

NOVAS SOLUÇÕES PARA O PROBLEMA DA ALOCAÇÃO DE SONDAS DE PRODUÇÃO A POÇOS DE PETRÓLEO COM UM GRASP+PATH-RELINKING**Alexandre Venturin Faccin Pacheco¹**
Glaydston Mattos Ribeiro^{1,2}¹Universidade Federal do Espírito Santo – UFES
Rodovia BR 101 Norte, Km 60, São Mateus/ES – Brasil - CEP 29932-540
alexandreventurin@gmail.com ; glaydstonribeiro@ceunes.ufes.br²University of Montreal
3000 Chemin de la Côte-Sainte-Catherine, Montreal/QC – Canadá - H3T 2A7
glaydston.ribeiro@cirreil.ca**Geraldo Regis Mauri**Universidade Federal do Espírito Santo – UFES
Alto Universitário s/n, Guararema, Alegre/ES – Brasil - CEP 29500-000
mauri@cca.ufes.br**RESUMO**

Os poços de petróleo em terra dependem de intervenções como recompletação, restauração, limpeza e estimulação para que se mantenham produtivos. Tais intervenções são desempenhadas por um número reduzido de sondas de produção terrestre, o que demanda uma programação de utilização. Vários fatores interferem nessa programação como a produção do poço, o tipo de serviço e a janela de tempo destinada ao serviço. Quando um poço necessita de intervenção, sua produção pára e só é re-estabelecida após a execução do serviço necessário. Assim, o Problema da Alocação de Sondas de Produção a Poços de Petróleo (PASPPP) consiste em definir, a melhor sequência de atendimento dos poços para um conjunto de sondas, minimizando a perda total de produção. Este artigo apresenta um GRASP com *Path Relinking* (PR) para o PASPPP. Bons resultados computacionais foram observados utilizando instâncias obtidas a partir de dados reais de um campo de produção localizado no Espírito Santo.

**PALAVRAS-CHAVE. GRASP. Path-Relinking. Sondas de Intervenção.
OC - Otimização Combinatória.****ABSTRACT**

Onshore oil wells depend on special services like reinstatement, cleaning and stimulation to keep them efficient. These services are usually performed by a short number of workover rigs, consequently, a scheduling must be determined. Several factors interfere on this scheduling such as production, service to be performed on each well and time windows for each service. When a well needs some service, its production is stopped and restarted only after the required service. Thus, the workover rig scheduling problem consists in finding the best sequence of wells for each workover rig, minimizing the total production loss. We present in this paper a Greedy Randomized Adaptive Search Procedure (GRASP) with Path-Relinking (PR) to solve this problem. Good computational results on instances obtained from real problems of a Brazilian oil field located at Espírito Santo State are reported.

**KEYWORDS: GRASP. Path-Relinking. Workover rigs.
CO - Combinatorial Optimization.**

1. Introdução

Uma maneira eficaz de se alcançar um diferencial competitivo na indústria petroleira é gerenciar de maneira eficiente os recursos críticos. Os poços de petróleo em terra, por serem em sua grande maioria não-surgentes, utilizam métodos de elevação artificiais executados por equipamentos dispendiosos, de custo elevado, operando sobre condições extremas durante longos períodos de tempo. Eventualmente, esses equipamentos necessitam de serviços de manutenção (intervenções), como avaliação, recompletação, restauração, limpeza ou estimulação, para manter a eficiência da extração. Para esses serviços, as empresas utilizam Sondas de Produção Terrestre (SPT's), ou simplesmente sondas.

Assim, dado um conjunto de poços que necessitam de intervenção e um conjunto de sondas disponíveis, deve-se efetuar uma programação de atendimento. Quando um poço necessita de intervenção sua produção pára. Portanto, é importante executar o serviço requisitado tão logo seja possível para que a produção volte ao normal. O não atendimento dentro dos prazos estabelecidos por um poço (janela de tempo) pode torná-lo inativo (Neves, 2007).

De acordo com Aloise *et al.* (2006), a perda de produção de cada poço, aguardando manutenção, é calculada sua produção média diária média, em regime de operação regular, pelo número de dias em que o mesmo ficou sem operar. Sendo assim, o Problema da Alocação de Sondas de Produção a Poços de Petróleo (PASPPP) consiste em encontrar a melhor sequência de atendimento de poços para cada sonda disponível, minimizando a perda de produção dos poços que aguardam intervenção e garantindo que: todos os poços sejam atendidos, cada poço seja alocado a uma única sonda e as janelas de tempo de cada poço sejam respeitadas.

Este artigo propõe a utilização do algoritmo *Greedy Randomized Adaptive Search Procedure* (GRASP) com *Path-Relinking* (PR) para o PASPPP. Resultados computacionais obtidos mostram que o GRASP+PR gera soluções melhores do que outras abordagens descritas na literatura. Este artigo é ainda resultado do trabalho de Iniciação Científica do aluno Alexandre Venturin Faccin Pacheco, orientado pelos Professores Glaydston Mattos Ribeiro e Geraldo Regis Mauri, todos da Universidade Federal do Espírito Santo (UFES).

O trabalho é organizado como segue: inicialmente é realizada uma breve revisão bibliográfica do problema na Seção 2. A modelagem matemática de programação linear inteira 0-1, proposta por Costa e Ferreira Filho (2004), é explicada na Seção 3. Na Seção 4 são apresentados os detalhes do GRASP+PR proposto. Na Seção 5 são exibidos os resultados computacionais obtidos, e as conclusões são apresentadas na Seção 6.

2. Breve Revisão Bibliográfica do PASPPP

Quando uma instância do PASPPP possui apenas uma sonda e o tempo de deslocamento (custo de transporte) da mesma não é considerado, Smith (1956) provou que a programação ótima é obtida quando os poços são ordenados de maneira decrescente de $P_i/\Delta t_i$, em que P_i é a perda de produção diária do poço i e Δt_i é o tempo estimado de intervenção no poço i . Essa ordenação é chamada de *Ordem Natural*.

Noronha *et al.* (2001) propôs uma heurística gulosa para um problema similar ao PASPPP. Nesse caso, foram consideradas prioridades para os poços baseadas na perda de produção e riscos ambientais envolvidos no serviço. Um GRASP foi proposto pelos autores.

Aloise *et al.* (2006) levaram em consideração os custos de transporte das sondas, o tempo necessário para cada intervenção e o tempo de deslocamento. Os autores propuseram uma heurística de Busca em Vizinhança Variável (VNS). As soluções iniciais foram geradas por uma heurística construtiva e nove definições diferentes de vizinhança foram utilizadas.

Costa e Ferreira Filho (2004; 2005) e Costa (2005) apresentaram uma formulação matemática para o problema, duas heurísticas construtivas e um GRASP para a solução do PASPPP com janelas de tempo. Costa (2005) construiu instâncias baseadas em casos reais de um campo de exploração de petróleo situado no norte do estado do Espírito Santo. Essas instâncias

não utilizam custo de transporte, pois os poços se encontram próximos uns aos outros, com uma distância média de aproximadamente 20km. Porém, essas instâncias apresentam janelas de tempo para as intervenções em cada poço e um horizonte de planejamento.

Costa (2005) propôs um GRASP, entretanto, os resultados obtidos foram inferiores aos encontrados pela Heurística de Máxima Prioridade Tricritério (Costa e Ferreira Filho, 2004) e pela Heurística de Montagem Dinâmica (Costa e Ferreira Filho, 2005). O GRASP proposto neste artigo possui três principais diferenças em relação ao proposto por Costa (2005), listadas na Seção 4.

As instâncias desenvolvidas por Costa (2005) têm sido utilizadas por vários autores. Alves e Ferreira Filho (2006) apresentaram um Algoritmo Genético em que as soluções (cromossomos) eram representadas por um vetor de poços. Para o operador de cruzamento foi utilizado o *Order Uniform Crossover* e, para operador de mutação, a troca aleatória de dois poços. Douro e Lorenzoni (2009) propuseram também um Algoritmo Genético, mas com trocas *2-Opt*. Esse algoritmo cria a solução inicial usando um método similar ao proposto por Alves e Ferreira Filho (2006) para distribuir os poços nas sondas. Como operador de cruzamento, foi utilizado o *Single Point*. Para o operador de mutação, os autores utilizaram três técnicas: o cruzamento assexuado, a mutação por inversão e a troca aleatória de dois poços. Eles ainda apresentaram uma estratégia baseada na troca de dois poços (*2-Opt*) para melhorar as soluções obtidas. Esse algoritmo apresentou resultados melhores do que a maioria dos outros métodos, como mostra a Tabela 1.

Pacheco *et al.* (2009) recentemente propuseram a heurística *Bubble Swap*, na qual as soluções foram representadas por matrizes: as colunas definem os poços, e as linhas as sondas. Essa heurística é rápida, porém em alguns casos as soluções não são boas.

3. Formulação Matemática para o PASPPP

Costa e Ferreira Filho (2004) apresentaram um modelo matemático de programação linear inteira 0-1 para o PASPPP com janelas de tempo. Seja $N = \{1,2,\dots,n\}$ o conjunto de n poços que demandam serviços de intervenção, $M = \{1,2,\dots,m\}$ o conjunto de m sondas e $T = \{1,2,\dots,hp\}$ o conjunto de instantes de tempo no horizonte de planejamento hp .

Cada poço i possui uma perda de vazão P_i , um tempo de intervenção Δt_i e uma janela de tempo $[e_i, l_i]$ para que o poço i receba uma sonda, sendo e_i o início e l_i o final da janela de tempo. A perda de vazão termina quando a sonda conclui seus serviços no poço. O modelo matemático proposto por Costa e Ferreira Filho (2004) é apresentado abaixo.

$$v(PASPPP) = \text{MIN} \sum_{i \in N} \sum_{j \in M} \sum_{t \in T} (t + \Delta t_i - e_i) P_i x_{ijt} \quad (1)$$

(PASPPP) Sujeito a:

$$\sum_{j \in M} \sum_{t \in T} x_{ijt} = 1 \quad \forall i \in N \quad (2)$$

$$x_{ijt} = 0 \quad \forall i \in N; j \in M; t \in T / t < e_i \text{ or } t > l_i - \Delta t_i \quad (3)$$

$$\sum_{i \in N} x_{ijt} \leq 1 \quad \forall j \in M; t \in T \quad (4)$$

$$x_{ijt} + x_{i'jt'} \leq 1 \quad \forall i \in N; j \in M; t \in T; i' \in N / i' \neq i; t' \in T / t \leq t' \leq t + \Delta t_i \quad (5)$$

$$x_{ijt} \in \{0,1\} \quad \forall i \in N; j \in M; t \in T \quad (6)$$

Sendo $x_{ijt} = 1$ se o poço i receber a sonda j no instante t para início da intervenção, e $x_{ijt} = 0$ caso contrário. A função objetivo (1) indica que a perda total de produção deve ser minimizada. As restrições (2) garantem que cada poço i é servido apenas uma única vez por qualquer sonda j durante o horizonte de planejamento hp . As restrições (3) asseguram que nenhum poço i será atendido antes do instante e_i ou após o instante $l_i - \Delta t_i$. As restrições (4)

certificam que cada sonda j , em um determinado instante t , estará executando intervenção em no máximo um poço i . As restrições (5) garantem que, se em um instante t uma sonda j iniciar o serviço em um determinado poço i , essa sonda ficará indisponível para iniciar serviço em outros poços até o instante $[t + \Delta t_j]$. As restrições (6) indicam que todas as variáveis são binárias. Observe que nesse modelo qualquer sonda pode realizar qualquer tipo de intervenção, conforme Costa e Ferreira Filho (2004). O conjunto de restrições definido em (5) cresce exponencialmente com o tamanho da instância, inviabilizando assim a aplicação direta de *solvers* comerciais de otimização.

4. GRASP + Path-Relinking

GRASP (Feo e Resende, 1989) e *Path-Relinking* (Glover, 1996) são algoritmos bem conhecidos e muito utilizados para solucionar problemas de otimização combinatória. Cada iteração do GRASP é composta por uma fase construtiva, na qual uma solução factível é construída, e por uma fase de busca local que, baseada na solução construída, aplica movimentos iterativos (melhorias) até que o ótimo local seja encontrado. Glover (1996) propôs o PR como uma estratégia de intensificação, explorando as trajetórias que conectam as soluções elite obtidas por métodos como Busca Tabu e *Scatter Search* (Glover *et al.*, 2000). Laguna e Martí (1999) utilizaram o GRASP+PR para intensificar a busca por melhores soluções. No caso do GRASP, a reconexão (*relinking*) consiste em, por exemplo, encontrar o caminho entre a solução encontrada por uma iteração e uma ou mais soluções elite selecionadas durante o processo de busca.

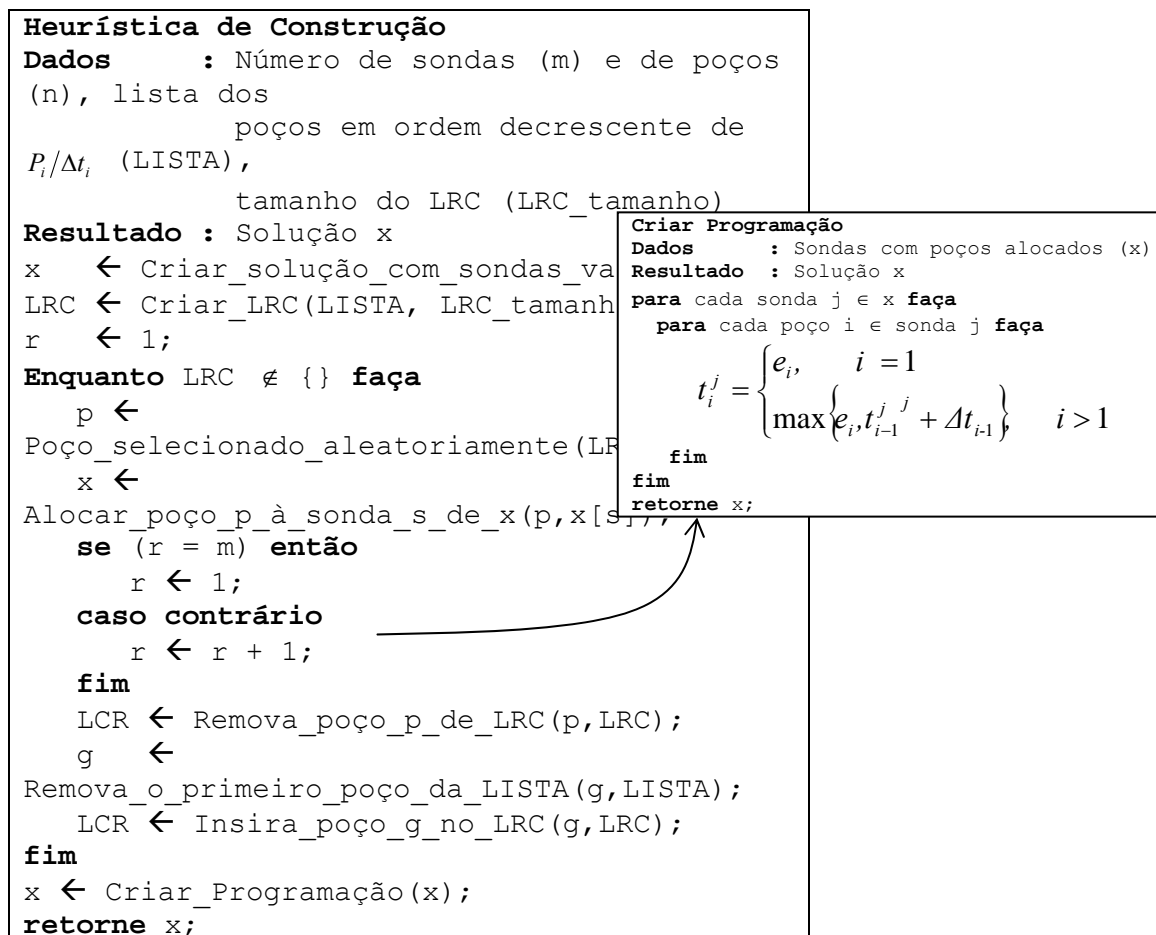


Figura 1 - Algoritmo da fase de construção.

FASE DE CONSTRUÇÃO

O procedimento de construção utilizado para o PASPPP cria, inicialmente, m sondas “vazias”, isto é, sem nenhum poço associado (Figura 1). Os poços estão ordenados em ordem decrescente da razão $P_i/\Delta t_i$. Os primeiros “LRC_tamanho” poços compõem a Lista Restrita de Candidatos (LRC). Dessa lista, um poço é aleatoriamente escolhido e alocado a uma sonda. As sondas são selecionadas em sequência, ou seja, o primeiro poço é alocado à primeira sonda, o segundo poço à segunda sonda, até a última sonda. O próximo poço será alocado à primeira sonda, o seguinte à segunda, e assim por diante. Esse procedimento de construção não assegura que as restrições de janela de tempo ou de horizonte de planejamento sejam atendidas. Após a execução do algoritmo de construção, o algoritmo de criação da programação (Figura 1) é usado para definir o instante em que as intervenções serão iniciadas.

Embora a fase de construção tradicional do GRASP considere apenas soluções viáveis, neste caso são permitidas soluções com violações, pois soluções infactíveis podem ser interessantes na busca por boas soluções factíveis de problemas muito complexos. Porém, faz-se necessário penalizar tais soluções infactíveis para que não sejam aceitas durante o processo de busca. Essa é a primeira diferença em relação ao GRASP proposto por Costa (2005).

$$f(x) = w_1 \sum_{i \in N} (t_i^j P_i) + w_2 \sum_{i \in N} (\max(0, t_i^j + \Delta t_i - hp)) + w_3 \sum_{i \in N} (\max(0, e_i - t_i^j) + \max(0, t_i^j - l_i)) \quad (7)$$

Uma solução x é avaliada pela Equação (7), que possui os pesos w_1 , w_2 e w_3 . O primeiro termo na Equação (7) indica a perda total de produção de todos os poços na solução x , o segundo corresponde ao tempo total além do horizonte de planejamento, e o terceiro calcula o tempo total extrapolando as janelas de tempo.

Path-Relinking	GRASP+PR
<p>Dados : Solução atual x, melhor solução encontrada x_t</p> <p>Resultado : A melhor solução x^* no caminho de x até x_t</p> <p>$x^* \leftarrow x$;</p> <p>for $p=1$ até n faça</p> <p style="padding-left: 20px;">$i \leftarrow$</p> <p>Encontre_sonda_com_o_poço_p_em(x);</p> <p style="padding-left: 20px;">$j \leftarrow$</p> <p>Encontre_sonda_com_o_poço_p_em(x_t);</p> <p style="padding-left: 20px;">se $i \neq j$ então</p> <p style="padding-left: 40px;">Remova_poço_p_da_sonda_i(x);</p> <p style="padding-left: 40px;">Insira_poço_p_na_sonda_j(x);</p> <p style="padding-left: 40px;">se $f(x) < f(x^*)$ então</p> <p style="padding-left: 60px;">$x^* \leftarrow x$;</p> <p style="padding-left: 40px;">fim</p> <p style="padding-left: 20px;">se $x = x_t$ então</p> <p style="padding-left: 40px;">retorne x^*;</p> <p style="padding-left: 20px;">fim</p> <p style="padding-left: 20px;">fim</p> <p>fim</p> <p>se $f(x^*) < f(x_t)$ então</p> <p style="padding-left: 20px;">retorne x^*;</p> <p>caso contrário</p> <p style="padding-left: 20px;">retorne x_t;</p> <p>fim</p>	<p>Dados : Número de sondas (m), poços (n), número máximo de iterações (I_{max}), número máximo de iterações da busca local (BL_{max}), tamanho da lista restrita de candidatos ($LRC_{tamanho}$), número de iterações para executar o path-relinking (PR_{iter})</p> <p>Resultado : Melhor solução x^*</p> <p>$f^* \leftarrow \infty$;</p> <p>$r \leftarrow 0$;</p> <p>para $i=1$ até I_{max} faça</p> <p style="padding-left: 20px;">$x \leftarrow$</p> <p>Heurística_de_Construção($m, n, LRC_{tamanho}$);</p> <p style="padding-left: 20px;">$x \leftarrow$</p> <p>Busca_Local(x, BL_{max});</p> <p style="padding-left: 20px;">se $f(x) < f^*$ então</p> <p style="padding-left: 40px;">$f^* \leftarrow f(x)$;</p> <p style="padding-left: 40px;">$x^* \leftarrow x$;</p>

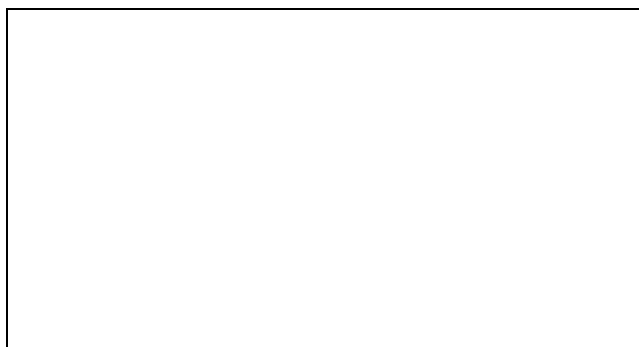


Figura 2 - Path-Relinking

```

fim
se (r = PR_iter) então
    x* ← Path-
    Relinking(x, x*);
    f* ← f(x*);
    r ← 0;
fim
    r ← r + 1;
fim
retorne x*;

```

Figura 3 - Algoritmo do GRASP+PR

BUSCA LOCAL

A construção da busca local proposta neste trabalho é baseada nos movimentos propostos por Aloise *et al.* (2006). No primeiro, chamado de Remove e Adiciona (RA), um poço é aleatoriamente escolhido de uma sonda e adicionado em outra. No segundo movimento, chamado de Troca de Poços de Diferentes Sondas (TPDS), dois poços são aleatoriamente escolhidos de sondas diferentes e trocados. A seleção das sondas utilizadas é sempre aleatória.

Dada uma solução x proveniente da fase de construção, em cada iteração da busca local é executado um desses movimentos (RA ou TPDS), e a nova solução é avaliada em seguida. Após BL_max iterações, a melhor solução, de acordo com a Equação (7), é retornada. Essa é a segunda diferença em relação ao GRASP proposto por Costa (2005).

PATH-RELINKING

O GRASP+PR proposto considera apenas uma solução elite x_i . Após $prIter$ iterações da fase de construção e do procedimento de busca local, a solução atual x é progressivamente transformada na solução elite x_i , por meio de um conjunto de movimentos definidos pela diferença $\Delta(x, x_i)$, i.e., um conjunto de movimentos necessários para alcançar x_i partindo de x (Glover *et al.*, 2000). Um grupo de soluções é gerado quando se caminha de x até x_i e a melhor delas é retornada (Figura 2). Experimentos computacionais mostram que esse método gera boas soluções, e assim, o uso do PR no GRASP se constitui como a terceira diferença em relação ao GRASP de Costa (2005). Considerando o algoritmo de construção, a busca local e o *Path-Relinking*, tem-se o algoritmo GRASP+PR proposto para o PASPPP (Figura 3).

5. Resultados Computacionais

O GRASP+PR foi implementado em C++ e executado em um computador com um processador AMD-Atlhon de 2.2 GHz e 1GB de memória RAM. Os testes computacionais foram realizados utilizando-se o grupo A das instâncias propostas por Costa (2005). Esses problemas são compostos por 25, 50, 75, 100 e 125 poços, com 2, 4, 6, 8 e 10 sondas.

Os parâmetros do GRASP+PR, em todos os testes, foram definidos empiricamente: $I_max = 20$, $BL_max = 40000$, $PR_Iter = 2$ e $LRC_tamanho = m$ (número de poços). Os pesos usados na equação (7) foram $[w_1, w_2, w_3] = [1, 100, 100]$. A Tabela 1 apresenta os resultados encontrados, sendo que, nas colunas do GRASP+PR, v_{melhor} representa o valor da função objetivo da melhor solução encontrada em dez experimentos; $v_{médio}$ é a média aritmética dessas dez execuções; $Desvio$ apresenta o desvio (Equação 8) obtido nos testes; e $Tempo$ (s) indica o tempo médio de execução, em segundos. As duas últimas colunas da Tabela 1 exibem os respectivos resultados do mesmo GRASP aqui proposto sem a utilização do *Path-Relinking*. Cabe ainda ressaltar que o GRASP+PR apresentou soluções factíveis para todos os testes com pequenos desvios, mostrando assim sua robustez.

$$Desvio (\%) = \left(\frac{v_{m\u00e9dio} - v_{melhor}}{v_{melhor}} \right) \times 100 \quad (8)$$

O CPLEX 12.1 (IBM, 2009) foi utilizado para tentar resolver as inst\u00e2ncias de forma \u00f3tima. Por\u00e9m, ele obteve \u00eaxito apenas nas inst\u00e2ncias com at\u00e9 50 po\u00e7os. Para inst\u00e2ncias com 75, 100 e 125 po\u00e7os o CPLEX n\u00e3o iniciou o processo de otimiza\u00e7\u00e3o, parando por falta de mem\u00f3ria. No pior caso, o modelo matem\u00e1tico gerou aproximadamente 50 milh\u00f5es de restri\u00e7\u00f5es (5).

A Tabela 1 apresenta ainda uma compara\u00e7\u00e3o entre o GRASP+PR proposto neste trabalho e outros cinco m\u00e9todos recentes encontrados na literatura. \u00c9 v\u00e1lido ressaltar que o GRASP+PR gerou boas solu\u00e7\u00f5es (algumas \u00f3timas - veja asteriscos). Por exemplo, a melhor solu\u00e7\u00e3o conhecida at\u00e9 ent\u00e3o para a inst\u00e2ncia 125-10 era igual a 93078 (Douro e Lorenzoni, 2009) e, agora, o GRASP+PR forneceu 93067. De modo geral, o GRASP+PR gerou solu\u00e7\u00f5es melhores do que o AG com trocas 2-Opt, encontrando 22 melhores solu\u00e7\u00f5es contra apenas 6 do AG-2-Opt.

6. Conclus\u00f5es

Este artigo apresentou um GRASP+*Path-Relinking* (PR) para o Problema de Aloca\u00e7\u00e3o de Sondas de Produ\u00e7\u00e3o a Po\u00e7os de Petr\u00f3leo. O GRASP+PR possui novas caracter\u00edsticas quando comparado com o GRASP desenvolvido por Costa (2005), e tamb\u00e9m gera melhores solu\u00e7\u00f5es para as inst\u00e2ncias criadas a partir de dados reais de um campo de petr\u00f3leo localizado no norte do estado do Esp\u00edrito Santo.

As solu\u00e7\u00f5es obtidas com o GRASP+PR foram melhores do que as solu\u00e7\u00f5es obtidas por outros m\u00e9todos recentemente publicados na literatura. O algoritmo desenvolvido se mostra robusto e apresenta um baixo desvio, utilizando, no pior caso, menos de 26 segundos para sua execu\u00e7\u00e3o. Resultados computacionais demonstram o potencial do GRASP+PR para solu\u00e7\u00e3o de problemas encontrados no crescente e promissor setor de petr\u00f3leo e g\u00e1s brasileiro.

Tabela 1 - Comparação com a literatura

Inst.	CPLEX 12.1		HMD (Costa e Ferreira)		GRASP (Costa, 2005)		Scatter Search (Oliveira)		Bubble Swap (Pacheco <i>et</i>		GA 2-Opt (Douro e Lorenzoni,		GRASP+PR				GRASP	
	V_{melhor}	T (s)	V_{melhor}	T	V_{melhor}	T (s)	V_{melhor}	T	V_{melhor}	T	V_{melhor}	T (s)	V_{melhor}	$V_{\text{média}}$	Desvio	T(s)	V_{melhor}	T (s)
A25-2	16329*	5,27	16421	---	17737	84,00	16338	---	16332	0,04	16329*	4,00	16329*	16329,60	0,004	6,86	16329	6,73
A25-4	10312*	4,42	10348	---	10825	88,00	10358	---	10338	0,04	10312*	5,00	10312*	10312,00	0,000	8,64	10314	6,09
A25-6	8497*	6,02	8555	---	8897	88,00	8561	---	8525	0,04	8499	2,00	8497*	8497,20	0,002	8,43	8497	6,88
A25-8	7733*	6,62	7735	---	7939	87,00	7772	---	7735	0,04	7736	1,00	7733*	7733,00	0,000	10,18	7733	9,03
A25-10	7322*	7,97	7329	---	7470	89,00	7352	---	7325	0,04	7325	1,00	7322*	7322,00	0,000	14,80	7322	8,97
A50-2	66904*	30,28	66920	---	81986	1176,0	67834	---	66926	0,19	66907	14,00	66904*	66906,00	0,003	11,40	66908	10,58
A50-4	37891*	40,64	37936	---	43451	1224,00	38232	---	37908	0,29	37896	15,00	37894	37896,30	0,006	9,34	37896	8,75
A50-6	28346*	54,47	28485	---	33635	1143,00	28674	---	28420	0,29	28353	5,00	28348	28350,70	0,010	10,45	28352	9,70
A50-8	23788*	72,50	23839	---	26910	1200,00	24083	---	23810	0,14	23788*	16,00	23789	23794,70	0,024	11,25	23795	10,23
A50-10	21348*	85,62	21409	---	23770	1207,00	21488	---	21390	0,14	21351	10,00	21350	21351,80	0,008	13,06	21351	12,69
A75-2	---	---	187358	---	239459	5030,00	192536	---	187339	0,23	187240	36,00	187240	187247,30	0,004	15,92	187264	15,39
A75-4	---	---	103364	---	137715	4435,00	106534	---	103301	0,32	103218	27,00	103212	103220,90	0,009	13,85	103225	13,05
A75-6	---	---	75871	---	97572	5444,00	77714	---	75602	0,37	75524	25,00	75500	75514,40	0,019	14,10	75516	13,33
A75-8	---	---	62179	---	78165	4964,00	63600	---	62048	0,51	61916	18,00	61886	61895,20	0,015	14,92	61890	12,84
A75-10	---	---	54099	---	66270	5161,00	55171	---	53956	0,56	53889	29,00	53883	53888,10	0,009	14,27	53888	13,75
A100-2	---	---	299093	---	405969	11377,00	319162	---	299105	0,37	299051	61,00	299049	299051,40	0,001	21,01	299063	20,88
A100-4	---	---	160016	---	209626	10786,00	169671	---	160007	0,56	159983	48,00	159968	159971,90	0,002	16,12	159969	15,47
A100-6	---	---	114456	---	148148	10785,00	120377	---	114372	0,65	114275	136,00	114285	114302,80	0,016	14,56	114317	14,55
A100-8	---	---	91954	---	115852	11101,00	96736	---	91875	0,70	91769	141,00	91779	91787,80	0,010	16,87	91790	15,39
A100-10	---	---	78541	---	96243	10918,00	83552	---	78478	0,79	78402	58,00	78394	78404,80	0,014	17,66	78397	16,84
A125-2	---	---	380631	---	534998	20287,00	413843	---	380674	0,65	380523	146,00	380521	380526,60	0,001	25,79	380537	25,06
A125-4	---	---	200408	---	265171	20232,00	218513	---	200471	0,89	200368	60,00	200325	200341,60	0,008	18,15	200334	17,06
A125-6	---	---	140648	---	188746	20050,00	152420	---	140634	1,08	140550	84,00	140533	140552,80	0,014	16,40	140555	15,97
A125-8	---	---	111015	---	145440	20269,00	119365	---	110980	1,31	110844	66,00	110816	110837,40	0,019	18,32	110866	17,83
A125-10	---	---	93280	---	122438	20888,00	103265	---	93159	1,40	93078	164,00	93067	93090,30	0,025	20,53	93086	19,08

Por terem sido utilizados diferentes computadores, não foi possível fazer uma comparação entre os tempos de execução.

Computadores utilizados:

Heurística de Montagem Dinâmica (HMD) e GRASP (Costa, 2005) – Processador Pentium IV com 1GB de memória RAM.

Scatter Search (Oliveira *et al.*, 2007) – Processador Athlon 2.2 Ghz com 512 MB de memória RAM.

Bubble Swap (Pacheco *et al.*, 2009) – Processador Pentium Core 2 Duo 1,8GHz com 1 GB de memória RAM.

Algoritmo Genético com 2-Opt (Douro e Lorenzoni, 2009) – Processador Pentium IV 2 GHz com 480 MB de memória RAM.

Agradecimentos: Os autores agradecem à FAPES (processo 45391998/09) e ao CNPq (processo 201509/2009-1) pelo apoio financeiro.

Referências

- Aloise, D. J., Aloise, D., Rocha, C. T. M., Ribeiro, C. C., Ribeiro Filho, J. C. e Moura, L. S. S.** (2006). Scheduling workover rigs for onshore oil production. *Discrete Applied Mathematics*, 154(5), 695-702.
- Alves, V. R. F. M. e Ferreira Filho, V. J. M.** (2006). Proposta de algoritmo genético para a solução do problema de roteamento e sequenciamento de sondas de manutenção. *Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional*, 1837-1848, CD-ROM. Goiás, Brasil.
- Costa, L. R.** (2005). Soluções para o problema de otimização de itinerário de sondas. *Dissertação de Mestrado*, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- Costa, L. R. e Ferreira Filho, V. J. M.** (2004). Uma heurística para o problema do planejamento de itinerários de sondas em intervenções de poços de petróleo. *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*, 1844-1853, CD-ROM. Minas Gerais, Brasil.
- Costa, L. R. e Ferreira Filho, V. J. M.** (2005). Uma heurística de montagem dinâmica para o problema de otimização de itinerários de sondas. *Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional*, 2176-2187, CD-ROM. Rio Grande do Sul, Brasil.
- Douro, R. F. e Lorenzoni, L. L.** (2009). Um algoritmo genético-2opt aplicado ao problema de otimização de itinerário de sondas de produção terrestre. *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional*, 2121-2132, CD-ROM. Bahia, Brasil.
- Feo, T. A. e Resende, M. G. C.** (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8, 67-71.
- Glover, F.**, Tabu search and adaptive memory programming – Advances, applications and challenges. In: R. S. Barr, R. V. Helgason e J. L. Kennington (Eds), *Interfaces in Computer Science and Operations Research*. Kluwer Academic Publishers, 1-75, (1996).
- Glover, F., Laguna, M. e Martí, R.** (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39, 653-684.
- IBM** (2009). IBM ILOG CPLEX v12.1: User's manual for CPLEX, 952p.
- Laguna, M. e Martí, R.** (1999). GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11, 44-52.
- Neves, T. A.** (2007). Heurísticas com memória adaptativa aplicadas ao problema de roteamento e scheduling de sondas de manutenção. *Dissertação de Mestrado*, Universidade Federal Fluminense, Rio de Janeiro, Brasil.
- Noronha, T. F., Lima, F. C. J., e Aloise, D. J.** (2001). Um algoritmo heurístico guloso aplicado ao problema do gerenciamento das intervenções em poços petrolíferos por sondas de produção terrestre. *Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional*, São Paulo, Brasil.
- Oliveira, E. F., Pagoto, F. B., Silva, F. T., e Lorenzoni, L. L.** (2007). Scatter search aplicado ao problema de otimização da alocação de sondas de produção em poços de petróleo. *Anais do XXVII Encontro Nacional de Engenharia de Produção*, 1-10, Paraná, Brasil.
- Pacheco, A. V. F., Dias Filho, A. C. T. e Ribeiro, G. M.** (2009). Uma heurística para o problema da alocação de sondas de produção em poços de petróleo. *Anais do XXIX Encontro Nacional de Engenharia de Produção*, Bahia, Brasil.
- Smith, W. E.** (1956). Various optimizers for single stage production. *Naval Research Logistics Quarterly*, 3, 59-66.