**4**

**Descrição do Programa**

**4.1. Introdução**

Neste capitulo pretende-se descrever os elementos chave do programa, a metodologia adotada no seu desenvolvimento assim como dar uma ideia clara da comunicação existente entre os vários componentes do software.

O programa é composto pelo algoritmo genético e pela interface de comunicação com o software FEM. Neste caso o software FEM utilizado é o Robot Structural Analysis da Autodesk.

A comunicação com o Robot foi feita com a API (Application Programming Interface) disponibilizada pela Autodesk e que permite controlo total do Robot de forma automática em qualquer linguagem com suporte de COM interface (C++, C#, VB, etc.)

Para desenvolver este programa utilizou-se a linguagem de programação C# pois oferecia um bom compromisso entre performance e tempo de desenvolvimento. Ao não utilizar C++ evitou-se o controlo manual de alocação de memória o que permitiu focar mais tempo no desenvolvimento do programa e não a resolver problemas de *memory leaks*. Por outro lado, ao não utilizar uma linguagem de mais alto nível como por exemplo pyhton obtemos um programa com melhor performance e onde a comunicação com o Robot fica facilitada.

Como software de teste na definição inicial do ADN de um individuo tipo foi utilizado o Rhinoceros + Grasshopper pois a sua capacidade de design paramétrico permitiu testar de forma rápida cada alteração feita no algoritmo de geração do ADN, esta metodologia sera descrita com mais detalhe mais à frente.

**4.2. Síntese de funcionamento**

Este subcapítulo pretende dar uma ideia geral dos vários componentes do programa e da comunicação entre eles, cada componente será detalhado nos subcapítulos seguintes.

Na primeira fase do programa é pedido ao utilizador para introduzir as características das secções que com que o programa deve procurar a estrutura mais eficiente. De seguida o utilizador introduz dados acerca da geometria da torre, parâmetros como largura entre montantes, altura dos cabos, e numero de cabos.

Com esta informação é gerada uma estrutura que é a representação do geométrica das informações contidas no código genético da estrutura base. Esta é uma estrutura em que cada nó está ligado aos nós imediatamente a cima e a baixo no seu próprio plano, ou seja a estrutura inicial é uma estrutura altamente redundante com todas as barras possíveis activas e é o ponto de partida do algoritmo genético.

Após criada esta primeira estrutura o utilizador pode no Robot criar os casos de carga adequados e que o programa vai aplicar a todas as estruturas analisadas durante a busca do opimo. É de seguida criada a população inicial com a dimensão desejada pelo utilizador e onde as secções das barras são escolhidas de forma aleatória (incluído a opção de descativar a barra) e as coordenadas dos nós também sofrem uma ligeira mutação nos pontos onde tal é permitido.

Em cada individuo da população inicial é feita a analise FEM do robot de onde se recebem esforços em cada barra, essa informação é introduzida como input na função de fitness.

De forma a determinar os comprimentos de encurvadura dos diferentes tipos de elementos da estrutura (Montantes, bracing, barras horizontais) a informação geométrica do modelo é processada por um algoritmo que percorre cada tipo de barra, analisa que outros elementos lhe estão conectados e daí cria uma lista de calculo com a informação de uma nova barra analítica, por que barras do model é ela constituída e o seu comprimento de encurvadura. Após finalizada esta rotina de processamento da geometria recebemos a lista de calculo a serem feitos para cada individuo de forma a obter o fitness de cada um.

Esse calculo é feito na função de verificação do EC3 presente na classe “Calc\_operations”, daí é recebido o fator de utilização (u/f) em cada barra. São elaboradas três listas: u/f baixo, aceitável, demasiado alto. Barras presentes na primeira ou ultima lista adicionam peso à estrutura como critério de penalização de barras sob/subdimensionadas.

Por fim à eventual pontuação resultante da penalização anterior é adicionado o peso real da estrutura que constitui a classificação de fitness da solução.

Após esta classificação da solução inicial a função de seleção é chama a escolher dois elementos para dar origem a um novo individuo da próxima geração. Esse individuo passa pelos operadores genéticos d crossover e mutação. Esse novo individuo substitui o pior individuo da população atual, é avaliado e o ciclo de evolução é repetido.

[IMAGEM DOS VARIOS ELEMENTOS E SETINHAS]

**4.3. Código genético base**

A descrição da estrutura base é critica para o algoritmo genético pois uma definição incorreta pode levar o algoritmo a procurar soluções desinteressantes, ou não adequadas ao problema, uma solução demasiado restritiva pode impedir o algoritmo de explorar caminhos que levariam a um melhor ótimo local.

Para não restringir o algoritmo à partida optou-se por gerar uma estrutura com todas as conexões entre nós possíveis (e aceitáveis numa solução ótima).

Para desenvolver o algoritmo que gera esta estrutura base foi utilizado o Rhinoceros em conjunto com o addon Grasshopper. Estes dois programas em conjunto aumentam a rapidez na definição da geometria pois qualquer alteração de código é imediatamente representada graficamente sem ser necessário compilar cada alteração feita. Apesar do grasshopper se comercialmente conhecido pela sua capacidade de programação visual esses componentes foram utilizados apenas para representar parâmetros de input como altura da estrutura e posição dos cabos. O algoritmo em si foi feito com o elemento “*C# scripting*” presente no programa pois desta forma a transferência do código para o programa final fica reduzida a uma operação de copiar e colar. A ligação das variáveis definidas

[imagem do grasshopper]

A descrição da estrutura base é critica para o algoritmo genético pois uma definição incorreta pode levar o algoritmo a procurar soluções desinteressantes, ou não adequadas ao problema, uma solução demasiado restritiva pode impedir o algoritmo de explorar caminhos que levariam a um melhor ótimo local.

**4.4. População inicial**

Pretende-se, com estas regras, estabelecer um conjunto de princípios que assegurem uma uniformidade adequada aos trabalhos a apresentar como Dissertações. Procurou-se que as mesmas não fossem demasiado rígidas e difíceis de entender mas, fundamentalmente, definir regras que deverão ser aplicadas às situações mais correntes e que possam, com sensatez, ser adaptadas para casos mais particulares.

Espera-se que os estudantes dediquem algum zelo à produção gráfica final dos seus documentos. Trata-se de algo que, ao contrário dos trabalhos produzidos no âmbito das restantes disciplinas cuja divulgação raramente ultrapassa o contexto da avaliação, ficará acessível de forma alargada ao meio científico, técnico e profissional, pelo que, depois do esforço na produção de conteúdos válidos, só fará sentido que estes sejam apresentados de forma profissional e graficamente atraente.

**4.5. Função “evaluate”**

**4.6. Função seleção**

**4.7. Operadores genéticos**

4.7.1. Crossover

4.7.2. Mutação gaussiana