operações. Esta alteração diminui o número de iterações necessárias para encontrar uma solução.

5.1.7. CRITÉRIO DE PARAGEM

Neste algoritmo existem dois critérios de paragem. Em primeiro lugar o algoritmo pára caso obtenha uma solução com 100% de aptidão. O segundo critério de paragem é o número de iterações, ou seja quando forem efectuadas 10 000 000 iterações, apresentando os resultados obtidos até ao momento.

Este segundo critério de paragem, tem como objectivo permitir ao utilizador verificar como o algoritmo está a convergir para uma solução, permitindo-lhe alterar o número de estações desejadas e o tempo de ciclo, isto com o intuito de possibilitar ao utilizador analisar os resultados e, se verificar que não existe evolução para uma solução, fazer as devidas alterações para a obtenção da mesma, e correr novamente o algoritmo com estes novos valores.

5.2. ALGORITMO GENÉTICO DE MELHORAMENTO DE RESULTADOS

Depois de implementado o Algoritmo Genético descrito e de este ter sido aplicado a um conjunto de problemas de teste, verificou-se que o sistema apresentava bons resultados mas não os resultados ideais. Por este motivo foi desenvolvido um segundo algoritmo, que melhora as soluções obtidas com o primeiro algoritmo. Este algoritmo tem por objectivo diminuir a folga da solução e diminuir o desvio padrão das folgas das estações.

5.2.1. ESTRUTURA GERAL DO ALGORITMO

Na Figura 17 é apresentada a estrutura deste segundo algoritmo. Nesta estrutura é possível ver todas as fases do processo do algoritmo, sendo estas explicadas nas subsecções seguintes.

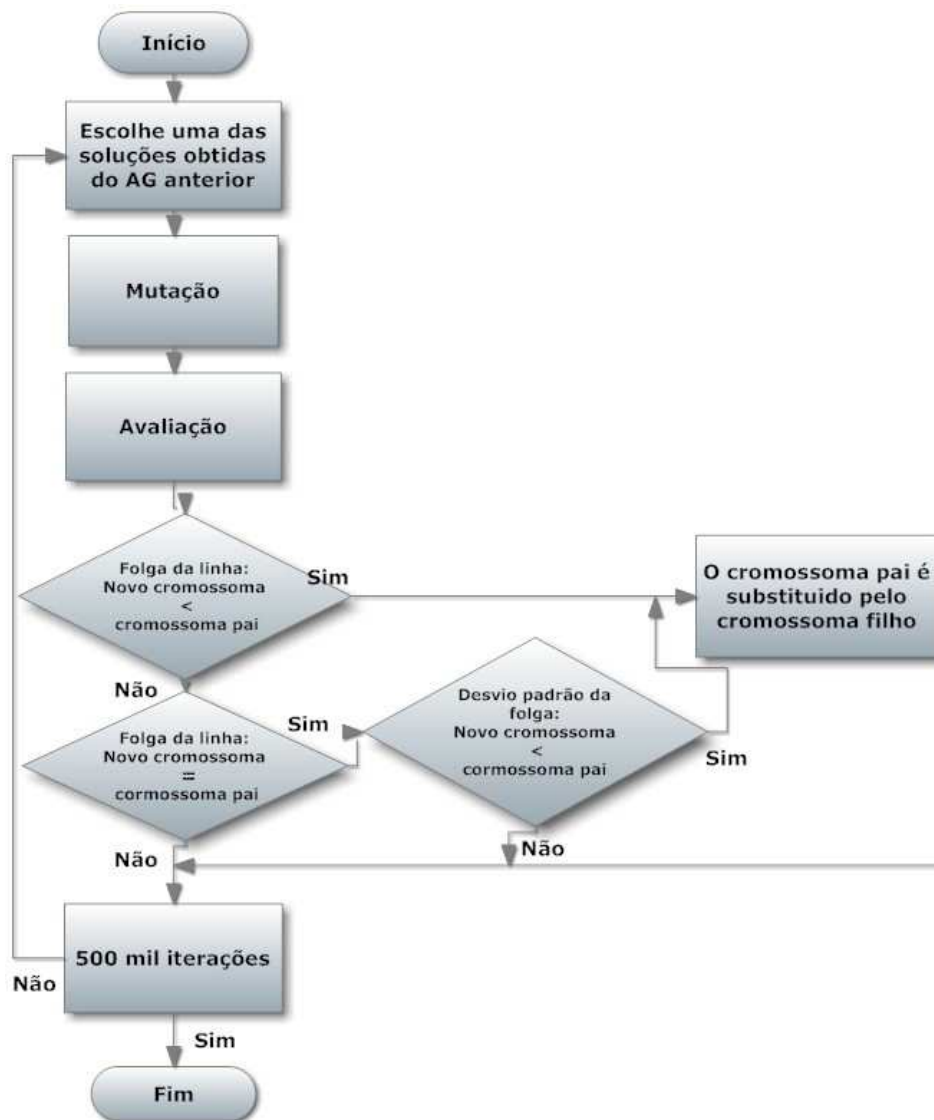


Figura 17 Estrutura do algoritmo de melhoria

5.2.2. SELECÇÃO

Como este algoritmo tem uma população de pequenas dimensões, pois a população é constituída pelas soluções obtidas no primeiro algoritmo, não é possível usar um método de selecção como o usado no primeiro algoritmo. Então o método de selecção adoptado é simplesmente a escolha aleatória de um elemento dessa população.

5.2.3. MUTAÇÃO

O processo de mutação implementado neste algoritmo é igual ao descrito no algoritmo principal, com a pequena diferença de que a mutação é apenas efectuada uma única vez em cada iteração.

5.2.4. FUNÇÃO DE AVALIAÇÃO

A função de avaliação deste segundo algoritmo é dividida em duas partes. Depois de se proceder à mutação, é aplicada a função de avaliação do algoritmo anterior para garantir que o novo cromossoma, criado pela mutação, garante que todas as precedências ainda são cumpridas e que o número de estações de trabalho ainda é cumprido. Caso isto se comprove é aplicada a nova função de avaliação, caso a solução tenha sido comprometida com a mutação efectuada é então rejeitada de imediato.

De seguida é aplicada uma segunda função de avaliação que calcula o valor da folga e do desvio padrão da folga da solução, comparando esses valores da nova solução com a solução seleccionada para mutação (cromossoma pai). Se esta nova solução apresentar um valor da folga inferior ao do cromossoma pai, então o cromossoma pai é substituído pelo novo cromossoma. Se a nova solução apresenta folga igual à do cromossoma pai então é comparado o desvio padrão de ambas as soluções e se a nova solução tiver um valor inferior ao do cromossoma pai então este é substituído pela nova solução. Caso a nova solução apresente uma folga superior à do cromossoma pai, é logo descartada pois é uma solução pior.

5.2.5. CRITÉRIO DE PARAGEM

O critério de paragem deste algoritmo é o número de iterações, ou seja o Algoritmo Genético pára quando é atingido o valor definido de 500 000 iterações.

precedência. Depois de introduzidas todas as precedências, se ainda existirem espaços vazios, estes são preenchidos de forma automática com “0”, o que corresponde a não existirem mais precedências para a operação em questão. Na Figura 19 mostra-se um exemplo onde o utilizador indica que o problema contém 5 operações e que a base de tempo é em segundos, e depois introduz os dados na tabela.

ID	Descrição	Tempo De Operação	Op. Prec. 1	Op. Prec. 2	Op. Prec. 3	Op. Prec. 4	Op. Prec. 5
1		12					
2		22	1				
3		10	1				
4		15	2				
5		30	4	3			

Figura 19 Exemplo de introdução de dados

Depois de toda a tabela preenchida, o utilizador tem a possibilidade de escolher entre dois géneros de problemas: o primeiro onde o tempo de funcionamento da linha e o número de peças pretendidas são conhecidos, e um segundo onde a linha tem um número de estações fixas e se pretende obter a produção máxima possível para esse número de estações. Na Figura 20 são visíveis as opções de selecção do problema, encontrando-se seleccionado na figura o caso em que o utilizador tem conhecimento do tempo de funcionamento e das peças a produzir e deseja calcular as estações desejadas e balancear a linha.

Figura 20 Selecção de tipo de problema

Como anteriormente descrito, esta aplicação apresenta dois passos principais, em que cada um corresponde a um Algoritmo Genético diferente. No caso do primeiro algoritmo

genético existe ainda outro campo a preencher pelo utilizador que define quantas soluções este deseja obter para depois escolher a mais adequada aos seus objectivos ou necessidades; este valor será a população do segundo algoritmo. Este campo é apresentado no canto superior direito da interface com o utilizador (como se pode ver na Figura 21), sendo apresentados por baixo deste campo os 20 cromossomas da população e respectiva aptidão e número de estações de cada cromossoma.



Figura 21 Selecção do número de soluções

Depois de todos os aspectos do problema estarem definidos, surge a primeira fase em que é calculado o tempo de ciclo e o número de estações para o problema introduzido e gerada a população inicial. De seguida é executado o primeiro AG que irá obter o número de soluções desejadas pelo utilizador. Este algoritmo é executado até obter o número de soluções desejadas ou até atingir 10 milhões de iterações. Se, após este número de ciclos, o AG não obteve as soluções pretendidas, é possível alterar os dados do problema, tais como o tempo de ciclo ou o número de soluções, e executar novamente o algoritmo.

Na Figura 22 é mostrada a janela da aplicação depois de executado o primeiro algoritmo, sendo que neste caso o algoritmo parou após as 10 milhões de iterações, tendo obtido até este momento duas soluções das 5 desejadas pelo utilizador (estas encontram-se marcadas a azul na Figura 22). Neste caso o utilizador pode colocar o algoritmo a executar novamente até obter as restantes soluções desejadas.

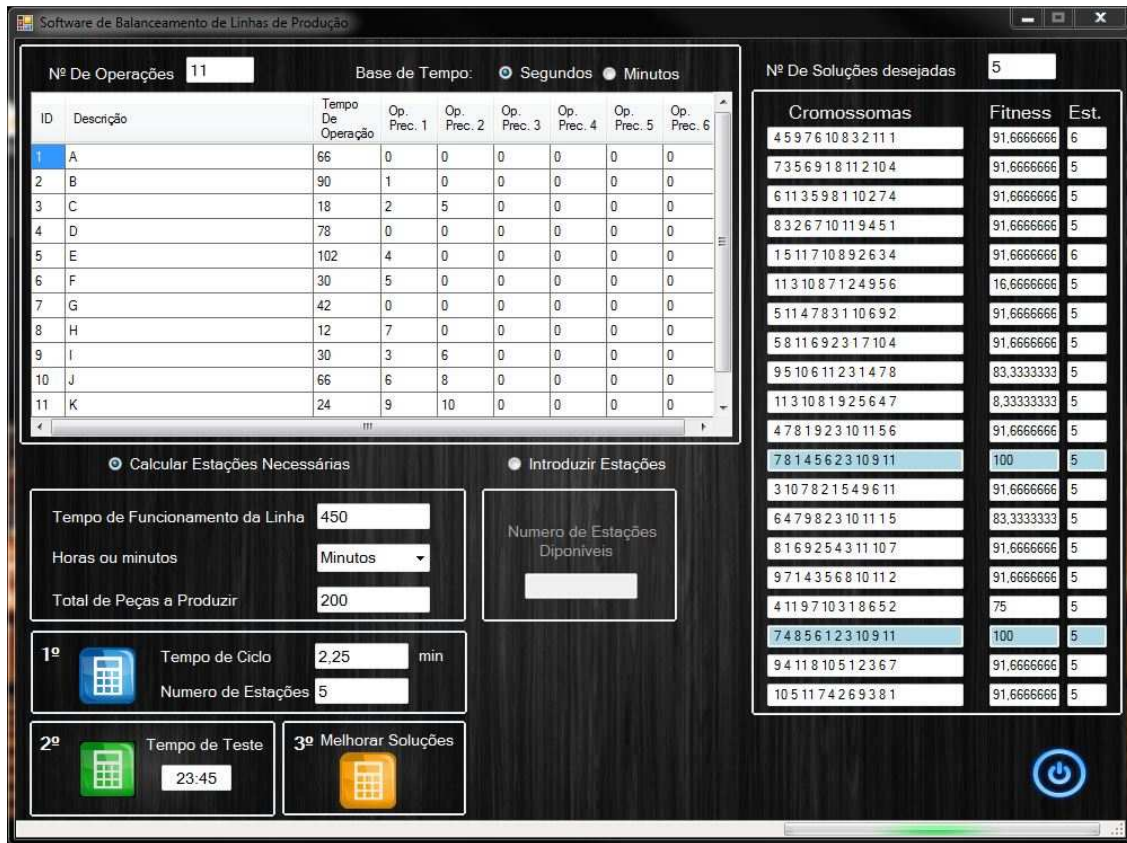


Figura 22 Interface gráfica da aplicação depois de executado o primeiro AG

No caso de o algoritmo executar as 10 milhões de iterações e não encontrar nenhuma solução, é apresentada uma mensagem ao utilizador e é-lhe permitido alterar o valor de tempo de ciclo, do número de estações, ou ambos (como este entender que seja necessário). Na Figura 23 é possível ver essa mensagem de aviso e Figura 24 um novo menu de botões que permite incrementar o tempo de ciclo e o número de estações.

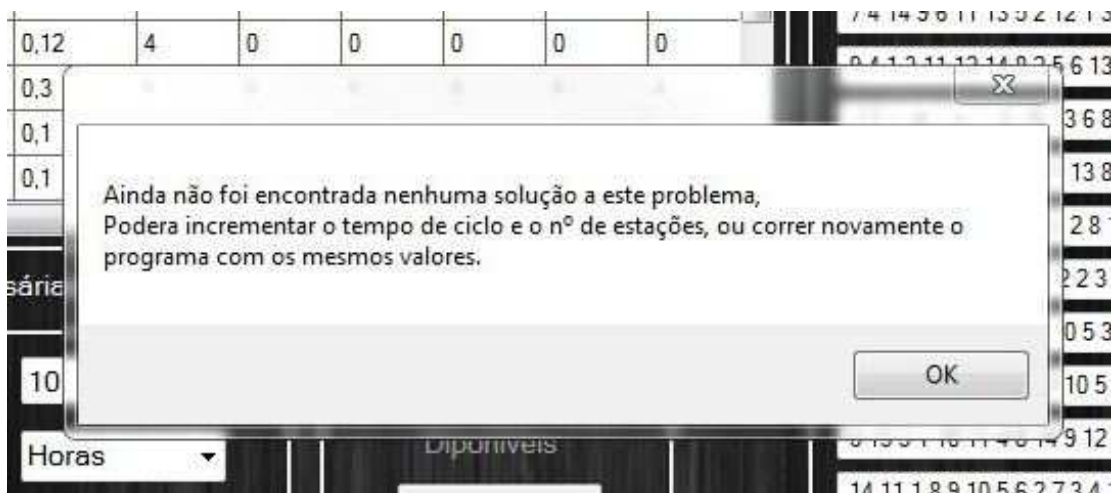


Figura 23 Janela principal da interface gráfica com mensagem de aviso



Figura 24 Menu para correção de dados do problema

Depois de finalizado o primeiro algoritmo é possível analisar cada solução e cada cromossoma da população. Para o efeito deve-se premir o rato sobre o cromossoma que se pretende analisar. Isto abre uma nova janela que permite a análise da solução segundo vários aspectos.

Na Figura 25 é apresentada a janela de verificação de resultados, onde é possível analisar estação a estação, verificar a eficiência da linha, o tempo total de paragem, o atraso do balanceamento, a folga da linha e o desvio padrão da folga. Esta janela contém ainda um gráfico que representa a ocupação temporal de cada estação, bem como a respectiva folga, e é possível ver a distribuição do tempo pelas estações verificando a ocorrência de sobrecarga em alguma estação (estrangulamento da linha).



Figura 25 Janela de análise de resultados

Após a análise dos primeiros resultados, e voltando à janela principal da aplicação, é possível correr o segundo AG que é o “responsável” por melhorar os resultados obtidos pelo primeiro algoritmo. Este segundo AG, como referido anteriormente, tem como

população inicial as soluções do algoritmo executado no passo anterior. Este algoritmo executa 500 mil iterações e pode ser executado quantas vezes o utilizador desejar. Na Figura 26 é apresentada a solução melhorada do problema em consideração, após o término da execução deste segundo AG.



Figura 26 Análise dos resultados do segundo algoritmo

6. TESTES DA APLICAÇÃO DESENVOLVIDA

Neste capítulo são apresentados os vários testes efectuados à aplicação desenvolvida, utilizando exemplos de problemas. Em cada teste são apresentados os resultados obtidos pela aplicação desenvolvida e os resultados obtidos resolvendo o problema através de heurísticas.

6.1. PRIMEIRO TESTE

Como primeiro teste é escolhido um pequeno problema de relativa simplicidade, para testar a interface da aplicação e as suas funcionalidades bem como os algoritmos implementados.

Considere-se uma empresa que produz um dado artigo, através de um conjunto de operações com as precedências indicadas na Tabela 9. Sabe-se que a empresa tem um horário de laboração de 8 horas diárias e que deseja produzir 320 artigos por dia.

É pretendido definir um balanceamento da linha de produção e obter a eficiência da mesma.

Tabela 9 Tabela de operações, tempos de execução e precedência do problema

Operações	Tempo de operação (seg.)	Precedências
A -1	70	-
B -2	80	A
C - 3	40	A
D - 4	20	A
E - 5	40	A
F - 6	30	B,C
G - 7	50	C
H - 8	50	D,E,F,G

O problema é introduzido na aplicação desenvolvida, é seleccionado na aplicação que se deseja obter uma solução e são executados os dois algoritmos. A aplicação demora 2 minutos a correr até apresentar uma solução possível para o problema. Considerando também o tempo de introdução do problema na aplicação, isto significa que o tempo para obter uma solução para este problema ronda 4 a 5 minutos.

Na Figura 27 é possível analisar a solução obtida pela aplicação informática.



Figura 27 Solução obtida para o problema 1

Depois de obtida a solução utilizando a aplicação desenvolvida, esta é então comparada com a solução obtida através da aplicação de métodos heurísticos de Helgeson-Birnie. Em termos de critérios de comparação das soluções, comparam-se as eficiências de ambas as soluções, as respectivas folgas de balanceamento e o desvio padrão das folgas. Considera-se que o mais importante dado de comparação de ambas as soluções será a folga e o desvio padrão da folga pois estes valores são os que informam da qualidade do balanceamento das linhas. Uma menor folga significa que a linha se encontra a produzir perto do máximo (ou mesmo no máximo) possível e que não existem recursos a ser desperdiçados; o menor desvio padrão significa que existe uma coerência em todas as estações e que nenhuma tem um tempo de inactividade muito superior a qualquer outra.

Na Tabela 10 são apresentados os valores obtidos para os critérios de comparação das soluções obtidas recorrendo aos dois métodos usados para resolver o problema. É possível verificar que ambas as soluções são viáveis e que ambas apresentam a mesma eficiência, mas pelos valores de folga e desvio padrão verifica-se que a solução obtida através da aplicação informática leva a uma linha melhor balanceada e é uma boa solução para o problema, sendo apenas necessário 1/3 do tempo para obter a solução. Conclui-se que, para este problema particular, a aplicação da computação evolutiva apresenta a melhor solução.

Tabela 10 Tabela de comparação de resultados para o problema 1

	Aplicação Desenvolvida	Heurísticas
Linha balanceada	A B E C G D F H	A B C E D G F H
Eficiência	84,4%	84,4%
Tempo total de paragem	70 Segundos	70 Segundos
Folga da linha	5 %	15,5%
Desvio padrão das folgas	4,90	16,7
Tempo de resolução	4 ~ 5 Minutos	10 ~ 15 Minutos

6.2. SEGUNDO TESTE

Neste segundo teste é utilizado um problema com um número superior de operações relativamente ao teste anterior. Considera-se que a montagem de um produto exige 11 operações, que se encontram listadas na Tabela 11, onde também são apresentados os respectivos tempos de operação e as precedências.

Tabela 11 Tabela de operações, tempos de execução e precedência do problema 2

Operações	Tempo de operação (min.)	Precedências
A	1,1	-
B	1,5	A
C	0,3	B,E
D	1,3	-

E	1,7	D
F	0,5	E
G	0,7	-
H	0,2	G
I	0,5	C,F
J	1,1	F,H
K	0,4	I,J

Começa-se por se introduzir os dados das operações na aplicação, esta calcula o número de estações necessárias e o tempo de ciclo, é introduzido o numero de soluções que se pretende obter e são executados os dois algoritmos (o primeiro para procurar as soluções e o segundo para melhorar as soluções). Obtém-se uma solução para este problema em 13 minutos. Considerando os 13 minutos de execução da aplicação mais o tempo de introdução dos dados, perfaz um tempo total de 15 minutos a 20 minutos para obter uma solução para este problema através da aplicação desenvolvida.

Na Figura 28 é apresentada a solução obtida pela aplicação para possível análise.

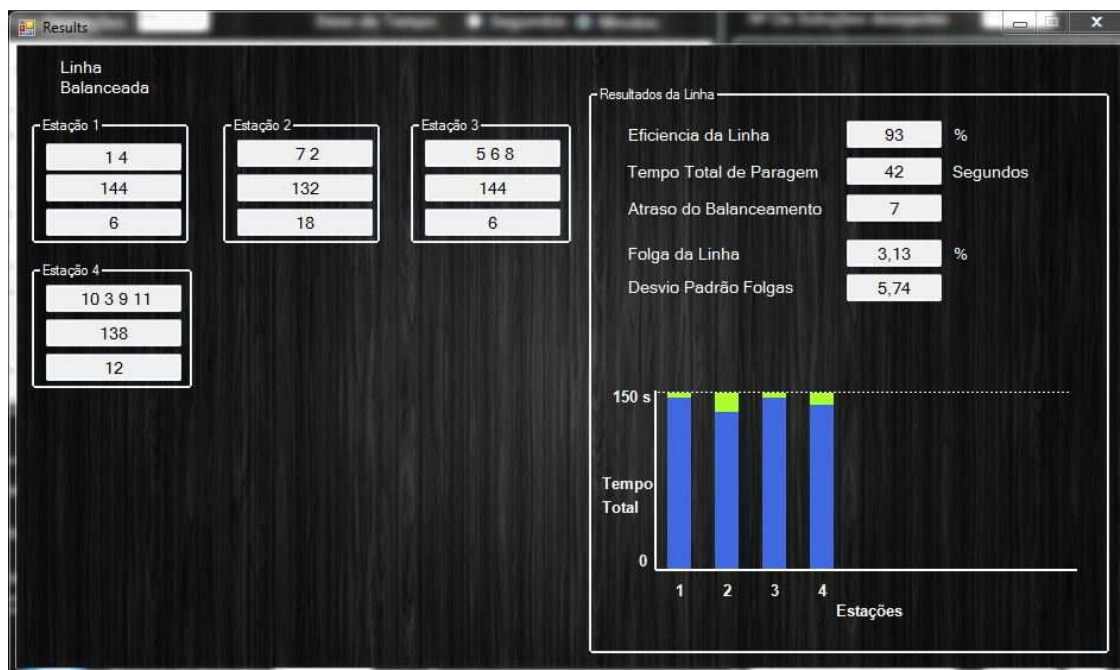


Figura 28 Solução obtida para o segundo problema de teste da aplicação

De seguida a solução obtida pela aplicação informática é comparada com a solução obtida através da aplicação das heurísticas de Helgeson-Birnie.

Tabela 12 Tabela de comparação de resultados para o problema 2

	Aplicação Desenvolvida	Heurísticas
Linha balanceada	A D G B E F H J C I K	E G D A B F C I K H J
Eficiência	93%	93%
Tempo total de paragem	42 Segundos	42 Segundos
Folga da linha	3,13 %	3,13%
Desvio padrão das folgas	5,74	5,74
Tempo de resolução	10 ~ 15 Minutos	20 ~ 25 Minutos

Através da análise da Tabela 12, é possível verificar que neste problema ambas as soluções são equivalentes, pois ambas apresentam valores iguais de eficácia, folga e desvio padrão de folga. A única diferença entre elas é o tempo em que se obtém a solução, sendo que a aplicação informática a obtém mais rapidamente que através do uso das heurísticas.

6.3. TERCEIRO TESTE

O terceiro teste é efectuado usando um problema de balanceamento onde existem 11 operações, descritas na Tabela 13. O objectivo desta linha de produção é produzir 30 peças por hora.

Tabela 13 Tabela de operações, tempos de execução e precedência do problema 3

Operações	Tempo de operação (seg.)	Precedências
A	25	-
B	30	A
C	15	A
D	30	A
E	40	C,D
F	20	D
G	10	B
H	15	G
I	35	E,F,H
J	25	I
K	25	J

Utilizando os dados desta tabela e a informação sobre a linha de montagem, este problema é introduzido na aplicação desenvolvida e são executados os algoritmos para obter uma solução. Neste caso a aplicação demorou 15 minutos a obter uma solução, a que se adiciona o tempo de introdução de dados e de configuração da aplicação.

Na Figura 29 é apresentada a solução encontrada pela aplicação informática para este problema de balanceamento de linha de produção. Como é possível verificar, esta solução contém apenas 3 estações, nas quais o tempo de ocupação é muito idêntico (levando a um desvio de padrão muito baixo) e todas têm uma folga baixa. Verifica-se então que esta é uma boa solução para o problema.



Figura 29 Solução obtida para o terceiro problema.

De seguida é apresentada uma comparação entre a solução obtida pela aplicação desenvolvida e a solução obtida através das heurísticas de Helgeson-Birnie, comparando eficiência, folga da linha e desvio padrão da linha.

Tabela 14 Tabela de comparação de resultados para o problema 3

	Aplicação Desenvolvida	Heurísticas
Linha balanceada	A D C F E B G H I J K	A B D F C E G H I J K
Eficiência	75%	75%
Tempo total de paragem	90 Segundos	90 Segundos
Folga da linha	5,26%	25%
Desvio padrão das folgas	5	36,05
Tempo de resolução	18 ~ 20 Minutos	20 ~ 25 Minutos

Através da análise dos dados apresentados na Tabela 14 é possível verificar que a solução apresentada pela aplicação desenvolvida é melhor do que a obtida pelas heurísticas. Apesar de ambas terem a mesma eficiência, o trabalho está melhor distribuído pelas estações de trabalho na solução obtida com os algoritmos genéticos.

6.4. QUARTO TESTE

Neste quarto teste, o problema seleccionado é um exemplo onde a procura faz com que a solução desejada não seja possível e seja necessário o ajuste das estações de trabalho.

É um problema em que se tem uma linha de montagem composta por 12 operações, e o objectivo é produzir 420 peças em 7 horas de serviço. Na Tabela 15 são apresentadas as operações, os tempos de operação de cada uma e as respectivas precedências.

Tabela 15 Tabela de operações, tempos de execução e precedência do problema 4

Operações	Tempo de operação (min)	Precedências
A	0,2	-
B	0,4	-
C	0,7	A
D	0,1	A,B
E	0,3	B
F	0,11	C
G	0,32	C
H	0,6	C,D
I	0,27	F,G,H
J	0,38	E
K	0,5	I,J
L	0,12	K

Depois de introduzidos os dados do problema na aplicação, e introduzido o tempo de funcionamento da linha e o número de peças desejadas, é executado o primeiro algoritmo. Este corre por 24 minutos, terminando sem encontrar soluções. Na Figura 30 é possível ver

a janela principal com os dados do problema com os resultados obtidos depois do primeiro algoritmo ter sido executado e não ter sido encontrada nenhuma solução



Figura 30 Janela principal depois do primeiro Algoritmo executado no quarto teste

Através da análise da população, é possível verificar que, devido à carga em cada estação é impossível obter soluções com apenas 4 estações, sendo apenas possível obter soluções com 5 estações. Sendo assim, o número de estações é incrementado em um e o primeiro algoritmo é executado de novo. Esta análise pode ser feita através de um cromossoma da população, tal como visível na Figura 31.



Figura 31 Análise de um cromossoma da população do quarto teste

A aplicação com o incremento do número de estações de trabalho demorou 2 minutos a obter uma solução, através da execução do primeiro algoritmo e do segundo (o algoritmo de melhoramento).

Na Figura 32 é possível ver a solução obtida, com as 5 estações de trabalho, e uma carga idêntica em todas as estações.



Figura 32 Solução obtida para o quarto teste

De seguida é apresentada a tabela de comparação entre o resultado obtido com a aplicação desenvolvida e analiticamente usando as heurísticas de Helgeson-Birnie.

Tabela 16 Tabela de comparação de resultados para o problema 4

	Aplicação Desenvolvida	Heurísticas
Linha balanceada	A D C F E B G H I J K	A B D F C E G H I J K
Estações de Trabalho	5	5
Eficiência	80%	80%
Tempo total de paragem	60 Segundos	60 Segundos
Folga da linha	13,04%	17.64%
Desvio padrão das folgas	6,50	10.86
Tempo de resolução	30 Minutos	20 ~ 25 Minutos

Na Tabela 16 é possível verificar que ambas as soluções contêm 5 estações de trabalho, e eficiências iguais, mas a solução obtida pelo *software* apresenta um valor de folga melhor e um desvio padrão inferior, sendo uma solução melhor para este problema.

7. CONCLUSÕES E PERSPECTIVAS DE DESENVOLVIMENTOS FUTUROS

No decorrer deste trabalho foram abordados e estudados múltiplos temas e aspectos relacionados com balanceamento de linhas de produção e algoritmos genéticos. Como referido anteriormente, no decorrer dos dias de hoje o balanceamento de linhas de produção é um importante factor para as fábricas de produção, por permitir obter uma eficiência de produção elevada e baixar custos de produção.

Neste trabalho foram desenvolvidos dois algoritmos genéticos para se proceder ao balanceamento de linhas de produção e uma interface gráfica para “interligar” o utilizador aos algoritmos e facilitar o uso dos mesmos.

De referir que os objectivos colocados inicialmente foram atingidos. Sendo que o projecto tinha como objectivo obter resultados de balanceamento de linhas usando AG equivalentes aos resultados obtidos usando heurísticas, concluiu-se que em diversos problemas a

aplicação desenvolvida obteve melhores resultados, obtendo melhores folgas e estações de trabalho com cargas de trabalho semelhantes.

Relativamente a perspectivas para desenvolvimentos futuros deste trabalho, existem alguns pontos que necessitam de melhorias, nomeadamente:

- Ser possível seleccionar o tamanho da população desejada;
- Melhoramentos na interface gráfica, nomeadamente apresentar o gráfico de precedências do problema;
- Eliminar a limitação no número de precedências por operação;
- Melhoramentos na janela de análise dos resultados, através da introdução de ajudas na análise, evidenciando quais os potenciais problemas na linha ou as causas dos resultados obtidos;
- Melhoramento no menu de correcção de dados dos problemas quando não é possível encontrar soluções, tal como dar a informação ao utilizador sobre qual a restrição no problema e que alterações são necessárias para corrigir o problema encontrado.
- Acrescentar operador cruzamento e verificar se existe melhoramentos nos resultados.

Referências Documentais

- [1] Phil Ament. (1997-2007, última consulta em Julho 2011). *Ransom Olds*. Disponível: <http://www.ideafinder.com/history/inventors/olds.htm>
- [2] James B. Dilworth, *Production and Operations Management*, Fourth Edition ed.: McGraw 1989.
- [3] Jay Heizer, Barry Render, *Production and Operations Management*, Second ed.: Allyn and Bacon, 1991.
- [4] Maria A. Carravilla. (Abril 1998, Layouts Balanceamento de Linhas.
- [5] Proplanner. (2002-2011, última consulta em Julho 2011). *ProBalance*. Disponível: <http://www.proplanner.com/index.cfm?nodeID=25083>
- [6] A. Balance. (2011, última consulta em Setembro 2011). *Balance*. Disponível: <http://www.avix.eu/en/our-products/avix-balance.html>
- [7] Sérgio Mayerle e Rodrigo Santos, "Algoritmo genético para o balanceamento de linhas de produção," *XXIII Encontro Nac. de Eng. de Produção* 21 a 24 de outubro 2003.
- [8] Bai Ying, Zhao Hongshun e Zhu Liao, "MIXED-MODEL ASSEMBLY LINE BALANCING USING THE HYBRID GENETIC ALGORITHM," *International Conference on Measuring Technology and Mechatronics Automation*, 2009.
- [9] Lawrence Davis, *Handbook of Genetic Algorithms*, 1992.
- [10] Goldberg, *Genetic algorithms in Search, Optimization, and Machine Learning*, 1989.