



PROPOSTA DE ALGORITMO GENÉTICO PARA A SOLUÇÃO DO PROBLEMA DE ROTEAMENTO E SEQUENCIAMENTO DE SONDAS DE MANUTENÇÃO

Vanessa Rennó Frota Moraes Alves

UFRJ / COPPE / Programa de Engenharia de Produção

vanfrota@yahoo.com.br

Virgílio José Martins Ferreira Filho

UFRJ / COPPE / Programa de Engenharia de Produção

virgilio@ufrj.br

RESUMO

Uma das atividades de grande importância na indústria do petróleo é a intervenção em poços para a manutenção da sua produção. Em campos terrestres, esse serviço é executado por sondas do tipo SPT (Sondas de Produção Terrestre), que estão disponíveis em um pequeno número se comparado ao número de poços que demandam manutenção. O Problema de Roteamento e Sequenciamento de Sondas de Manutenção (PRSSM) consiste em achar o melhor “schedule” para o conjunto de SPT’s disponíveis, de forma a minimizar a perda de produção associada aos poços que estão aguardando serviço. Este trabalho apresenta uma proposta de solução para esse problema através da metaheurística Algoritmo Genético. E ainda apresenta resultados computacionais de testes realizados em instâncias artificiais.

Palavras-chave: Algoritmo Genético, Problema de Roteamento e Sequenciamento, Sondas de Manutenção

ABSTRACT

One of the activities of great importance in the petroleum industry is the operation for the maintenance of wells production. In terrestrial fields, this maintenance service is executed by workover rigs, that are available in a small number if compared with the number of wells demanding service. The Routing and Scheduling Workover Rigs Problem (RSWRP) consists in finding the best schedule for the available workover rigs, so as to minimize the production loss associated with the wells awaiting the service. This paper presents a solution to solve this problem proposing a Genetic Algorithm metaheuristic. Computational results on artificial instances of the problem are also presented.

Keywords: Genetic Algorithm, Routing and Scheduling Problem, Workover Rigs

1. INTRODUÇÃO

Uma das atividades de grande importância na indústria do petróleo é a intervenção em poços para a manutenção da sua produção. As intervenções de manutenção podem ser classificadas como de avaliação, recompletação, restauração, limpeza, estimulação, mudança de método de elevação e abandono. Esses serviços são realizados por sondas, que no caso das intervenções terrestres são denominadas sondas do tipo SPT (Sondas de Produção Terrestre) (Costa, 2005).

Devido ao seu elevado custo para aquisição e operação, existem poucas sondas se comparado ao número de poços que demandam manutenção. A decisão de qual sonda deve ser enviada para realizar determinado serviço de manutenção é baseada em fatores, tais como, produção do poço, atual localização das sondas de manutenção e do tipo de serviço que precisa ser executado (Aloise *et al.*, 2006).

Como exemplo real desse problema, citado em Costa (2005), temos a PETROBRAS, que possui uma unidade de negócios no norte do Estado do Espírito Santo (UN-ES). Tal unidade possui cerca de centenas de poços distribuídos em dezenas de campos distantes de 20km, na média entre si. Para atender às solicitações de serviços de intervenções de poços a empresa dispõe de apenas quatro sondas SPT, portanto, é de se esperar que com frequência aconteçam problemas de congestionamento e surgimento de filas para atendimento. Como em geral, os atendimentos visam recuperar a produção do poço, restabelecendo a sua vazão normal, o surgimento das filas causa prejuízos financeiros. O setor de intervenção de poços da empresa tem como meta a redução das perdas de vazões de todo o sistema atrelado a algumas restrições de política empresarial, bem como casos de avaliação exploratória que possuam prazos mais apertados definidos pela concessão dada pela Agência Nacional de Petróleo (ANP).

Dentro desse cenário, o que se quer, é resolver o problema de roteamento e sequenciamento de um conjunto de sondas do tipo SPT e um conjunto de poços que demandem serviços de intervenções, de forma a minimizar a perda das vazões de todo o sistema respeitando as restrições de datas a alguns serviços. Este problema será chamado de Problema de Roteamento e Sequenciamento de Sondas de Manutenção (PRSSM) com janela de tempo. Janela de tempo é o intervalo de tempo no qual o poço poderá ser trabalhado.

O PRSSM com janela de tempo foi estudado por Costa (2005), que o trata como Problema de Otimização de Itinerário de Sondas (POIS). Foram propostas uma modelagem matemática, duas heurísticas, e ainda, uma metaheurística GRASP (Greedy Randomized Adaptive Search Procedures) para a obtenção de soluções. Outros trabalhos que também estudaram o PRSSM foram o de Aloise *et al.* (2006) e Trindade (2005). O primeiro usa uma abordagem metaheurística VNS (*Variable Neighborhood Search*) e o segundo uma abordagem metaheurística GRASP.

O objetivo desse trabalho é propor uma metaheurística Algoritmo Genético para a solução do PRSSM sem janela de tempo. E ainda, usar essa metaheurística proposta para resolver o problema para o mesmo conjunto de dados usado por Costa (2005).

2. FORMULAÇÃO DO PROBLEMA

O PRSSM com janela de tempo foi modelado por Costa (2005) como um problema de programação linear inteira com variáveis de decisão binárias. Mas antes de apresentar a formulação do problema serão feitas algumas considerações iniciais. A cada poço que entra na fila de poços que demandam intervenção está associado:

- Um valor de perda de captação, que indica o quanto aquele poço está deixando de produzir em unidades de volume por unidade de tempo.

- Um intervalo de tempo no qual ele poderá ser trabalhado, que é a janela de tempo.
- O tempo de intervenção estimado, que depende unicamente do tipo de serviço no poço.

Não foram considerados os tempos de deslocamento entre cada par de poços. Dado que na região de um campo de produção de petróleo o deslocamento entre poços não é, em geral, um tempo significativo perto do tempo dos trabalhos nos poços. Não há distinção entre os possíveis tipos de serviços que as SPT's podem realizar, bem como não há distinção entre os equipamentos, e portanto, a frota é considerada homogênea.

Para atender a necessidade de se atribuir janelas de tempos às ordens de atendimento, o autor optou, por uma nova modelagem a qual, ao invés de trabalhar com ternas que indicam qual sonda vai trabalhar em qual poço em que ordem, trabalha com ternas que indicam em qual instante uma determinada sonda inicia os serviços em um determinado poço.

2.1. PARÂMETROS DE ENTRADA E VARIÁVEIS DE DECISÃO

Sejam os conjuntos:

$N = \{1..n\}$: o conjunto dos n poços que estão sujeitos à intervenção;

$M = \{1..m\}$: o conjunto das m sondas disponíveis e

$T = \{1..hp\}$: o conjunto dos instantes de tempo no horizonte de planejamento hp .

Os valores de tempos e datas são expressos em intervalos de tempos inteiros em uma unidade comum que melhor se adequar ao conjunto de dados.

Os parâmetros de entrada a seguir são considerados conhecidos e determinísticos.

P_i : a perda de vazão dada em m^3 /unidade de tempo;

d_i : a data de liberação para início dos serviços;

D_i : a data para término dos serviços;

Δt_i : o tempo de serviço no poço.

As variáveis de decisão X_{ikt} do problema são binárias e assumem o valor 1 quando a máquina k inicia os serviços no instante t no local i , e assume o valor 0 caso contrário. Logo, o problema possuirá $n \cdot m \cdot hp$ variáveis do tipo 0-1.

2.2. FORMULAÇÃO MATEMÁTICA

A função objetivo (1) procura minimizar as perdas de vazão calculando os produtos das perdas unitárias de vazão P_i pelos tempos de espera até a liberação do local $(t + \Delta t_i - d_i)$.

$$\text{Min} \quad \sum_t \sum_i \sum_k (t + \Delta t_i - d_i) \cdot P_i \cdot X_{ikt} \quad (1)$$

Sujeito a :

$$\sum_t \sum_k X_{ikt} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_i \sum_k \sum_{t=0}^{d_i-1} X_{ikt} = 0 \quad (3)$$

$$\sum_i \sum_k \sum_{t=D_i-\Delta t_i+1}^{hp} X_{ikt} = 0 \quad (4)$$

$$\sum_{j \in N} \sum_{t'=t-\Delta t_j+1}^{t'+\Delta t_j-1} X_{jkt'} \leq 1 \quad \forall k \in M; \quad \forall i \in N; \quad \forall t \in T \quad (5)$$

$$X_{ikt} \in \{0,1\} \quad \forall i \in N; \quad \forall k \in M; \quad \forall t \in T \quad (6)$$

A igualdade (2) é a restrição de atendimento, e indica que cada poço i deve ser atendido exatamente uma única vez e por uma única sonda. As igualdades (3) e (4) são as restrições de janela de tempo, elas garantem que todo local i não poderá começar a ser atendido por qualquer sonda k após o instante $(D_i - \Delta t_i)$, nem antes de d_i . Estas duas restrições em conjunto com a restrição (2) nos permite dizer que o local i deverá estar pronto até a data devida D_i . A desigualdade (5) é a restrição de sonda e ininterruptção dos serviços, ela garante que cada sonda em cada instante de tempo só inicie o serviço em até no máximo um poço. E ainda garante que quando uma máquina k inicia os trabalhos no local i no instante t ela fica indisponível para iniciar outros trabalhos nos instantes t' compreendido na janela de tempo $[t, t + \Delta t_i]$ em todos os outros locais j diferentes de i . As variáveis X_{ikt} são definidas como binárias em (6).

Podemos perceber que o número de restrições para o problema é grande. Somando o número de equações dos conjuntos de restrição (2) a (5) temos o total de $(n+1+1+ n \cdot m \cdot hp)$ restrições. Para um caso onde se tenha 125 poços aguardando manutenção, 10 sondas disponíveis e um horizonte de tempo de 40 unidades de tempo, o número de restrições chega a 50127! Portanto, é preciso encontrar alternativas viáveis de se resolver o problema, ou de pelo menos obter soluções viáveis e de boa qualidade. Uma opção é o uso de metaheurísticas, que são estruturas algorítmicas que guiam outras heurísticas na busca por soluções viáveis em domínios onde essa tarefa é difícil, como nos problemas de otimização combinatória.

3. METAHEURÍSTICA ALGORITMO GENÉTICO (AG)

O Algoritmo Genético, pertencente à classe dos algoritmos evolutivos, tem o seu funcionamento baseado na analogia com a teoria das espécies de Charles Darwin, onde indivíduos que representam um ponto no espaço de busca são incentivados a se recombinarem gerando novas soluções (filhos) que são avaliados e reintroduzidos na população substituindo os pais. Esses procedimentos são repetidos até que um critério de parada seja satisfeito retornando-se a melhor solução encontrada (Aloise *et al.*, 2000).

As possíveis soluções do problema a ser otimizado são representadas por estruturas de dados chamadas cromossomos. Nos algoritmos genéticos clássicos os cromossomos tem uma estrutura de cadeia de bits $\{0,1\}$. Porém essa representação binária não atende ao problema de busca de soluções em um espaço de permutações, devido à dificuldade de se associar cadeias binárias a permutações de valores inteiros e ainda de se preservar essas permutações durante os processos de crossover e mutação. Então, a maneira de se contornar esse problema é usar cromossomos com uma estrutura de um vetor que armazene essas permutações, e aplicar operadores de crossover (tais como order crossover ou PMX – partially matched crossover) e mutação (tal como trocas aleatórias entre dois elementos da permutação) que preservem as permutações (Bäck & Schwefel, 1996).

Esse tipo de representação de cromossomos, como permutações, aparece nos trabalhos de Almeida *et al.* (2003) e Aloise *et al.* (2000). O primeiro caso trata do problema de otimização do scheduling de produção na divisão de óleo combustível e asfalto de uma refinaria de petróleo. Essa divisão é uma planta multi-produto com dois estágios de máquinas, um mixer e um conjunto de tanques, sem tempo para setup e com operação contínua. O algoritmo genético é usado para se estabelecer a seqüência de produção e o tamanho dos lotes de todos os produtos demandados. O cromossomo, neste caso, representa a seqüência de prioridades na hora de fazer o despacho desses produtos. No segundo caso o algoritmo genético busca otimizar o emprego de uma única unidade móvel de pistoneio (UMP) na extração diária de petróleo. Os cromossomos representam a ordem em que os poços são atendidos pela UMP. Este problema é semelhante ao PRSSM com apenas uma sonda de manutenção disponível.

Os passos da execução do algoritmo genético implementado nesse trabalho são apresentados no pseudocódigo da Figura 1 e detalhados nas subseções a seguir.

```

1. Gera população inicial
2. ENQUANTO critério de parada não for atingido FAZ
    2.1 Avalia população
    2.2 Seleciona indivíduos para reprodução
    2.3 Gera descendência: aplicação do operador de crossover
    2.4 Aplica operador de mutação
FIM ENQUANTO
    
```

Figura 1 – Pseudocódigo do algoritmo genético

3.1. REPRESENTAÇÃO E AVALIAÇÃO DOS CROMOSSOMOS

Como já foi dito, o objetivo do PRSSM é alocar n poços a m sondas de manutenção de forma a minimizar a vazão de óleo perdida. Dada uma sequência qualquer dos n poços de manutenção, esta pode ser entendida como a sequência com que os poços devem ser alocados às sondas, e assim pode ser calculado qual foi a perda de vazão devido a essa estratégia de alocação. Logo, para essa abordagem do problema, o espaço de busca de soluções compreende o conjunto de permutações dos poços que demandam manutenção.

Com isso, a estrutura usada para o cromossomo é a de um vetor de números inteiros, onde cada inteiro representa um poço que está necessitando intervenção.

A decodificação do cromossomo é feita através da seguinte regra: começando na primeira posição, para cada poço i que está armazenado no cromossomo, alocar na primeira sonda disponível.

Terminada a alocação dos poços, é calculado o total da vazão perdida.

A função de avaliação do cromossomo é definida abaixo

$$f = \sum_i P_i \cdot (t_i + \Delta t_i),$$

onde P_i é a perda de vazão dada em m^3 /unidade de tempo, t_i é o instante em que o poço começa a ser atendido, e Δt_i é o tempo de intervenção no poço.

Para ilustrar a regra de decodificação do cromossomo, considere o conjunto hipotético de dados da Tabela 1, abaixo.

i	1	2	3	4	5
P_i	10	30	40	1	30
Δt_i	1	4	2	1	2

Tabela 1 – Conjunto de dados para ilustração da decodificação do algoritmo

Dado o cromossomo

2	4	1	5	3
---	---	---	---	---

a sua decodificação, para uma situação onde se tenha 3 sondas de manutenção, é mostrada na Figura 2 na forma de diagrama de Gantt.



Figura 2 – Sequenciamento dos 5 poços em 3 sondas de manutenção

A avaliação desse cromossomo é dada por

$$f = \sum_i P_i \cdot (t_i + \Delta t_i) = [10 \cdot (0 + 1)] + [30 \cdot (0 + 4)] + [40 \cdot (1 + 2)] + [1 \cdot (0 + 1)] + [30 \cdot (1 + 2)] = 341$$

3.2. DEFINIÇÃO DA POPULAÇÃO INICIAL E MÉTODO DE SELEÇÃO

As permutações que formam a população inicial são escolhidas de forma totalmente aleatória. O processo de seleção constrói uma população intermediária, de onde serão selecionados os cromossomos pais para a operação de crossover. A seleção é feita por torneio binário. Neste caso, um par de cromossomos é escolhido aleatoriamente na população atual e o cromossomo com o menor valor para função de avaliação é copiado para a população intermediária, e então o par de cromossomos são recolocados na população original (Beasley *et. al.*, 1993a). O processo se repete até preencher a população intermediária, que neste caso tem o mesmo tamanho da população atual.

3.3. DEFINIÇÃO DOS OPERADORES DE Crossover E MUTAÇÃO

A escolha dos operadores de crossover e mutação foram feitas com base no trabalho de Almeida *et. al.* (2003).

3.3.1. Uniform order based crossover

Esse operador é o operador de crossover uniforme aplicado a representações inteiras de cromossomos. No crossover uniforme, para representações binárias, o filho 1 é criado copiando-se os genes correspondentes de um dos pais, escolhidos de acordo com uma máscara de (0's e 1's) gerada aleatoriamente. Onde existir um 1 na máscara, o gene é copiado do primeiro pai, e onde existir um 0 na máscara, o gene é copiado do segundo pai. O processo é repetido com a troca dos pais para gerar o filho 2. Uma máscara nova é gerada para cada par de pais (Beasley *et. al.* 1993).

Juntando o crossover uniforme com o order based crossover (Whitley & Yoo, 1994) temos o uniform order based crossover que funciona da seguinte forma: uma máscara é gerada aleatoriamente, os genes do pai 2 correspondentes à posição 1 na máscara são localizados no pai 1 e são reordenados e copiados no filho 1, então eles aparecem na mesma ordem relativa que

aparecem no pai 2, mas nas posições ocupadas pelo pai 1. Depois, os elementos no pai 1 que não corresponderem aos elementos selecionados no pai 2 são passados diretamente para o filho 1.

Exemplo:

Pai 1:	A	B	C	D	E	F	G
Pai 2:	C	F	E	B	A	D	G
Máscara:	0	1	0	1	1	0	0

Os elementos selecionados no pai 2 são F, B e A. Assim, esses elementos são reordenados no pai 1.

Reordenar os elementos **A B _ _ F _** do pai 1 produz: **F B _ _ A _**.

Todos os outros elementos são copiados diretamente do pai 1.

(**F B _ _ A _**) combinado com (**_ _ C D E _ G**) produz o filho 1: F B C D E A G.

Os pais 1 e 2 são trocados para gerar o filho 2.

Dada a população intermediária, cada cromossomo tem uma probabilidade, chamada de taxa de crossover, para participar dessa operação. E após a geração dos filhos, estes substituem seus pais, na população intermediária.

3.3.2. Mutação por troca

Após as operações de crossover terem sido aplicadas à população intermediária, a mutação é aplicada a cada cromossomo a uma dada probabilidade denominada de taxa de mutação. Caso o cromossomo seja selecionado, dois genes são escolhidos aleatoriamente e suas posições são trocadas. O novo cromossomo substitui, imediatamente, o antigo.

Após esse último passo, a população intermediária substitui totalmente a população atual, e será avaliada na próxima iteração do algoritmo.

4. RESULTADOS COMPUTACIONAIS

Para testar a qualidade das soluções obtidas pelo algoritmo genético, foram realizados experimentos com o mesmo conjunto de dados utilizado por Costa (2005). E os resultados foram comparados com os resultados obtidos pelas duas heurísticas propostas nesse mesmo trabalho. O conjunto de dados usados foi gerado artificialmente pelo autor com base no comportamento dos valores de vazão e de tempos de intervenções em poços analisados a partir de um banco de dados real, fornecido pela unidade de negócios da PETROBRAS no norte do Estado do Espírito. Não foram consideradas instâncias com janelas de tempo.

Foram estabelecidos cinco grupos de problemas baseados na quantidade de poços demandando intervenção: Grupo 25, Grupo 50, Grupo 75, Grupo 100 e Grupo 125. Para cada grupo foram geradas 10 instâncias, chamadas de A, B, C, ..., I, J. Com isso, foram gerados 50 exemplos de problemas. Cada problema foi resolvido para 6 quantidades diferentes de sonda de manutenção, são elas: 1, 2, 4, 6, 8 e 10 sondas. Ao todo foram resolvidas 300 instâncias, sendo 60 para cada grupo. Cada uma das instâncias é rotulada com o código PNX-M, onde N é a quantidade de poços, X é a letra do exemplo (varia de A a J) e M é a quantidade de sondas SPT.

No conjunto de dados criado, os valores da perda de vazão e tempo de intervenção nos poços tiveram que ser modificados para se tornarem valores inteiros. Para a conversão dos dados apresentados nos resultados dos testes, basta dividir o valor da Função Objetivo por 20. Por exemplo, um valor igual a 1000 representa uma perda de captação de 50m^3 ($1000/20$).

As heurísticas com as quais o algoritmo genético foi comparado são, a Heurística de Máxima Prioridade Tricritério (HMPT) e a Heurística de Montagem Dinâmica (HMD). Essa duas heurísticas foram resolvidas usando 3 critérios diferentes para a alocação de um novo poço nas sondas, são eles: critério 1 – menor valor da vazão perdida (P_i), critério 2 – menor valor de $P_i/\Delta t_i$ e critério 3 – menor valor de $P_i * \Delta t_i$.

O algoritmo genético foi implementado em MATLAB 6.5. Os experimentos computacionais foram executados em um 800 MHz Pentium III com 512 Mbytes de memória RAM, plataforma Windows. Os valores adotados para o tamanho da população (80 indivíduos), taxa de crossover (90% - 70%) e taxa de mutação (5% - 35%) foram baseados no trabalho de Almeida *et. al.* (2003). Esses valores foram mantidos constantes em todos os experimentos. As taxas de crossover e mutação variam linearmente durante as gerações. No começo do processo evolutivo, a taxa de crossover é alta enquanto que a taxa de mutação é baixa, então o algoritmo pode tirar vantagem das características individuais. No fim da evolução, que é a fase em que os indivíduos tendem a ficar parecidos, a taxa de crossover cai enquanto a de mutação aumenta, então o algoritmo pode explorar novas regiões no espaço de busca e aumentar a diversidade da população.

Foram utilizados dois critérios de parada: número máximo de gerações (300) ou quando o valor da função objetivo (perda de vazão) era obtido por 15 gerações consecutivas. O número médio de gerações obtido para a execução do algoritmo genético é dado na Tabela 2 a seguir.

Grupo 25	Grupo 50	Grupo 75	Grupo 100	Grupo 125
70	156	271	300	300

Tabela 2 – Número médio de gerações para execução do AG

O número máximo de gerações adotado foi alto para se ter uma idéia de quando o valor da função objetivo se torna relativamente estável, ou seja, se mantém igual por 15 gerações. Mesmo com esse alto valor de gerações, as instâncias dos Grupos 100 e 125 foram resolvidas, em média, em 97 seg. e 122 seg., respectivamente.

Cada uma das 300 instâncias geradas foi resolvida por cada uma das duas heurísticas, em seus três possíveis critérios. São eles HMPT1, HMPT2, HMPT3, HMD1, HMD2 e HMD3. Para o grupo de 25 poços, as instâncias ainda foram resolvidas pelo solver CPLEX 9.0. Conhecendo a solução ótima destas instâncias foi possível fazer uma análise do GAP do algoritmo genético em relação a esses valores. O valor do GAP foi calculado usando a seguinte relação: $GAP = (\text{resultado AG} - \text{resultado CPLEX}) / \text{resultado CPLEX}$. Os valores de GAP calculados foram agrupados no histograma mostrado no Gráfico 1.

Os resultados obtidos pelo Algoritmo Genético para o grupo de 25 poços se mostraram muito satisfatórios, já que os resultados das instâncias não excederam a solução ótima em mais de 0,6%. Na análise realizada em Costa (2005), as melhores heurísticas (HMPT2, HMD1, HMD2 e HMD3) não excedem a mais que 1,79% do valor ótimo.

A Tabela 3, que contém o resumo do número de vitórias alcançadas por cada método de solução após a execução das 300 instâncias, reforça o melhor desempenho do algoritmo genético em relação às heurísticas para o Grupo 25. E também indica que esse desempenho cai com o aumento do número de poços que necessitam de manutenção.

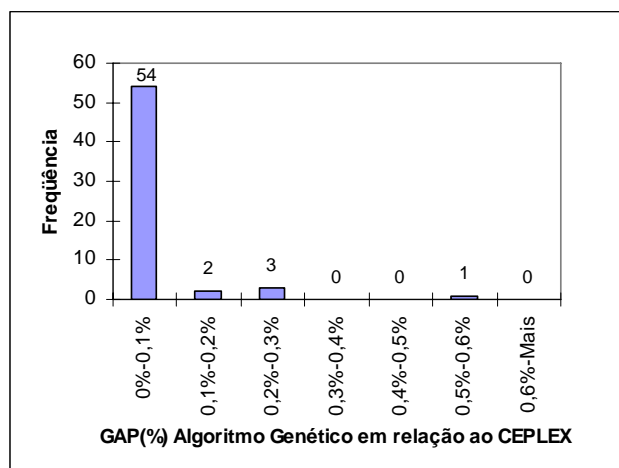


Gráfico 1 – Histograma do GAP(%) do Algoritmo Genético em relação ao CPLEX 9.0, para instâncias do Grupo 25.

	AG	HMPT1	HMPT2	HMPT3	HMD1	HMD2	HMD3
Grupo 25	50 83.33%	0 0.00%	13 21.67%	0 0.00%	11 18.33%	13 21.67%	11 18.33%
Grupo 50	46 76.67%	0 0.00%	12 20.00%	0 0.00%	10 16.67%	12 20.00%	10 16.67%
Grupo 75	41 68.33%	0 0.00%	19 31.67%	0 0.00%	10 16.67%	19 31.67%	10 16.67%
Grupo 100	27 45.00%	0 0.00%	30 50.00%	0 0.00%	11 18.33%	30 50.00%	12 20.00%
Grupo 125	13 21.67%	0 0.00%	40 66.67%	0 0.00%	13 21.67%	40 66.67%	14 23.33%

Tabela 3 – Resumo das vitórias em todos os grupos

Para uma melhor visualização de como os resultados se comportam com o aumento do número de poços que demandam intervenção, foram apresentados nos Gráfico 2, Gráfico 3 e Gráfico 4, apenas os resultados das instâncias artificiais caracterizadas pela letra A.

Embora exista um decrescimento no número de vitórias do AG em relação às heurísticas propostas por Costa, os Gráfico 2 e Gráfico 3 mostram que os resultados obtidos por ambos os métodos são praticamente os mesmos. Foi feito o cálculo do GAP dos resultados obtidos pelo AG em relação aos melhores resultados obtidos pelas heurísticas HMPT e HMD.

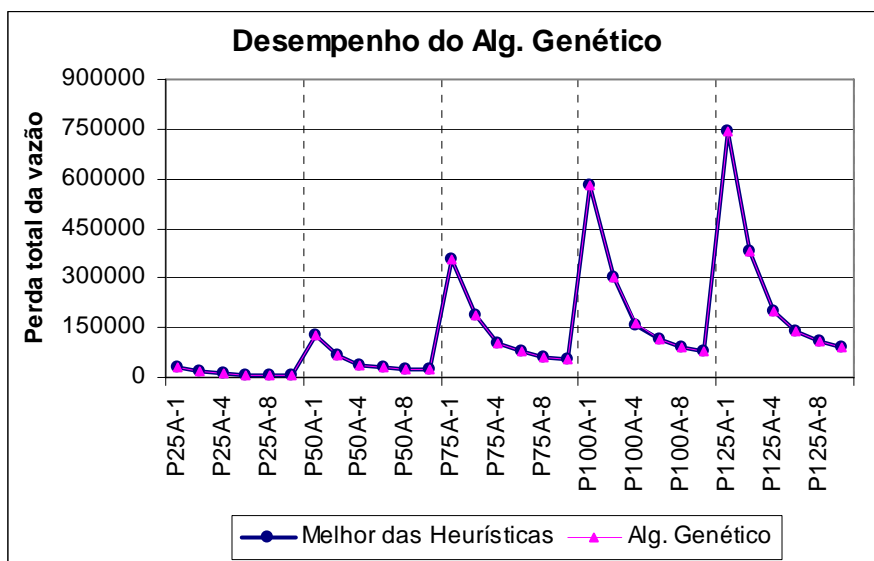


Gráfico 2 – Comparação entre os resultados obtidos pelo Algoritmo Genético e os melhores resultados obtidos pelas heurísticas HMTP e HMD

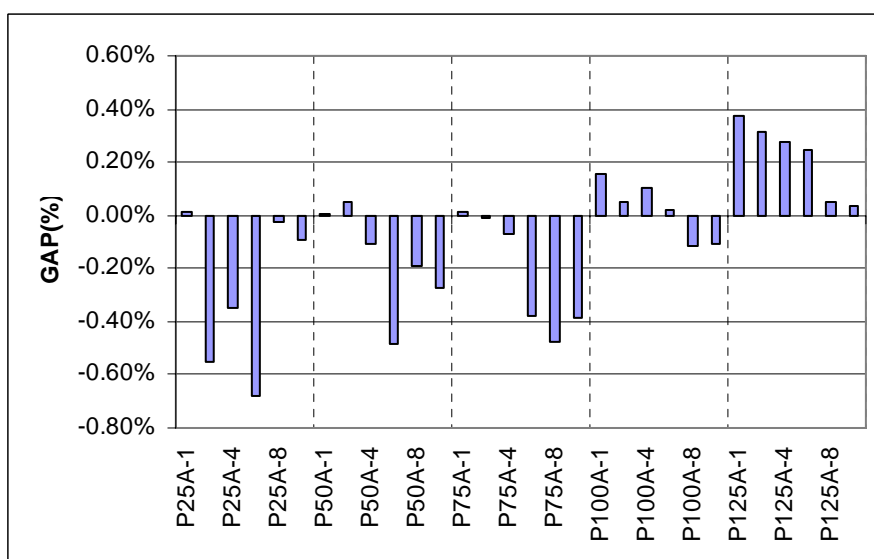


Gráfico 3 – Histograma do GAP(%) do Algoritmo Genético em relação ao melhor resultado obtido pelas heurísticas HMTP e HMD

O maior GAP encontrado para essas instâncias, foi inferior a 0,4%, indicando que o resultado obtido pelo AG excedeu em 0,4% o melhor resultado obtido pelas heurísticas, para essa instância. Na outra direção, o menor GAP encontrado ficou próximo ao valor de -0,7%, indicando que o resultado obtido pelo AG é 0,7% melhor que o melhor resultado obtido pelas heurísticas para essa instância. Para as demais instâncias, o comportamento dos resultados encontrados foi semelhante, os valores de GAP ficaram sempre no intervalo [-0,8% , 0,8%].

O Gráfico 4, apresenta o comportamento do tempo de processamento do AG com o aumento do número de poços. Tal comportamento se repete nas demais instâncias, sendo que os tempos de processamento não ultrapassam 140 segundos (2 min. e 20 seg.).

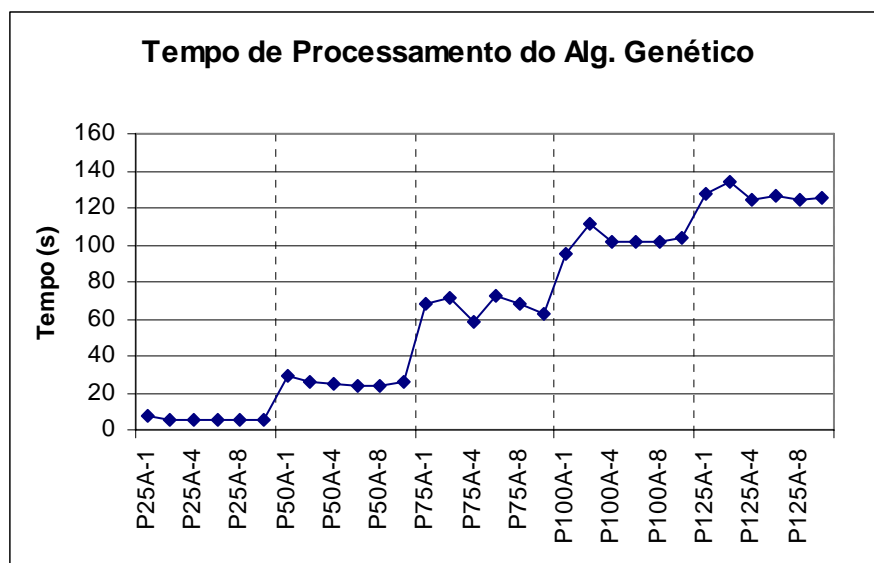


Gráfico 4- Tempo de execução do Algoritmo Genético

5. CONCLUSÕES

Neste trabalho foi proposta uma metaheurística algoritmo genético para a solução de problemas de roteamento e sequenciamento de sondas de manutenção. De acordo com os resultados computacionais, esse método produz soluções de boa qualidade para as instâncias de 25 poços. Em 90% dos casos o GAP(%) em relação à solução exata obtida pelo CPLEX 9.0 foi inferior 0,1%. Para as instâncias com um número maior de poços não foi possível comparar os resultados obtidos pelo AG com as soluções exatas. Na comparação dos resultados do AG com os melhores resultados obtidos usando as heurísticas MTP e MD foi verificado que os valores da função objetivo ficaram próximos. O GAP do resultados do AG em relação ao melhor resultado obtido das heurísticas ficou no intervalo $[-0,8\%, 0,8\%]$ para todas as instâncias executadas.

Pela análise feita em Costa (2005), entre as heurísticas propostas, a melhor é a HMD. Como todos os critérios utilizados em algum instante podem obter solução melhor que os demais, e o tempo de execução da heurística é mínimo, é recomendado efetuar a solução nos três critérios da HMD para obtenção de melhores resultados.

Pelos estudos realizados não foi possível constatar uma vantagem significativa no uso do AG para a solução do problema apresentado, e nem foi possível desmerecer esse método. Então, como o tempo de execução do AG é inferior a 2,5 min., a sugestão é que as soluções do problema sejam obtidas pelo AG e pela HMD para que se possa escolher o melhor resultado.



6. REFERÊNCIAS

- ALMEIDA, M.R., HAMACHER, S., PACHECO, M.A.C., VELLASCO, M.M.B.R., 2003, "Optimizing the Production Scheduling of a Petroleum Refinery Through Genetic Algorithms", *International Journal Of Industrial Engineering Theory Applications And Practice*, USA, v. 10, n. 1, p. 35-44.
- ALOISE, D.J., NEVES, J.A., BARROS, C.A., MOURA, L.S.S., ASSMANN, B.W., 2000, "Um algoritmo genético na otimização do emprego da unidade móvel de pistoneio", *Anais do XXXII Simpósio Brasileiro de Pesquisa Operacional*, Viçosa – MG, 18 a 20 de outubro de 2000.
- ALOISE, D.J., ALOISE, D., ROCHA, C.T.M., RIBEIRO, C.C., RIBEIRO FILHO, J.C., MOURA, L.S.S., 2006, "Scheduling workover rigs for onshore oil production", *Discrete Applied Mathematics*, 154(5), pp.695-702.
- BÄCK, T., SCHWEFEL, H.P., 1996, "Evolutionary Computation: An Overview", *Proceedings of the Third IEEE Conference on Evolutionary Computation 1996*, pp. 20-29, IEEE Press, Piscataway NJ.
- BEASLEY, D., BULL, D.R., MARTIN, R.R., 1993a, "An Overview of Genetic Algorithms: Part 1, Fundamentals", *University Computing*, 15(2), pp. 58-69.
- BEASLEY, D., BULL, D.R., MARTIN, R.R., 1993b, "An Overview of Genetic Algorithms : Part 2, Research Topics", *University Computing*, 15(4), pp. 170-181.
- COSTA, L.R., 2005, "Soluções para o problema de otimização de itinerário de sondas", Dissertação de Mestrado, Universidade Federal do Rio de Janeiro.
- TRINDADE, V.A., 2005, "Desenvolvimento e análise experimental da metaheurística GRASP para um problema de planejamento de sondas de manutenção", Dissertação de Mestrado, Universidade Federal Fluminense.
- WHITLEY, D., YOO, N.W., 1994, "Modeling Simple Genetic Algorithms for Permutation Problems", *Foundations of Genetic Algorithms (FOGA 1994)*, pp. 163-184.