

Maintenance Planning in Wind Farms with Allocation of Teams Using Genetic Algorithms

I. Fonseca, J. T. Farinha e F. M. Barbosa, *Sénior Member, IEEE*

Abstract— Renewable energy is essential for global sustainability. Wind power gave a great contribution to this energy paradigm. The paper describes a maintenance management system for inshore and offshore wind farms, as well as a method to schedule maintenance using genetic algorithms. Under this subject there exists a sub-problem that concerns with the determination of the cheapest path between two points. The paper compares the performance of three methods: genetic algorithms, *Dijkstra's* algorithm and ant optimization algorithm to the determination of the cheapest path between different wind parks for maintenance.

Keywords— Ant optimization, *Dijkstra's* algorithm, genetic algorithms, optimization, renewable energy, wind energy, wind farm maintenance.

I. INTRODUÇÃO

A energia eólica é um mercado em ascensão que tem vindo a ter um impacto crescente no quadro energético nacional e mundial, contribuindo para a redução de poluentes e consequente ajuda na transformação do paradigma relacionado com as alterações climáticas.

A União Europeia tem incentivado os estados membros a fazer uma forte aposta na energia eólica. Em Portugal, no final de 2008, havia uma potência instalada de 2.862 MW e de 3.535 MW no final de 2009, o que corresponde a um acréscimo de 2,4% [1]. Todo este cenário torna relevante o estudo da manutenção de aero-geradores tendo como objectivo o incremento dos seus ciclos produtivos, através da minimização do *downtime*, a optimização no aproveitamento dos regimes de ventos, conjugados com os preços de venda da energia.

O presente artigo versa sobre a primeira das vertentes precedentes, apresentando, na primeira parte, um sistema integrado de gestão de manutenção, que comporta uma metodologia para optimizar os ciclos de manutenção e, por consequência, para maximizar o potencial de produção. O sistema permite ainda a inclusão de novas metodologias de forma simples e eficiente através do módulo de manutenção

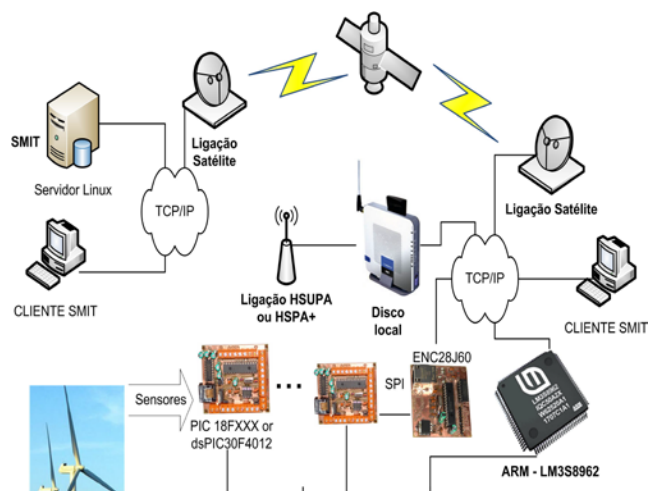
I. Fonseca, Instituto Superior de Engenharia de Coimbra, e Centro de Engenharia Mecânica da Universidade de Coimbra, Portugal, inacio@isec.pt.

J. T. Farinha, Instituto Superior de Engenharia de Coimbra, e Centro de Engenharia Mecânica da Universidade de Coimbra, Portugal. torres.farinha@dem.uc.pt.

F. M. Barbosa, Faculdade de Engenharia do Porto, e INESC Porto, Porto, Portugal, fmb@fe.up.pt.

de condição integrado no *software* de manutenção SMIT (Sistema Modular Integrado de Terologia) [2-5]. Este sistema, SMIT, inclui os módulos tradicionais de gestão da manutenção, tais como Ordens de Trabalho, Planos de Manutenção, Peças-de-Reserva, Técnicos, Ferramentas, Objectos de Manutenção, Diagnóstico de Avarias, entre outros módulos, sendo exposto no ponto seguinte enquanto elemento integrador das metodologias apresentadas neste artigo, que são as matérias nucleares do mesmo.

Os algoritmos genéticos têm sido utilizados para optimizar a localização geográfica dos aero-geradores e dos parques eólicos [6-7]. No artigo [8], Baskar apresenta uma metodologia fundamentada em algoritmos genéticos para planear a manutenção de geradores eléctricos nos Sistemas Eléctricos de Potência. O livro [9] escrito por M. Shahidepour descreve metodologias e algoritmos de escalonamento e optimização da manutenção aplicados aos novos processos de produção de energia eólica. Em [10] Andrawus apresenta os princípios para optimizar a manutenção de aero-geradores, considerando um modelo de falhas empregando simulação de Monte Carlo, em conjugação com um modelo matemático de falhas de atraso temporal baseado na distribuição de Weibull e outras. David McMillan [11] quantifica os benefícios da manutenção condicionada de aero-geradores *offshore*. O autor apresenta uma metodologia que faz a correspondência entre a informação do modelo de



manutenção condicionada, e os seus estados modelados por cadeias de Markov.

Figura 1. Sistema integrado de gestão da manutenção para sistemas eólicos

O artigo encontra-se organizado da seguinte forma: o ponto II apresenta resumidamente o sistema de gestão da manutenção SMIT. O ponto III expõe a descrição formal da metodologia utilizada para a optimização das tarefas das equipas de manutenção. O ponto IV descreve os algoritmos utilizados na determinação da sequência das visitas para as intervenções de manutenção, com especial ênfase nos algoritmos genéticos, no algoritmo da formiga e num algoritmo específico de suporte. No ponto V determina-se o melhor caminho entre dois pontos de um grafo, utilizando algoritmos genéticos, algoritmo da formiga e o algoritmo de *Dijkstra* [12]. No ponto VI é apresentado um exemplo prático e, finalmente no ponto VII, são descritas as conclusões referentes à metodologia apresentada.

II. O SISTEMA SMIT PARA A GESTÃO DA MANUTENÇÃO

A. O Sistema Global

O sistema central (Fig. 1) funciona sobre o sistema operativo *Linux* em conjunto com o motor de base de dados *PostgreSQL* e um servidor de páginas *Web Apache* com *PHP* integrado [13-14]. A aquisição de dados pode ser efectuada através de um sistema de aquisição de baixo custo e alto desempenho, bem como através de sistemas profissionais, tais como os da *National Instruments* usando *LabView* e *PLC's* com *Ethernet*.

B. A Tecnologia de Software do SMIT

Um dos principais objectivos do sistema é utilizar, sempre que possível, *software* livre. Actualmente, o SMIT é suportado pela distribuição *Slackware Linux* [13], versão 13.0 e *FreeBSD 7.0*, [33].

Ao nível da Base de Dados, utiliza *PostgreSQL*, versão 8.2.3 e 7.4.16, [14]. Todas as características, tais como a validação de dados e a sua transacção entre tabelas, bem como operações associadas à gestão da manutenção, tal como é o caso do encerramento das Ordens de Trabalho, são executadas através de funções da base de dados escritas na linguagem *PL/pgSQL*.

Os utilizadores do SMIT operam a base de dados através de uma aplicação desenvolvida em *Delphi* para ambiente *Windows*, especialmente preparada para actualização e manutenção remotas [15]. Do ponto de vista da segurança, utiliza-se o *OpenSSL* [13] em todas as ligações à base de dados, tendo esta sido encriptada ao nível do servidor, designadamente as funções *PL/pgSQL* e os ficheiros de configuração. A distribuição do *PHP* foi alterada para permitir utilizar os seus ficheiros encriptados no servidor SMIT.

Para o tratamento dos dados do sistema de aquisição foram integrados no SMIT os programas de processamento matemático *Octave*, versão 3.0.3, e o *R*, versão 2.8.0 [16]. Por defeito, o SMIT incorpora alguns algoritmos de predição através de séries temporais [17-20]. Um dos principais objectivos da previsão é responder à seguinte questão: quantos dias faltam para que uma determinada variável atinja um valor

limite? [21]

C. A Tecnologia de Hardware do SMIT

O SMIT incorpora o sistema de aquisição de dados através de uma rede IPv4. O sistema de baixo custo utiliza três placas de *hardware*: a) uma para operar como *gateway* entre a rede *Ethernet* e a rede *CAN*, b) uma para aquisição de dados e, c) uma para acondicionamento de sinal. Para cada uma das opções existem duas versões, uma para baixo e outra para alto desempenho de aquisição. Mais informações podem ser



consultadas em [2-5], [21]. Exemplos de sistemas semelhantes podem ser encontrados em [22-23].

Figura 2. Funcionamento do sistema de aquisição de baixo custo

A Fig. 2 exibe o diagrama de fluxo de informação entre os diferentes componentes do sistema, no qual circulam os dados, os sinais de acerto de relógio e os comandos de aquisição. Os sinais de acerto de relógio permitem a aquisição síncrona nos diferentes pontos onde se instala o sistema, através da utilização de protocolos apropriados à gestão deste tipo de informação [24-25]. Os algoritmos utilizados no *software* SMIT encontram-se justificados em [3-5], [21], na vertente da detecção da condição de funcionamento de aerogeradores através da instrumentação de características físicas, tais como temperatura, potência produzida, vibrações, entre outras.

Depois da caracterização sintética do sistema de informação de gestão de manutenção, e do modelo de aquisição e tratamento dos dados, passa-se à apresentação dos novos algoritmos a embeber no sistema visando a ampliação das suas capacidades na vertente da manutenção de condição.

No caso dos geradores eólicos, o SMIT inclui um sistema de aquisição de dados com o seu respectivo tratamento através de algoritmos escritos em linguagem *Octave*, com ênfase na predição através de séries temporais [21].

III. OPTIMIZAÇÃO DAS TAREFAS DAS EQUIPAS DE MANUTENÇÃO

O problema da optimização dos trajectos entre diferentes parques eólicos é relevante, aumentando de importância com o incremento do número de parques. A partir da localização dos parques e de cada aerogerador, pretende-se optimizar as distâncias a percorrer, tendo como condicionantes as seguintes variáveis: os custos, designadamente com as perdas de produção de energia; a velocidade do vento estimada para determinado período de tempo; o custo inerente à

disponibilidade dos equipamentos necessários às intervenções de manutenção; os tempos de intervenção; e os tempos de percurso. Existe ainda outra restrição, que é o número de vias alternativas ao trajecto, o qual é limitado entre os diferentes parques, designadamente quando estes estão instalados em zonas montanhosas de difícil acesso. O oposto verifica-se num parque marítimo onde, por via aérea, a deslocação se pode fazer em linha recta; já no caso da via ser marítima, depende da navegabilidade dos percursos e das respectivas rotas.

Considerando o sistema GPS e o modelo geodésico terrestre, a partir de dois pontos GPS é possível estimar distâncias e, por consequência, mesmo em localizações geográficas não catalogadas, podem-se determinar pontos de intercepção nas estradas construídas para aceder aos parques eólicos. Assim, cada um dos pontos pode ser catalogado em dois grupos distintos:

$$G = \{(Lat_1, Long_1), \dots, (Lat_n, Long_n)\} \quad (1)$$

com $(Lat_i, Long_i) \in \{\text{posição dos aero-geradores}\}$

e o conjunto

$$C = \{(Lat_1, Long_1), \dots, (Lat_n, Long_n)\} \quad (2)$$

com $(Lat_i, Long_i) \in \{\text{ponto entre dois aero-geradores}\}$

onde o conjunto C (caminhos) agrega as restrições impostas aos elementos do conjunto G (geradores) para viajar entre dois quaisquer elementos; a Fig. 3 representa graficamente o problema.

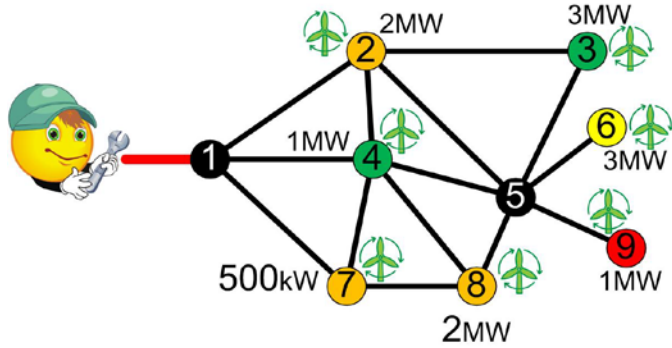


Figura 3. Aero-geradores: a vermelho, equipamento a intervencionar; a amarelo claro e escuro, diferentes prioridades para futuras intervenções

Supondo a utilização de duas equipas de manutenção, o objectivo é determinar qual deverá ser a sequência das intervenções de manutenção nos aero-geradores.

Na resolução do problema podem ser necessárias várias visitas ao mesmo nó, caso o número de dias de reparação seja superior a um. O problema ilustrado na Fig. 3 pode ser decomposto em dois problemas básicos que consistem em:

- escolher o melhor caminho entre dois nós;
- escolher qual a sequência de visitas aos nós a intervencionar (vermelho), ou com previsão de intervenção (amarelo), tendo em consideração a previsão do vento, do preço de venda da energia, entre outros.

A. Formulação da Função de Custo

A função de custo é expressa em Unidades Monetárias (UM) e representa os custos directos e indirectos, o de mão-de-obra e da sua logística - nas deslocações e pernoites - e o custo associado à perda de produção, respectivamente. O objectivo é maximizar o balanço entre débitos e créditos. Assim, o custo financeiro associado a um nó será dado por:

$$\text{custoNó} = \text{perdaPorInt} + \text{perdaManut} + \text{custoPern} \quad (3)$$

O custo das deslocações está associado a um trajecto e, portanto, a um nó origem e a um nó destino. Assim, genericamente, o custo associado a um trajecto entre o nó i e o nó j é dado por:

$$\text{custoTrajecto}_{i,j} = \text{custoViagem}_{i,j} + \text{custoTecnico}_{i,j} \quad (4)$$

Parâmetros associados ao custo do nó

Para calcular a Equação (3), o potencial financeiro diário de um determinado nó (por dia), isto é, a contribuição, no caso de se encontrar inoperacional e não produzir, é dado por:

$$\text{perdaPorInt} = P_{\text{gerador}} \times n_{\text{HprodPorDia}} \times \$_{\text{porMW}} \quad (5)$$

Seguidamente pode-se calcular a perda financeira no caso de determinado nó estar em manutenção:

$$\text{perdaManut} = \begin{cases} \text{perdaPorInt} \times \text{TaxaManut, manutenção} & (6) \\ 0, \text{ noutros casos} \end{cases}$$

O nó só se encontra em intervenção se tiver sido afectada uma equipa de técnicos para realizar os trabalhos e esta só é alocada caso $n_{\text{DiasPraIntervencao}} \leq \text{limiar}$ ou $n_{\text{DiasManutencao}} > 0$. O limiar é um valor para o qual se considera credível a predição do número de dias até que uma variável de condição atinja um valor limite.

Associado a um nó existe ainda o custo financeiro de pernoitar nesse ponto, que é dado por:

$$\text{custoPern} = \begin{cases} n_{\text{ósPernoitar}} \times \$_{\text{porNoite, equipa pernoita}} & (7) \\ 0, \text{ noutros casos} \end{cases}$$

Para estimar estes factores tem que se entrar em consideração com os seguintes parâmetros (que deverão ser fornecidos à entrada do algoritmo):

- P_{gerador} - Potência máxima disponível num aerogerador;
- $n_{\text{HprodPorDia}}$ - Número de horas de produção por dia;
- $_{\text{porMW}}$ - Preço do MWh produzido;
- $n_{\text{DiasPraIntervencao}}$ - Número de dias estimados para que uma variável de condição atinja um valor limite (predição com séries temporais - assume-se como conhecido o valor);
- $n_{\text{DiasIntervencao}}$ - Número de dias de intervenção para a avaria estimada;
- TaxaManut - Factor de perda quando um aerogerador está sujeito a uma intervenção de manutenção;
- $n_{\text{Técnicos}}$ - Número de equipas disponíveis em

simultâneo para realizar os trabalhos. Não se considera possível que duas ou mais equipas possam trabalhar em simultâneo num mesmo aero-gerador. Cada equipa tem transporte autónomo;

- *nósPernoitar* - Nós onde se pernoita (valor = 1);
- *\$porNoite* - Preço por noite, por equipa, num determinado nó.

Parâmetros associados ao custo das deslocações

Para calcular a Equação (4), define-se o custo de viagem e o custo da mão-de-obra dos técnicos da seguinte forma:

$$\begin{aligned} \text{custoViagem}_{i,j} &= \$\text{porKm}_{i,j} \times n\text{KmViagem}_{i,j} \\ \text{custoTecnico}_{i,j} &= n\text{DiasViagem}_{i,j} \times \$\text{porDiaViagem}_{i,j} \end{aligned} \quad (8)$$

Nas deslocações considera-se que cada equipa tem um meio de transporte autónomo. Para estimar os custos das deslocações entra-se em consideração com os seguintes parâmetros:

- *\$porKm* - Preço por quilómetro de deslocação num determinado troço;
- *nKmViagem* - Número de quilómetros num troço;
- *nDiasViagem* - Número de dias de viagem entre dois nós adjacentes (em Portugal Continental, valor = 1);
- *\$porDiaViagem* - Custo pago às equipas técnicas pela viagem entre dois nós adjacentes de um troço.

IV. DETERMINAÇÃO DA SEQUÊNCIA DAS VISITAS PARA AS INTERVENÇÕES DE MANUTENÇÃO

Na secção V é apresentada a resolução do problema de encontrar o melhor caminho entre dois nós. Para resolver a escolha da sequência de visitas aos aero-geradores são apresentadas duas soluções, uma baseada em algoritmos genéticos [26-28] e outra num algoritmo específico desenvolvido para o efeito. Considerando que em Portugal os aero-geradores apenas se encontram instalados em terra, fez-se a simplificação de considerar que as viagens entre dois nós é muito inferior a um dia. Este factor é importante para a representação ilustrada na Tabela I, com o objectivo de codificar uma solução (indivíduo no algoritmo genético).

TABELA I
CODIFICAÇÃO DE UMA SOLUÇÃO

DIA 1		DIA 2	
Técnico1	...	Técnico N	...
Nó trab. Nó dormir	...	Nó trab. Nó dormir	...

O comprimento n da solução é dado por:

$$\begin{aligned} n &= nE \times n\text{Tecnicos} \times 2 + n\text{Tecnicos} \\ nE &= \max\left(\sum n\text{DiasIntervencao}, M\right) \end{aligned} \quad (9)$$

$$M = \max(\text{sizey}(n\text{HprodPorDia}), \text{sizey}(\$ \text{porMW}))$$

onde se considera que cada equipa de manutenção inicia o seu "caminho" num nó de entrada pré-estabelecido (na Fig. 3, nó 1) que não está incluído na representação genética indicada na Tabela I por ser idêntico em todas as soluções. Cada solução é denominada no texto seguinte como "indivíduo". Em síntese,

o algoritmo segue os seguintes passos: aleatoriamente gera uma população de n soluções, solução para o problema em causa, avalia o seu desempenho, escolhendo os melhores indivíduos para produzir a geração seguinte através de vários operadores genéticos, tais como a descendência, a mutação, a roleta aleatória, entre outros.

A. Operadores genéticos

Para toda a população calculam-se grupos aleatórios de oito elementos e para o melhor indivíduo desses oito elementos aleatórios geram-se cinco novos indivíduos (Fig. 4):

- Mantém-se o próprio indivíduo;
- Trocam-se de lugar dois elementos aleatoriamente - considera-se elemento o conjunto de todas as visitas no mesmo dia (a ordem das visitas num dia de todos os técnicos não sofre nenhuma alteração);
- Inverte-se a ordem a um conjunto aleatório de n dias;
- Move-se um bloco de n dias uma casa à frente;
- Aleatoriamente, gera-se um novo indivíduo ou efectua-se o cruzamento entre dois elementos aleatórios, corrigindo, se necessário, para obter uma solução do problema.

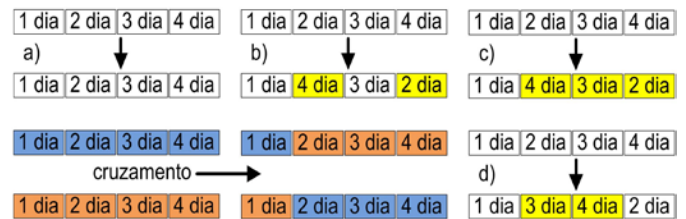


Figura 4. Operadores genéticos ao nível do dia

Ao nível da manipulação diária são gerados três indivíduos com base nos operadores a), b) e c) da Fig. 4, a partir do melhor indivíduo do grupo de oito, de acordo com o seguinte:

- Para um intervalo de dias aleatório, trocam-se de lugar dois técnicos escolhidos aleatoriamente;
- Inverte-se a ordem a um conjunto aleatório de técnicos em x dias aleatórios;
- Move-se um bloco de n técnicos uma casa à frente para x dias aleatórios.

Estas três últimas operações incluem a aleatoriedade das permutações entre as várias equipas ao nível dos trabalhos diários. Indubitavelmente, verifica-se ainda o facto dos operadores genéticos, ao cruzarem indivíduos, poderem gerar novas populações que não são solução do problema, o que é um aspecto importante que condiciona a sua utilização ou, ainda, obriga à "reparação" dos indivíduos da nova geração de modo a serem solução do problema ou, ainda, em alternativa, a descartá-los.

B. Algoritmo específico de suporte

O problema em discussão pode ser resolvido com base num algoritmo específico escolhendo os nós com os custos mais elevados após subtrair os dias previstos para eventuais avarias e os respectivos dias de intervenção, bem como os dias de manutenção planeada, dando a antevisão do número de dias

até aos custos crescerem:

$$DiasDeFolga = nDiasPraIntervencao - nDiasIntervencao$$

Para os valores $DiasDeFolga \leq 0$ haverá perdas financeiras, em caso de haver possibilidade de produção e, portanto, deve-se dar prioridade à alocação de uma equipa a estes nós. A metodologia para calcular esta solução pseudo-óptima está apresentada na Fig. 5 (Algoritmo I) e é uma das soluções para o algoritmo genético - fornece um técnico para a população, em princípio "mais apto" do que um técnico gerado aleatoriamente. Deixa-se a ressalva de que o Algoritmo I não contempla todas as possibilidades de optimização - serve para gerar uma solução mais eficaz, dando assim um ponto de partida ao algoritmo genético, e como ponto de comparação para aferir da capacidade de optimização dos operadores utilizados no algoritmo genético.

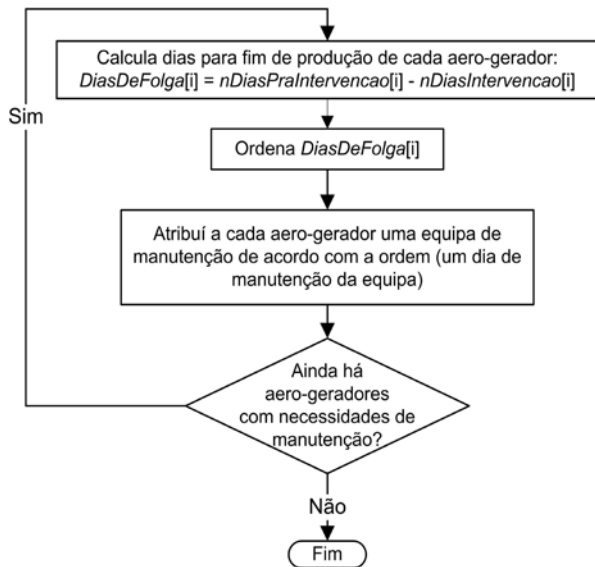


Figura 5. Algoritmo I, específico de suporte à resolução do problema

V. ESCOLHA DO MELHOR CAMINHO ENTRE DOIS NÓS

A solução de encontrar o melhor caminho entre dois pontos é oriunda da teoria de grafos onde se encontram alguns algoritmos propostos. Uma das soluções, quando os custos associados aos trajectos são positivos, é dada pelo algoritmo de *Dijkstra* [12]. No caso dos custos associados aos trajectos apresentarem valores negativos este algoritmo não funciona, sendo uma das alternativas, o algoritmo de *Bellman Ford* [29].

A. Algoritmo da Formiga

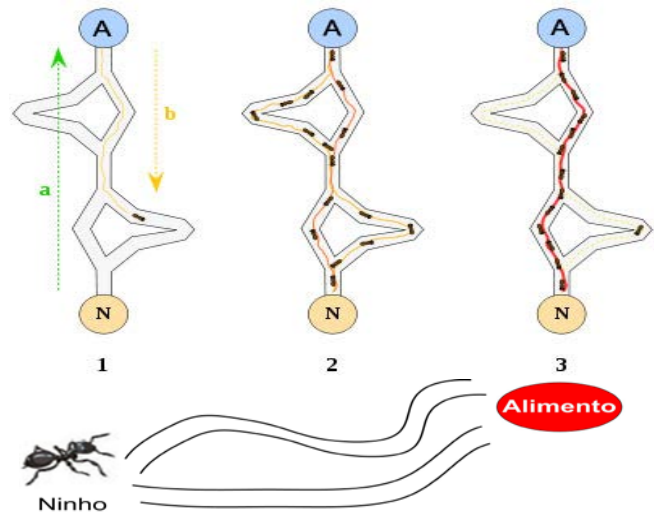
Para além da teoria dos grafos existem métodos alternativos tais como o algoritmo da formiga [30-32].

A Fig. 6 apresenta os passos básicos observados na natureza com esta espécie animal e mostra que, após algum tempo a percorrer um espaço entre dois pontos, as formigas conseguem encontrar o caminho mais curto. Assim, num primeiro passo, uma formiga escolhida aleatoriamente procura alimento, encontrando um trajecto entre o ninho e a comida (situação *a*). As formigas quando encontram o alimento voltam ao ninho pelo caminho mais curto largando produtos químicos para marcar o trajecto (situação *b*). Caso existam

dois caminhos entre o ninho e a comida (passo 2), atendendo a que o produto químico tem um tempo de vida curto e, por consequência, no caminho mais longo o seu rasto terá tendência a desaparecer. A quantidade de formigas que faz o trajecto também influencia a longevidade do efeito do produto químico deixado naquele percurso sempre que voltam ao ninho (passo 3). Numa colónia de formigas há, pelo menos, dois tipos de tarefas, que são a de recolher comida e a de procurar comida. Este sistema baseia-se no *feedback* positivo (o depósito do produto químico atrai outras formigas que reforçarão o trilho) e negativo (dissipação do trilho por evaporação, que impede o sistema de ficar caótico).

Existe uma variedade de problemas onde este algoritmo pode ser aplicado, designadamente na análise de imagens através da correspondência de contornos de silhuetas de objectos, pessoas, etc. A sua aplicação estende-se à análise de imagens médicas [31].

Figura 6. Funcionamento do algoritmo de pesquisa da formiga - adaptado de



[30]

No sector da optimização da topologia da rede eléctrica, vários autores têm utilizado este género de algoritmos, procurando configurações minimizadoras da perda de energia nos vários troços e de melhoria da qualidade da energia fornecida [32].

A vantagem da optimização pelo algoritmo das formigas relativamente aos outros métodos aludidos, advém do facto deste lidar com maior facilidade com as alterações dinâmicas do grafo representativo dos caminhos [30], como acontece, por exemplo, na modelização de problemas de tráfego, de que o urbano é um exemplo.

Do ponto de vista do formalismo matemático, o algoritmo estabelece uma probabilidade de uma formiga viajar do nó i para o nó j dada por [32]:

$$p_{i,j} = \frac{[\tau_{i,j}(t)]^\alpha \times [\eta_{i,j}(t)]^\beta}{\sum [\tau_{i,j}(t)]^\alpha \times [\eta_{i,j}(t)]^\beta} \quad (10)$$

onde $\tau_{i,j}(t)$ é a quantidade de produto químico existente no trajecto. $\eta_{i,j}(t)$ representa a atractividade do troço (tipicamente o inverso da distância), onde α e β são parâmetros de controlo da influência de cada um dos termos.

Actualização do produto químico nos troços em cada iteração

Representando $\tau_{i,j}(t) = (1 - \rho) \times \tau_{i,j}(t) + \Delta\tau_{i,j}(t)$ a quantidade de produto químico existente no trajecto e ρ a sua taxa de evaporação, o termo $\Delta\tau_{i,j}(t)$ dá a quantidade do novo produto químico depositado, cuja expressão é dada por:

$$\Delta\tau_{i,j}(t) = \begin{cases} \frac{1}{L}, & \text{se a formiga circula} \\ 0, & \text{noutros casos} \end{cases} \quad (11)$$

onde, neste caso, L representa o custo do trajecto. Este algoritmo é directamente aplicável ao problema da optimização dos percursos das intervenções de manutenção em parques eólicos.

TABELA II

DADOS INICIAIS DO PROBLEMA DE OPTIMIZAÇÃO

Dados	Nó 1	Nó 2	Nó 3	Nó 4	Nó 5	Nó 6	Nó 7	Nó 8	Nó 9
Potência (MW)	0	2	3	1	0	3	0,5	2	1
nDiasPraIntervencao	Inf	8	Inf	Inf	Inf	15	6	4	0
nDiasIntervencao	0	3	0	0	0	4	3	4	6
TaxaManut	0	0	0	0	0	0	0,4	0	0
nósPernoitar	1	0	0	0	1	0	0	0	0
SporNoite	60	0	0	0	50	0	0	0	0
\$porMW (h)									
1º Dia	0	76	76	94	0	92	99	79	89
2º Dia	0	76	76	94	0	92	99	79	89
Outros Dias	0	76	76	76	0	85	90	76	87
Nº de Horas de Produção (nHprodPorDia)									
1º Dia	12	12	12	12	12	12	12	12	12
2º Dia e outros dias	12	12	12	12	12	12	12	12	12

B. Algoritmos genéticos

Para a resolução do problema do caminho mais curto, a codificação dos indivíduos no algoritmo genético é relativamente simples, bastando um vector com n elementos onde n é o número de nós. Há uma restrição, que é a do 1º elemento ser sempre fixo, sendo este o nó de origem; os outros são obtidos aleatoriamente através das permutações dos restantes nós. Visitando a sequência, o indivíduo genético “termina” na posição onde se encontrar o nó destino. Este aspecto significa que o tamanho de cada indivíduo gerado pode variar. Os operadores genéticos aplicados estão expostos na Fig. 4, tendo sido adaptados à representação indicada.

VI. EXEMPLO PRÁTICO: DADOS DO PROBLEMA

O exemplo apresentado nesta secção reporta-se à Fig. 3 que representa 9 nós. Para executar o algoritmo, os dados fornecidos são os indicados nas tabelas II, III, IV, V e VI.

TABELA III
DADOS: DISTÂNCIA EM KM ENTRE CADA NÓ (LIGAÇÃO DIRECTA)

Km	Nó 1	Nós ...							Nó 9
Nó 1	0	20	Inf	40	Inf	Inf	60	Inf	Inf
Nó 2	20	0	60	30	40	Inf	Inf	Inf	Inf
Nó 3	Inf	60	0	Inf	20	Inf	Inf	Inf	Inf
Nó 4	40	30	Inf	0	40	Inf	30	80	Inf
Nó 5	Inf	40	20	40	0	30	Inf	60	20
Nó 6	Inf	Inf	Inf	Inf	30	0	Inf	Inf	Inf
Nó 7	60	Inf	Inf	30	Inf	Inf	0	50	Inf
Nó 8	Inf	Inf	Inf	80	60	Inf	50	0	Inf
Nó 9	Inf	Inf	Inf	Inf	20	Inf	Inf	Inf	0

TABELA IV

DADOS: MONTANTE PAGO POR KM ENTRE CADA NÓ (LIGAÇÃO DIRECTA)

\$/Km	Nó 1	Nós ...							Nó 9
Nó 1	0	1	Inf	1	Inf	Inf	1	Inf	Inf
Nó 2	1	0	1	1	1	Inf	Inf	Inf	Inf
Nó 3	Inf	1	0	Inf	1	Inf	Inf	Inf	Inf
Nó 4	1	1	Inf	0	1	Inf	1	1	Inf
Nó 5	Inf	1	1	1	0	1	Inf	1	1
Nó 6	Inf	Inf	Inf	Inf	1	0	Inf	Inf	Inf
Nó 7	1	Inf	Inf	1	Inf	Inf	0	1	Inf
Nó 8	Inf	Inf	Inf	1	1	Inf	1	0	Inf
Nó 9	Inf	Inf	Inf	Inf	1	Inf	Inf	Inf	0

TABELA V

DADOS: Nº DE DIAS DE VIAGEM (LIGAÇÃO DIRECTA)

Dias	Nó 1	Nós ...							Nó 9
Nó 1	0	0,04	0	0,04	0	0	0,04	0	0
Nó 2	0,04	0	0,04	0,04	0,04	0	0	0	0
Nó 3	0	0,04	0	0	0,04	0	0	0	0
Nó 4	0,04	0,04	0	0	0,04	0	0,04	0,04	0
Nó 5	0	0,04	0,04	0,04	0	0,04	0	0,04	0,04
Nó 6	0	0	0	0	0,04	0	0	0	0
Nó 7	0,04	0	0	0,04	0	0	0	0,04	0
Nó 8	0	0	0	0,04	0,04	0	0,04	0	0
Nó 9	0	0	0	0	0,04	0	0	0	0

TABELA VI

DADOS: MONTANTE PAGO POR CADA DIA DE VIAGEM (LIGAÇÃO DIRECTA)

\$	Nó 1	Nós ...							Nó 9
Nó 1	0	2,5	0	2,5	0	0	2,5	0	0
Nó 2	2,5	0	2,5	2,5	2,5	0	0	0	0
Nó 3	0	2,5	0	0	2,5	0	0	0	0
Nó 4	2,5	2,5	0	0	2,5	0	2,5	2,5	0
Nó 5	0	2,5	2,5	2,5	2,5	0	2,5	2,5	2,5
Nó 6	0	0	0	0	2,5	0	0	0	0
Nó 7	2,5	0	0	2,5	0	0	0	2,5	0
Nó 8	0	0	0	2,5	2,5	0	2,5	0	0
Nó 9	0	0	0	0	2,5	0	0	0	0

Considera-se ainda a existência de duas equipas de técnicos.

A Tabela VII apresenta as soluções encontradas pelo algoritmo genético, pelo algoritmo Dijkstra e pelo algoritmo da formiga, na procura da melhor solução para ir de um qualquer nó da rede a outro qualquer nó da rede, com indicação do custo, do caminho a seguir e do tempo de execução de cada algoritmo.

Relativamente ao algoritmo genético, no cálculo dos custos de trajecto entre dois nós, considerou-se que a melhor solução poderia não ser o caminho directo. Assim, “obrigou-se” o algoritmo genético a procurar outra solução, mesmo existindo um caminho directo, razão pela qual, na Tabela VII, constam, por exemplo, caminhos entre os nós 2-1, 5-2, etc. Face aos dados iniciais do problema específico apresentado na Fig. 3, concluiu-se de que o melhor caminho entre esses nós era o directo mas, caso não fosse, o algoritmo determinaria o caminho alternativo mais eficiente e eliminaria o caminho

TABELA VII
SOLUÇÕES: MELHOR CAMINHO ENTRE UM NÓ ORIGEM E UM NÓ DESTINO

Custo	Genético (segundos)	Dijkstra (segundos)	Formiga (segundos)	Origem	Destino	1º	2º	3º	4º
80.3	0.1101584	0.0000001	0.7310512	1	9	1	2	5	9
80.3	0.1101584	0.0000001	0.4506480	9	1	9	5	2	1
20.1	0.1001440	0.0000001	0.6309072	5	9	5	9	-	-
20.1	0.1101584	0.0000001	0.1301872	9	5	9	5	-	-
90.3	0.1402016	0.0000001	0.6609504	1	6	1	2	5	6
90.3	0.1101584	0.0000001	0.4506480	6	1	6	5	2	1
30.1	0.1101584	0.0000001	0.5507920	5	6	5	6	-	-
30.1	0.1101584	0.0000001	0.1301872	6	5	6	5	-	-
20.1	0.1101584	0.0000001	0.2002880	1	2	1	2	-	-
20.1	0.1101584	0.0000001	0.2503600	2	1	2	1	-	-
40.1	0.1101584	0.0000001	0.3004320	5	2	5	2	-	-
40.1	0.1101584	0.0000001	0.3605184	2	5	2	5	-	-
60.1	0.1101584	0.0000001	0.4005760	1	7	1	7	-	-
60.1	0.1101584	0.0000001	0.3905616	7	1	7	1	-	-
70.2	0.1101584	0.0000001	0.4806912	5	7	5	4	7	-
70.2	0.1001440	0.0000001	0.3505040	7	5	7	4	5	-
110.2	0.1101584	0.0000001	0.4806912	1	8	1	7	8	-
110.2	0.1101584	0.0000001	0.5207488	8	1	8	7	1	-
60.1	0.1101584	0.0000001	0.5708208	5	8	5	8	-	-
60.1	0.1101584	0.0000001	0.3304752	8	5	8	5	-	-
60.2	0.1101584	0.0000001	0.3605184	5	1	5	2	1	-

TABELA VIII
SOLUÇÕES: DO PROBLEMA GLOBAL

Sol.	Custo	Técnicos	1º Dia	2º Dia	3º Dia	4º Dia	5º Dia	6º Dia	7º Dia	8º Dia	9º Dia	10º Dia	11º Dia	12º Dia
1	10,443.10	Equipa 1	8 5	8 5	8 5	8 5	9 5	2 1	2 1	2 1	6 5	6 5	6 5	6 5
		Equipa 2	9 5	9 5	9 5	9 5	7 1	9 5	7 1	7 1	1 1	1 1	1 1	1 1
2	10,362.90	Equipa 1	8 5	8 5	8 5	8 5	7 1	2 1	2 1	7 1	6 5	6 5	6 5	6 5
		Equipa 2	9 5	9 5	9 5	9 5	9 5	9 5	7 1	2 1	1 1	1 1	1 1	1 1
3	10,362.90	Equipa 1	8 5	8 5	8 5	8 5	7 1	2 1	2 1	2 1	6 5	6 5	6 5	6 5
		Equipa 2	9 5	9 5	9 5	9 5	9 5	9 5	7 1	7 1	1 1	1 1	1 1	1 1
4	10,362.90	Equipa 1	8 5	8 5	8 5	9 5	7 1	2 1	2 1	7 1	6 5	6 5	6 5	6 5
		Equipa 2	9 5	9 5	9 5	8 5	9 5	9 5	7 1	2 1	1 1	1 1	1 1	1 1
5	10,362.90	Equipa 1	8 5	9 5	9 5	8 5	7 1	2 1	2 1	7 1	1 1	1 1	1 1	1 1
		Equipa 2	9 5	8 5	8 5	9 5	9 5	9 5	7 1	2 1	6 5	6 5	6 5	6 5
6	10,696.80	Equipa 1	9 5	8 5	9 5	9 5	2 1	2 1	7 1	2 1	6 5	6 5	6 5	1 1
		Equipa 2	8 5	9 5	8 5	8 5	9 5	9 5	6 5	7 1	7 1	1 1	1 1	1 1
7	10,696.80	Equipa 1	8 5	8 5	8 5	8 5	2 1	2 1	2 1	6 5	6 5	6 5	6 5	1 1
		Equipa 2	9 5	9 5	9 5	9 5	9 5	9 5	7 1	7 1	1 1	1 1	1 1	1 1
8	10,706.80	Equipa 1	8 5	8 5	9 5	9 5	6 5	2 1	2 1	7 1	7 1	1 1	1 1	1 1
		Equipa 2	9 5	9 5	8 5	8 5	9 5	9 5	7 1	2 1	6 5	6 5	6 1	1 1
9	11,396.80	Equipa 1	2 1	8 5	8 5	9 5	9 5	9 5	9 5	7 1	6 5	6 5	6 5	6 5
		Equipa 2	8 5	9 5	9 5	8 5	7 1	2 1	7 1	2 1	1 1	1 1	1 1	1 1
10	11,722.50	Equipa 1	8 5	9 5	9 5	8 5	9 5	9 5	9 5	9 5	6 5	6 5	6 5	6 5
		Equipa 2	2 1	8 5	8 5	7 1	7 1	7 1	2 1	2 1	1 1	1 1	1 1	1 1

TABELA IX
SOLUÇÕES: RESTANTE INFORMAÇÃO RELATIVA À TABELA VIII

Sol.	Custo	Usa a 1ª Solução	Nº Soluções	Nº Iterações	Nº de Indivíduos	1º N6	Último N6
1	10,443.10	1	1	1	1	1 1	1 1
2	10,362.90	1	2	350	100	1 1	1 1
3	10,362.90	1	2	500	150	1 1	1 1
4	10,362.90	1	3	200	100	1 1	1 1
5	10,362.90	0	10	12500	2056	1 1	1 1
6	10,696.80	0	17	400	100	1 1	1 1
7	10,696.80	0	15	5000	150	1 1	1 1
8	10,706.80	0	11	350	100	1 1	1 1
9	11,396.80	0	16	400	100	1 1	1 1
10	11,722.50	0	24	400	100	1 1	1 1

directo entre os nós. O algoritmo genético usa 25 indivíduos e 150 iterações. O algoritmo da formiga utiliza 9 formigas e 100 iterações.

As Tabelas VIII e IX apresentam os resultados para várias simulações em função dos dados iniciais do problema, expostos nas Tabelas II, III, IV, V e VI, relativamente à determinação da sequência das visitas para as intervenções.

A 1ª solução foi obtida a partir do Algoritmo I. De acordo com o valor do custo da tabela ainda não se encontra no ponto óptimo. A coluna "Usa a 1ª Solução" da Tabela IX com o valor um, indica que esta solução está incluída na primeira geração de indivíduos do algoritmo genético. Assim, de acordo com esta coluna, as soluções nºs 2, 3 e 4, foram obtidas com a inclusão do indivíduo descrito na solução nº 1. Significa que o algoritmo genético usou esta solução e tentou evolui-la para encontrar melhores soluções. A coluna "Nº de Soluções Encontradas" indica exactamente quantas soluções óptimas foram encontradas até à obtenção da que se encontra apresentada (inclusive). As restantes soluções (5ª,..., 10ª) foram obtidas apenas com o algoritmo genético, sem inclusão de nenhuma solução específica inicial.

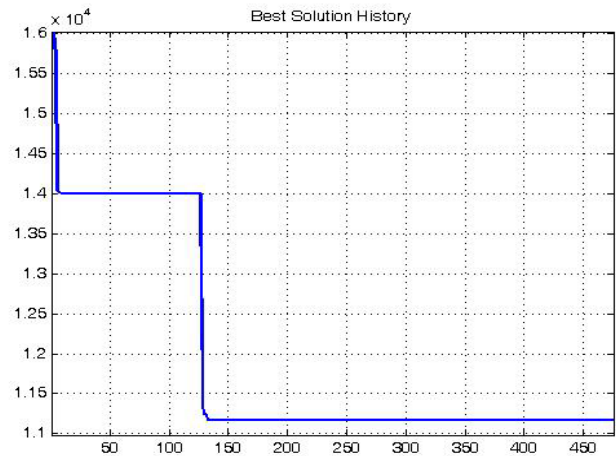


Figura 7. Número de soluções encontradas apenas através do algoritmo genético (800 iterações, 100 indivíduos, encontradas 12 soluções distintas)

Observando as nove simulações, pode-se concluir que o algoritmo genético, por si só, consegue encontrar a solução óptima, mas necessita de algum esforço computacional. Isso é claro através dos resultados obtidos nas soluções nºs 6, 7, 8, 9 e 10, onde, com um número de gerações e indivíduos diferentes, se obtiveram resultados diferentes. Não esquecendo o facto de um algoritmo genético ser uma solução de pesquisa aleatória direccionada, só haverá 100% de certeza de chegar ao ponto óptimo com infinitas iterações. A solução nº 5 confirma parcialmente este aspecto, apesar de não ser conclusivo, devido ao número de soluções parciais óptimas ser mais baixo do que no caso das soluções descritas abaixo, na tabela. Uma conclusão óbvia é o facto do algoritmo genético ter conseguido evoluir a solução nº 1 com um esforço computacional muito reduzido, devido à utilização dos operadores de manipulação diária ao nível dos indivíduos.

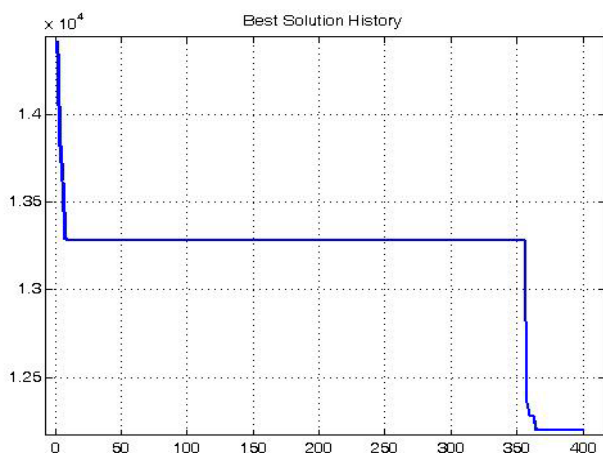


Figura 8. Número de soluções apenas com a utilização do algoritmo genético (400 iterações, 100 indivíduos, encontradas 11 soluções distintas)

Ainda relativamente à elevada utilidade em introduzir um indivíduo “pseudo-óptimo” na geração inicial do algoritmo genético, para originar uma diminuição drástica do custo computacional na procura da solução óptima, houve o cuidado e a necessidade de empregar “operadores aleatórios” de forma a garantir a não convergência do algoritmo para mínimos locais. Extremamente importante igualmente são as características dos operadores utilizados para “descender” cada geração com indivíduos mais “habilitados”. Os exemplos simulados mostram a importância dos operadores ao nível da troca diária das tarefas entre as várias equipas que elaboram trabalhos de manutenção no mesmo dia. Sem estes operadores, o algoritmo genético mostrou-se incapaz de evoluir a solução oferecida pelo Algoritmo I exposto na Fig. 5, para uma melhor solução.

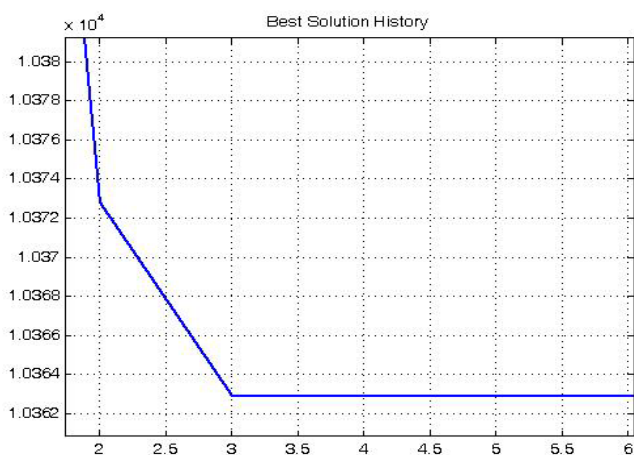


Figura 9. Número de soluções utilizando uma solução óptima inicial (1400 iterações, 100 indivíduos, encontradas 3 soluções distintas)

As Figs. 7, 8 e 9 evidenciam alguns dos aspectos apontados, nomeadamente as Figs. 7 e 8, onde foi utilizado apenas o algoritmo genético, dando origem a mais soluções parciais óptimas a cada iteração. A Fig. 9 evidencia a influência da solução indicada pelo Algoritmo I ao ser utilizada como uma das soluções iniciais no algoritmo genético. O nº de soluções parciais diminui dado a proximidade do ponto óptimo.

VII. CONCLUSÕES

O artigo expõe uma metodologia para solucionar o problema da afectação dos recursos humanos às acções de manutenção em geradores eólicos. A solução apresentada utiliza algoritmos genéticos. Os operadores utilizados para obtenção de percursos óptimos, dada a codificação usada, inicialmente, trabalhavam ao nível do dia apenas.

Paralelamente desenvolveu-se um algoritmo específico, baseado numa heurística, cuja solução é não óptima, mas com “qualidade” apreciável. O objectivo deste algoritmo era comparar a sua solução com a do algoritmo genético, e mais tarde esta ser incorporada no algoritmo genético como um ponto de partida, diminuindo o esforço computacional.

Constatou-se, após várias simulações, a ineficiência na convergência para soluções óptimas, do algoritmo genético comparativamente aos resultados originados com o algoritmo específico. As alterações introduzidas para ultrapassar esta limitação na implementação do algoritmo genético versaram a utilização dos mesmos operadores para recombinar as equipas técnicas que actuavam no mesmo intervalo de tempo. Esta ligeira alteração demonstrou eficácia, tendo o algoritmo genético produzido soluções superiores ao algoritmo específico. A inclusão da solução do algoritmo específico, como ponto de partida para o algoritmo genético, diminuiu o esforço computacional deste último, na procura da solução óptima, tal como foi previsto.

Para solucionar o problema concreto de encontrar o melhor caminho entre dois nós são propostas três alternativas: algoritmo da formiga, de *Dijkstra* e algoritmos genéticos. Os três algoritmos encontraram a solução óptima, mas com diferentes custos computacionais. O algoritmo de *Dijkstra* é o mais eficiente, seguido do algoritmo genético e do algoritmo da formiga. Este último tem a vantagem de se conseguir adaptar a uma matriz de custos temporalmente dinâmica, embora a exploração desta característica esteja fora do âmbito deste artigo.

Os resultados alcançados são bons, e como consequência dos desenvolvimentos efectuados, constata-se o interesse em evoluir, no futuro, para estudos visando a optimização através da programação linear, bem como da alteração do algoritmo genético de forma a obter soluções universais.

A metodologia apresentada neste artigo integra um sistema de informação mais vasto denominado SMIT, cujo objectivo deste é efectuar a gestão da manutenção de aero-geradores.

AGRADECIMENTOS

Este trabalho foi suportado financeiramente por uma bolsa da FCT com o contrato SFRH/BD/39225/2007, com co-financiamento do POPH/FSE e da UE.

REFERÊNCIAS

- [1] EWEA – *European Wind Energy Association*, Annual report of 2008, [Online]. Disponível em: www.ewea.org, visitado em 12 de Abril de 2010.
- [2] Farinha, J. M. T.; Vasconcelos, B. C., *SMITH - Sistema Modular Integrado de Terologia Hospitalar*, Actas do 4º Congresso Nacional de Manutenção Industrial, Porto, 1994.

- [3] J.Torres Farinha, et Al, *A Global View of Maintenance Management - From Maintenance Diagnosis to Know-how Retention and Sharing*, WSEAS Transactions on Systems, Issue 4, Volume 3, June 2004, ISSN 1109-2777, pp. 1703-1711.
- [4] Inácio Fonseca, J.Torres Farinha, F. P. Maciel Barbosa, *Wind Turbines Maintenance – an integrated approach*, CEE07, International Conference on Electrical Engineering, Coimbra, 26-28 November, 2007, Portugal.
- [5] Inácio Fonseca, J. M. Torres Farinha, F. P. Maciel Barbosa, *A computer system for predictive maintenance of wind generators*, 12th WSEAS International Conference on Computers, Heraklion, Greece, July 23-25, 2008, ISSN: 1790-5109, pages 928-933.
- [6] Emami, Alireza, Nogreh, Pirooz, *New approach on optimization in placement of wind turbines within wind farm by genetic algorithms*. Renewable Energy: An International Journal Jul 2010, Vol. 35 Issue 7, p1559-1564.
- [7] Grady S.A., Hussaini M.Y., Abdullah, M.M., *Placement of wind turbines using genetic algorithms*. Renewable Energy: An International Journal Feb. 2005, Vol. 30 Issue 2, p259-270.
- [8] Baskar S., Subbaraj P., Rao M.V.C., Tamilselvi S., *Genetic algorithms solution to generator maintenance scheduling with modified genetic operators*. IEE Proceedings on Generation, Transmission and Distribution, Volume 150, issue 1, p56-60.
- [9] M. Shahidehpour, M. Marwali, *Maintenance Scheduling in Restructured Power Systems*. Kluwer's Power Electronics and Power Systems Series, ISBN 0-7923-7872-5.
- [10] Jesse A. Andrawus, John Watson, Mohammed Kishk, *Wind Turbine Maintenance Optimisation: principles of quantitative maintenance optimisation*. Wind Engineering, Volume 31, Number 2 / March 2007.
- [11] David McMillan, Graham W. Ault, *Quantification of Condition Monitoring Benefit for Offshore Wind Turbines*. Wind Engineering, Volume 31, Number 4 / May 2007, pages 267-285.
- [12] MIT Open Courses, Eric Demaine, *Lecture 17: Shortest Paths I: Properties, Dijkstra's Algorithm, Breadth-first Search*, [Online]. Disponível em: videlectures.net/mit6046jf05_demaine Lec17, visitado em 3 de Fevereiro de 2010.
- [13] Opensource software and commercial hardware, [Online]. Disponível em: ni.com; vaisala.com; codegear.com; linux.com; slackware.org; php.org; apache.org; openssl.org, visitado em 3 de Fevereiro de 2010.
- [14] PostgreSQL, [Online]. Disponível em: www.postgresql.org, história em: www.postgresql.org/about/history, visitado em 3 de Fevereiro de 2010.
- [15] CPAU program for context switching, [Online]. Disponível em: www.joeware.net/freetools/tools/cpau/index.htm, visitado em 3 de Fevereiro de 2010.
- [16] *Scientific opensource software*, Octave and R, [Online]. Disponível em: www.gnu.org/software/octave e www.r-project.org, visitado em 3 de Fevereiro de 2010.
- [17] Manabu GOUKO and Koji ITO, *An Action Generation Model Using Time Series Prediction*, Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007.
- [18] Ricardo de A. Araújo, et all, *An Evolutionary Morphological-Rank-Linear Approach for Time Series Prediction*, 2007 IEEE Congress on Evolutionary Computation (CEC 2007), Singapore.
- [19] Yang Lan, Daniel Neagu, *A New Time Series Prediction Algorithm based on Moving Average of nth-order Difference*, Sixth International Conference on Machine Learning and Applications, 2007 IEEE.
- [20] Ming-Wei Chang, et all, *Analysis of nonstationary time series using support vector machines*, Dep. of Statistics, University of Taipei.
- [21] Inácio Fonseca, Torres Farinha, F. P. Maciel Barbosa, *On-Condition Maintenance for Wind Turbines*, IEEE, PowerTech 2009, apresentado em Bucharest.
- [22] *Advanced Maintenance and Repair for Offshore wind farms using fault prediction and Condition Monitoring Techniques*, E.U. final report of project NNE5/2001/710.
- [23] Mari Cruz Garcia, et Al, *SIMAP: Intelligent System for Predictive Maintenance Application to the health condition monitoring of a windturbine gearbox*, Science@Direct, Elsevier, 2006.
- [24] SNTP, [Online]. Disponível em: www.cis.udel.edu/~mills/database/rfc/rfc4330.txt, visitado em 3 de Fevereiro de 2010.
- [25] PTP, *Precision Time Protocol*, [Online]. Implementação do PTPD, disponível em: ptpd.sourceforge.net, visitado em 3 de Fevereiro de 2010.
- [26] A.John Kumar, J. Arunadevi, V. Mohan, *Intelligent Transport Route Planning Using Genetic Algorithms in Path Computation Algorithms*, European Journal of Scientific Research, ISSN 1450-216X Vol.25, Nº3 (2009), pp.463-468.
- [27] David Montana, Marshall Brinn, Sean Moore, Garrett Bidwell, *Using Genetic Algorithms for Complex, Real Time Scheduling*, Network Operations and Management Symposium, 1998. NOMS 98, IEEE.
- [28] Wolfgang Banzhaf Markus Braimeier, *Linear genetic programming - genetic and evolutionary computation*. First Edition, 2007, Springer, ISBN: 0-387-31029, pp. 261-287.
- [29] MIT Open Courses, Eric Demaine, *Lecture 18: Shortest Paths II: Bellman-Ford, Linear Programming, Difference Constraints*, [Online]. Disponível em: videlectures.net/mit6046jf05_demaine Lec18, visitado em 3 de Fevereiro de 2010.
- [30] Wikipedia. *Ant colony optimization*, [Online]. Disponível em: en.wikipedia.org/wiki/Ant_colony_optimization, visitado em 3 de Fevereiro de 2010.
- [31] Oliver van Kaick, Ghassan Hamarneh, Hao Zhang, and Paul Wighton. *Contour correspondence via ant colony optimization*. In Proc. 15th Pacific Conference on Computer Graphics and Applications (PG'2007), pages 271–280, 2007.
- [32] I. Zamboni e L.H. A. Monteiro. *Optimization of the topology of electric energy distribution networks by using algorithm inspired on ant behaviors*. In IEEE Latin America Transactions, Vol 7, No. 1, March 2009, pages 85–91, 2009.
- [33] FreeBSD, *FreeBSD the power to serve*, [Online]. Disponível em: www.freebsd.org, visitado em 12 de Abril de 2010.



Inácio Fonseca nasceu em 1971 em Coimbra. Concluiu a Licenciatura em Eng^a Electrotécnica na Universidade de Coimbra (curso de 5 anos), em 1994, e o Mestrado em 1998. É Professor no Instituto Superior Engenharia Coimbra e aluno de doutoramento na FEUP, Universidade do Porto. As Energias Renováveis são um dos seus tópicos de investigação.



José Torres Farinha nasceu em Lisboa, Portugal, em 1961. Concluiu a Licenciatura em Eng^a Electrotécnica na Universidade de Coimbra (curso de 5 anos), em 1984, e o Doutoramento em 1994 na Universidade do Porto.

Actualmente, é Professor Coordenador no Instituto Superior de Engenharia de Coimbra. O seu interesse científico principal é na área de gestão de activos e assuntos relacionados, tais como, sistemas de informação de gestão da manutenção, manutenção em condição e diagnóstico de falhas.



Fernando Maciel Barbosa (S'78-M'80-SM'94) licenciou-se em Eng^a Electrotécnica pela Universidade do Porto FEUP (curso de cinco anos), em 1971, e obteve o Mestrado e o Doutoramento em Sistemas de Energia na UMIST em 1977 e 1979, respectivamente. Os seus

interesses de investigação centram-se nos diferentes aspectos da Análise de Sistemas Eléctricos e interligação na rede dos Sistemas de Energia Renovável. É Professor no Departamento de Engenharia Electrotécnica e de Computadores na FEUP, desde 1971, sendo Professor Catedrático desde 1989. É membro do Cigré e participa no Cigré WG6, Sistemas de Distribuição e Geração Dispersa. É membro Conselheiro da Ordem dos Engenheiros Portuguesa.