



ALGORITMOS EVOLUCIONÁRIOS NA SOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO DO EMPREGO DE SONDAS DE PRODUÇÃO EM POÇOS DE PETRÓLEO

Elizabeth Ferreira Gouvêa*

bgouvea@ufrnet.br

Marco César Goldberg*

gold@dimap.ufrn.br

Wagner Emanuel Costa*

wmano@dimap.ufrn.br

*Universidade Federal do Rio Grande do Norte – DIMAp – Campus Universitário, Natal, 59078-900

Resumo. O presente trabalho apresenta uma abordagem evolucionária para a solução do problema de programação de operações de sondas de produção em um campo de petróleo. Este artigo relata um experimento computacional e compara os resultados da aplicação de um algoritmo memético e de um transgenético para o problema.

Palavras Chave: Metaheurísticas, Algoritmos Evolucionários, Técnicas de Otimização.

Abstract. This work presents an evolutionary approach to solve the problem of scheduling the operation of workover rigs in petroleum fields. This paper reports the computational experiments and compares the results of applying a memetic and a transgenetic algorithm to the workover rigs problem.

Keywords: Metaheuristics, Evolutionary Algorithms, Optimization Techniques.

1. Introdução

Para que um poço num campo petrolífero opere de forma econômica e segura é necessário realizar operações de equipagem, ou ainda injetar fluidos nos reservatórios. Este conjunto de operações é denominado de completação. Mesmo em um poço adequadamente equipado e, em funcionamento produtivo, são indispensáveis outras intervenções de manutenção genericamente denominadas de workover. Em Thomas (2001) há uma descrição mais aprofundada das operações e do contexto do trabalho das sondas de produção. Para a realização destas operações utiliza-se um equipamento especializado denominado Sonda de Produção. Os custos das operações de completação e de workover são elevados, bem como as perdas com a não produção de poços aguardando serviços de sondas de produção. O atual trabalho reporta a descrição desse problema através de um modelo da família Scheduling e sua solução através da aplicação de um Algoritmo Memético e um Algoritmo Proto-Gene.

2. Definição do Problema

O problema da *Otimização do Emprego de Sondas de Produção em Poços de Petróleo (OSOP)* consiste em estabelecer uma programação de trabalho para a operação de uma frota de sondas sobre um conjunto de n poços de petróleo. Os custos envolvidos no roteiro de intervenções de uma sonda incluem custos de montagem e operação que são funções do tempo e da distância entre os poços. Existe também um custo associado ao atraso na produção dos poços que permanecem com a produção parada aguardando o atendimento de uma sonda. O custo e o tempo de cada serviço pode



variar em função tanto do tipo de sonda empregada como da própria operação a ser realizada. Existe a possibilidade de que a frota de sondas seja reforçada com sondas alugadas, se necessário. O afretamento de sondas muitas vezes é acertado por contratos que definem um número de sondas que ficam constantemente no campo e um conjunto de sondas reservas que podem vir a atuar no campo. As sondas em reserva via de regra possuem um custo mais elevado por hora trabalhada. Os tempos das intervenções são significativamente maiores que os tempos necessários para o deslocamento das sondas entre os poços, principalmente no ambiente marítimo.

A literatura destaca principalmente duas abordagens para o cálculo dos custos das sondas de produção. A primeira, descrita em Paiva (1997), avalia a perda de poços considerando um reajuste na pressão do reservatório que poderá resultar em um retorno do poço com uma produção suplementar temporária. Esse fenômeno faz com que as perdas reais sejam menores do que a estimada ao se considerar o volume não produzido como sendo a vazão do poço antes do fechamento multiplicada pelo tempo sem produzir do poço. A perda calculada sem a consideração do efeito de vazão suplementar é denominada de perda aparente. Quanto maior o número de poços atuando no reservatório maior será a distancia da perda real para a perda aparente de volume.

A segunda abordagem, descrita em Chen (2001), admite a ocorrência de produção suplementar nos poços mas dependendo do risco associado ao fato de que a intervenção a ser realizada pode não atingir completamente seus objetivos. Portanto uma alternativa para levar em conta essas incertezas é calcular um coeficiente de custo composto para a intervenção em cada poço, levando em conta as prováveis perdas de produção e os custos da sonda com base no histórico de produção do poço.

O (OSOP) é um problema que envolve o atendimento de tarefas que deverão ser executadas em poços demandantes através de equipamentos móveis que se deslocam entre esses poços. Nesse sentido o problema poderia ser pensado tanto como um modelo da classe dos casos de roteamento e scheduling ou como um problema de scheduling. A primeira abordagem compreenderia (OSOP) segundo um modelo em rede de demandas. Em um caso on-line (OSOP) poderia também ser modelado como um problema de análise competitiva (Manasse e McGeoch, 1990) correspondendo a deslocar k-servos – as sondas de produção – sobre uma rede de poços que continuamente demandariam trabalho segundo certas janelas de tempo (Oliveira, 1996).

Como os tempos necessários para movimentar as sondas de produção entre os poços de um campo são da ordem de minutos ou no máximo horas, e o tempo de montagem e desmontagem da sonda nos poços, somado ao tempo característico de intervenção – o tempo acumulado de visita – é da ordem de dias ou semanas (Paiva, 1997 e D'Almeida, 1996), o tempo de visita é um fator claramente preponderante. Nessas condições Kizilisik (1999) sugere o uso de modelo de scheduling.

3. Algoritmo Memético para a Solução do (OSOP)

Os Algoritmos Genéticos (AGs) foram introduzidos por Holland (1975) e sua metáfora é baseada nos princípios evolucionários de Darwin. Os AGs foram bem sucedidos na obtenção de soluções heurísticas no contexto da solução de problemas NP-Árduos (Goldberg, 1989). Recentes trabalhos têm reportado excelentes resultados com a incorporação de procedimentos de busca local aos AGs (Merz e Freisleben, 2000), de modo que para a solução de (OSOP) a atual pesquisa optou por desenvolver um algoritmo memético – Mem_SOP – com as seguintes características:

Codificação: Os cromossomos de Mem_SOP são constituídos por poços e sondas da forma sugerida pela figura 1.

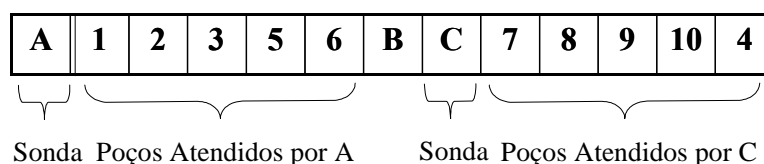




Figura 1: Cromossomo Utilizado nos Algoritmos Evolucionários

Uma solução codificada será viável se o primeiro gene for uma sonda e nenhum poço se repetir.

Função Adequação: a função adequação é baseada no tempo estimado de duração dos serviços, bem como a perda de produção decorrente da paralisação do poço, segundo a seguinte expressão:

$$f = \sum_{i=1}^m \sum_{j \in W_i} C_i Ta_j + \alpha_j Te_j \quad i = 1, \dots, m$$

onde m é o número de sondas, C_i é o custo de operação da sonda i , Ta_j é a duração da intervenção da sonda j , α_j é a perda de produção do poço j , Te_j é o tempo estimado para a interrupção. Observe-se que a perda real de produção de um poço depende de diversas variáveis. Chen (2001) apresenta uma pesquisa desses parâmetros e variáveis e define um valor de atratividade para a perda de produção de um poço. No presente trabalho a perda é calculada como uma função de retorno pós-intervenção em m^3/dia de óleo. No experimento computacional esse parâmetro é selecionado aleatoriamente em um intervalo [10,100].

Cruzamento: Mem_SOP utiliza o operador definido por Shi *et al.* (1997) por sua eficiência na solução de problemas de seqüenciamento de tarefas da família Job Shop conforme trabalhos de Jain e Meeran (1998). O tamanho da população é constante, e no processo de cruzamento dois pais produzem dois filhos.

Seleção: Mem_SOP ordena a população de cromossomos pela adequação em ordem não decrescente, escolhendo sempre um pai por sorteio no primeiro terço da população e o segundo pai dentre os demais da população. Após o cruzamento permanecem na população os dois melhores indivíduos do conjunto de dois pais e dois filhos.

Mutação: O algoritmo utiliza os operadores de mutação sobre os filhos de cada geração. Para realizar a mutação, Mem_SOP emprega os operadores 2-troca e “backward insertion”. No operador de 2-troca, duas posições aleatórias, diferentes da primeira posição, são escolhidas para serem trocadas no cromossomo. O operador “backward insertion” seleciona um gene que não esteja na primeira ou na última posição e insere esse gene em uma posição, sorteada aleatoriamente, anterior a que ele ocupava. Os demais genes são movidos uma posição para frente.

A seção 5 deste trabalho, que faz referência aos experimentos computacionais, relata como foram ajustados os parâmetros do algoritmo Mem_SOP descrito no Quadro I.

Início

Gerar aleatoriamente a população de cromossomos

Repetir

Avaliar a população e guardar o indivíduo de melhor adequação

Selecionar pais para reprodução

Efetuar cruzamento

Efetuar mutações

Escolher cromossomos que retornam à população

Realizar busca local nos cromossomos escolhidos

Até que critério de parada seja atendido



Fim

Quadro I: Algoritmo Memético, Mem_SOP

4. Algoritmo Transgenético para a Solução do (OSOP)

A Transgenética Computacional – TC – é uma metaheurística que emprega a infiltração planejada de informações como uma ferramenta do aperfeiçoamento do processo evolucionário. A operacionalização da manipulação da TC é realizada através de agentes de informação especialmente criados para esse fim. A TC utiliza-se de várias fontes para obter as informações necessárias ao processo de manipulação, contudo o método de codificação dessas informações procura imitar o paradigma epigenético. As informações em formato não genético que são utilizadas para informar o processo evolucionário são aqui denominadas *memes*.

Paradigma Epigenético

A cultura, como a biologia, exhibe padrões de evolução – exploração adaptativa e transformação de um espaço de informações através de variação, seleção e transmissão (Boyd e Richerson, 1985). A dificuldade de considerar a epigenética no contexto da Computação Evolucionária é discutida por Chattoe (1998). Pesquisas mostram que gene e cultura são inerentemente ligados (Lumsden e Wilson, 1981). No entanto a evolução cultural exhibe padrões e regras de desenvolvimento diferentes da biológica. Os memes são unidades de informação que podem difundir-se por uma cultura, como os genes disseminam-se por um reservatório gênico (Wright, 2000 e Plotkin, 1996). Apesar de ambos os construtos – gene e meme – codificarem em última análise informações, o primeiro está direcionado ao processo de construção de fenótipos, enquanto o segundo objetiva posicioná-los dentro dos desafios de um certo meio ambiente.

A ligação entre genes e memes ocorre uma vez que os genes prescrevem as regras epigenéticas, as quais são regularidades da percepção sensorial e do desenvolvimento mental que animam e canalizam a aquisição da cultura. A cultura, por sua vez, ajuda a determinar quais dos genes prescritos sobrevivem e se multiplicam de uma geração à próxima. Novos genes bem-sucedidos alteram as regras epigenéticas da população. As regras epigenéticas alteradas mudam a direção e a eficácia dos canais de aquisição cultural realimentando o processo de co-evolução gene x meme (Wilson, 1998).

Memes e os genes criam comportamento (Lynch, 1998). Memes e os genes são também constituídos em padrões auto-replicantes. Na computação os primeiros trabalhos dentro desse tema simplesmente associaram memes às informações obtidas fora do fluxo extracelular. Nos algoritmos Meméticos (Merz e Freisleben, 2000; Radcliffe e Surry, 1994) o estágio epigenético é expresso, principalmente, através de um aperfeiçoamento dos cromossomos via procedimentos de busca local. Tal processo foi denominado, por alguns autores, de evolução Lamarckiana (Whitley *et al.*, 1994) por analogia a proposta de Jean-Baptiste Lamarck. A chamada evolução Lamarckiana, vista da ótica da TC, pode ser interpretada como uma forma de atuação dentro do fluxo intracelular visando produzir macro-mutações benéficas à adequação do cromossomo.

Para emprego no contexto computacional, contudo, o atual trabalho propõe que os memes sejam definidos em conformidade com a sugestão de Plotkin (1996) que os considera como informações codificadas e transmitidas através de meios não genéticos. Neste trabalho um meme é uma informação que se origina em um contexto específico, sendo utilizada para re-arranjar um cromossomo ou um bloco de genes. A memética na TC será utilizada no sentido de criar um processo contínuo de obtenção e infiltração de informações para o interior do contexto genético. Esse fluxo de informações, como exigido pela evolução cultural, será guiado pela aplicação de regras epigenéticas. Por outro lado, os memes poderão ser sujeitos a processos de competição e seleção pertinentes ao meio cultural. A informação memética no contexto computacional pode se originar de várias fontes, inclusive do planejamento intencional. O contato entre os agentes da TC e os cromossomos redundam em um processo de co-evolução cromossomos x agentes que é operacionalizado por mecanismos semelhantes aos utilizados pela transferência horizontal biológica – assimilações – e de



recombinações procarióticas. As técnicas de manipulação intencional, derivadas da Engenharia Genética, também são admissíveis no sentido de facilitar que a informação memética possa se compor com a informação genética.

Pontos Relevantes do Fluxo Intracelular

O fluxo intracelular engloba as informações que conseguem atravessar a barreira celular e aquelas que são originadas em seu interior. As mutações genéticas, por exemplo, são fenômenos do fluxo intracelular. Os algoritmos genéticos híbridos ao realizarem procedimentos de busca local visando a melhoria da adequação, provocam um efeito semelhante ao de uma macro-mutação, ou seja, poderiam ser imaginados como procedimentos que utilizam a via intracelular para informar o contexto genético. Segundo vários autores esse fluxo é fundamental para a emergência da inovação dentro do processo evolucionário, proporcionando o caminho para que saltos evolutivos ocorram (Margulis e Sagan, 1991; Morowitz, 1992).

A Engenharia Genética (EG) (Duddley, 1990) é uma parte da ciência que estuda os mecanismos intracelulares de interação genética, bem como seu impacto sobre os organismos. As técnicas da EG abreviam o processo natural de manipulação da informação genética promovendo, simultaneamente, um aumento de eficácia nos resultados. Para facilitar o processo de manipulação a EG tem empregado agentes naturais especializados em navegar no fluxo intracelular. Tais agentes são principalmente os plasmídeos bacteriais e os vírus. Plasmídeos são pequenos anéis de ADN encontrados em diversas bactérias. Os plasmídeos são uma forma rápida das bactérias trocarem informações e redesenharem, de modo guiado, seu código genético (Shapiro, 1988).

Fundamentos da Metáfora da Transgenética Computacional

A Transgenética Computacional é baseada em três idéias centrais: a primeira é infiltrar, de forma planejada, informações no processo evolucionário com o objetivo de informa-lo e guiá-lo. Nesse sentido imita a manipulação da Engenharia Genética. A segunda idéia é planejar o processo de infiltração de informações e de construção de agentes através do uso do paradigma epigenético. Nesse sentido incorpora a idéia de evolução cultural dos memes. Finalmente utiliza a recombinação procariótica como ferramenta complementar de preservação da informação genética. Nesse sentido minimiza os obstáculos que impedem a transformação da informação memética em genética.

Como a informação memética não possui acesso ao reservatório genético, a transformação da informação epigenética em informação genética se dá, na natureza, através de mecanismos bastante complexos. Na computação essa dificuldade inexistente. Os memes da TC podem ser obtidos de várias fontes como a própria população, procedimentos heurísticos, resultados teóricos para o problema ou do próprio projetista do algoritmo. Ainda cabe ressaltar que, além de co-evolucionário, o processo evolutivo da TC pode ser competitivo em vários níveis. O processo pode envolver competição tanto na obtenção de memes (nível do projeto dos agentes), como na escolha dos agentes a empregar (nível da seleção dos agentes) e no processo de transcrição (nível da interface meme x gene).

A figura 2 resume o caminho percorrido pelas informações meméticas desde sua obtenção até sua infiltração nos cromossomos. O meme pode ser carregado em um agente transgenético já traduzido em um formato genético – originando um bloco de montagem – ou como uma regra de manipulação genética. As regras aqui denominadas do “tipo 1” são as responsáveis por traduzir os memes para uma representação compatível com o método do agente transgenético. O método de atuação de um agente é definido pela aplicação de regras que são chamadas do “tipo 2”. A co-evolução gene x meme tem seu controle definido pelas regras denominadas do “tipo 3”.

O reservatório das informações meméticas constitui um banco de informações ou banco de memes. Esse reservatório cultural pode congrega informações originárias de qualquer resultado associado ao problema. Por exemplo, a árvore geradora mínima é um limite inferior para o problema do Caixeiro Viajante. Esse fato pode gerar regras epigenéticas tais como: “usar caminhos de uma árvore geradora mínima (AGM) para compor blocos de montagem”. Assim, um banco de memes pode conter uma série de seqüências de nós – minimais no sentido da conexão – originadas de caminhos



formados na AGM e deduzidas da aplicação da regra epigenética anteriormente exemplificada. Se os cromossomos são representados como permutações de vértices, então as seqüências de vértices obtidas pela aplicação da regra anterior terão uma correspondência com segmentos de ADN – estarão expressas em formato genético.

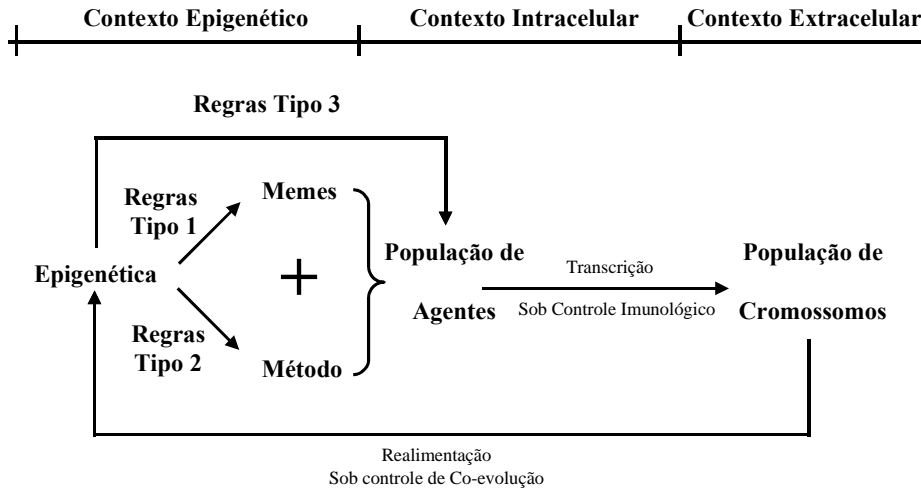


Figura 2: O Processo de Construção dos Agentes Transgenéticos

O Agente Partícula Genética Móvel

Os agentes transgenéticos são entidades lógicas que cumprem o papel de integrar as informações do formato memético com as codificadas no cromossomo em formato genético. Um agente transgenético é composto por um ou mais memes e um método de operação. Os agentes transgenéticos podem competir entre si ou com as informações que já estão incorporadas aos cromossomos alvo de manipulação. O algoritmo ProtoG do presente trabalho utiliza-se somente da Partícula Genética Móvel, PGM. Outros agentes transgenéticos são apresentados no trabalho de Goldberg e Gouvêa (2001).

As PGMs são fragmentos de ADN que podem ser fagocitados (incorporados) pelas células, acabando por se integrar ao ADN do núcleo. No contexto computacional uma PGM é um agente constituído por uma cadeia de informação (I) e um método de contaminação (C), formando uma dupla $PGM = (I, C)$. A cadeia de informação da PGM é formada por um fragmento de ADN – bloco de montagem. Um cromossomo pode resistir ao ataque de um agente. A regra de resistência imunológica ao ataque da PGM é a da estrita melhoria da adequação, ou seja, a transcrição da cadeia de informação da PGM em um cromossomo só ocorre quando redundante em melhoria para a adequação do cromossomo. A transcrição da informação da PGM se dá através da aplicação de um operador específico. O operador e os demais critérios que transformarão o cromossomo de modo a incorporar a informação da PGM constituem o método de contaminação desse agente. A figura 3 mostra um exemplo de um ataque de uma PGM que resulta em transcrição. A cadeia de informação transportada pode ser, por exemplo, constituída por poços cujo atendimento pode ser realizado seqüencialmente por uma certa sonda.

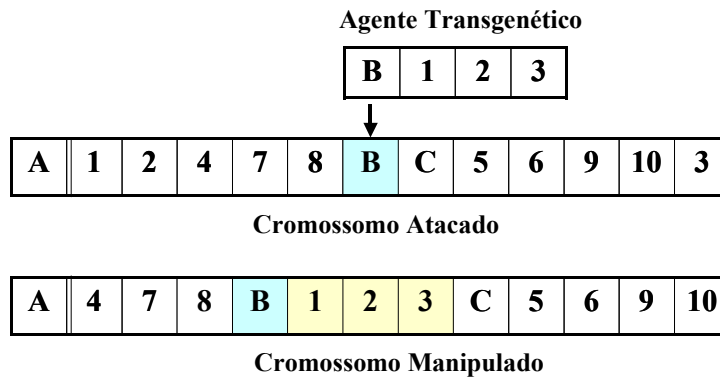
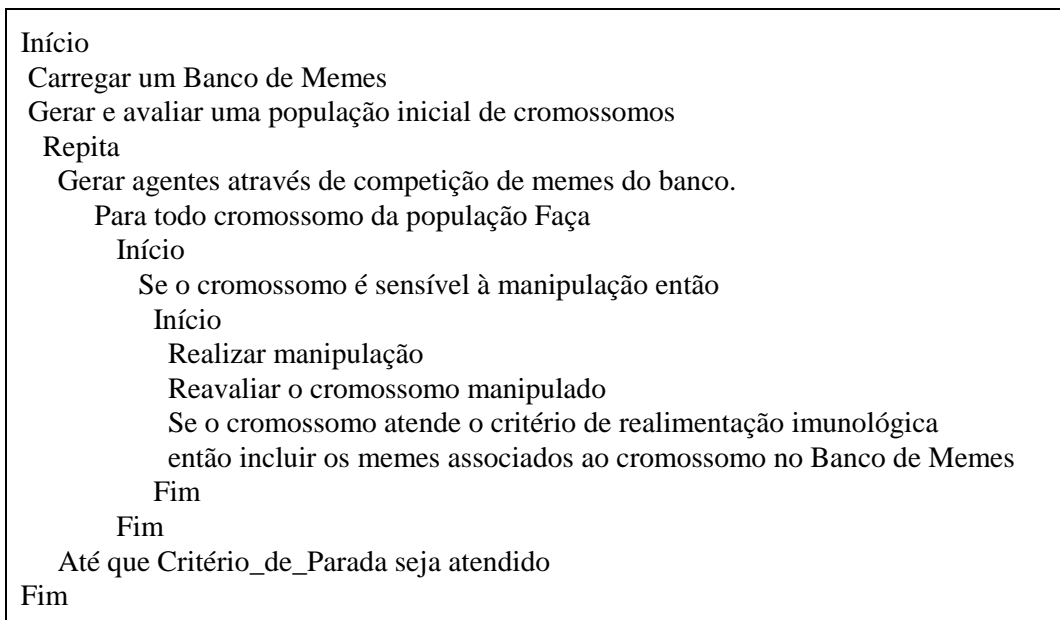


Figura 3: Um Processo de Manipulação Transgenética

Proto-Gene (ProtoG) – Algoritmo de Recombinação Procariótica

Os algoritmos da classe ProtoG atendem exclusivamente ao paradigma procariótico, ou seja, desenvolvem sua solução evolucionária partindo da suposição que o código genético se aperfeiçoa principalmente exposto a trocas de informações intencionalmente selecionadas. Esses algoritmos são constituídos em duas fases. A primeira caracteriza o projeto dos agentes e formaliza o processo de competição entre os memes para a constituição de agentes. Na segunda fase os agentes atacam os cromossomos da população procurando infiltrar suas informações. Assim sendo, a cada iteração do ProtoG, um ou mais agentes percorrem a população manipulando os cromossomos que se mostrarem capazes de aceitar incorporação dessa informação. No caso da emergência de uma informação de boa qualidade como um bloco de montagem promissor ou um cromossomo campeão, o algoritmo poderá incluir novos memes no reservatório de informação não genética através do exame da informação genética obtida. O Quadro II apresenta um algoritmo ProtoG.



Quadro II: Algoritmo Proto_G

5. Resultados Computacionais e Conclusões

O experimento computacional foi realizado com base em um banco de 27 instâncias geradas aleatoriamente e envolvendo sorteio dos seguintes fatores: número de poços, número de sondas fixas e



número de sondas alugadas. O custo unitário das sondas não alugadas foi considerado igual a 100 unidades monetárias e o custo das alugadas sorteado entre 100 e 200 unidades, de acordo com uma distribuição uniforme. A restrição de horizonte de tempo foi relaxada no experimento computacional. O tempo de disponibilidade inicial das sondas, alugadas ou não, foi sorteado entre 0 e 100 unidades de tempo. Os tempos de serviço foram sorteados entre 24 e 100 unidades de tempo.

5.1 Algoritmo Memético

Para ajustar os parâmetros a serem utilizados no Mem_SOP foram realizados testes piloto em instâncias com 20 poços. Os testes rodaram por 2000 iterações e compararam a influência do tamanho da população, da taxa de cruzamento e da taxa de mutação no desempenho qualitativo do algoritmo. Os resultados obtidos para essas instâncias foram estendidos ao restante do experimento.

Tamanho da população: foram testadas populações com 800, 900, 1000, 1200 e 1500 indivíduos. Em virtude do ensaio adotou-se a população com 1000 cromossomos, especialmente para o melhor o desempenho em tempo das iterações do algoritmo.

Taxa de Cruzamento: essa taxa é estimada em função do tamanho da população e representa a porcentagem de cruzamentos que serão realizados em cada iteração face ao tamanho da população. Shi *et al.* (1997) recomendam que essa taxa varie entre 60% e 100% da população. Nos experimentos teste, os melhores resultados foram obtidos para valores de 80%.

Taxa de Mutação: alguns trabalhos utilizam uma taxa variando entre 2 e 5% do tamanho da população para a mutação como, por exemplo no trabalho de Beasley (1993). Shi *et al.* (1997) recomendam uma taxa de mutação de 20% para problemas de job schedule, considerando o mesmo operador que é utilizado no presente trabalho. A melhor taxa de mutação segundo os testes foi encontrada para um valor de 30%.

Busca Local: a vizinhança da busca local é definida pela operação de 2-troca entre as posições 2 e p , onde p é o tamanho do cromossomo. O procedimento de melhoria iterativa realiza, no máximo, três iterações.

5.2 Algoritmo ProtoG

No algoritmo ProtoG apresentado neste trabalho a proporção entre o número de PGMs carregadas com memes do banco de memes e o número de PGMs carregadas com informações ambientais de diversificação foi objeto de um estudo piloto que levou a adoção de uma taxa de 2 ataques de agente de diversificação para cada 3 ataques de agentes carregados com memes do banco. Essa taxa é um fator importante para o equilíbrio entre os esforços de diversificação e intensificação do algoritmo. O agente ambiental, ou seja, o agente carregado com uma informação de origem aleatória, representa uma eficiente forma de manter a diversificação do processo de evolução procariótica.

A população do ProtoG foi adotada de igual tamanho à população do Mem_SOP para reduzir o número de variáveis livres do experimento. O banco de memes foi carregado inicialmente com uma solução advinda da aplicação da heurística WSPT (Rothkoph, 1966) e uma da aplicação da heurística LPT1 (Lee, 1966). Na medida do desenvolvimento do algoritmo o critério de realimentação imunológica determina a inclusão no banco de memes de toda solução campeã que seja constituída no processo. Em consequência da inclusão da melhor solução corrente deve deixar o banco a solução de pior adequação.

Para comparar o desempenho relativo dos algoritmos e otimizar o esforço do experimento computacional a regra de parada foi estabelecida por tempo computacional e igual a três minutos de processamento em um microcomputador Pentium IV 1.2 GHz. Como as figuras 4 e 5 mostram, essa regra de parada é bastante sensível as principais melhorias do processo evolucionário, especialmente em se tratando do algoritmo memético, denominado genericamente de “genético” no gráfico.

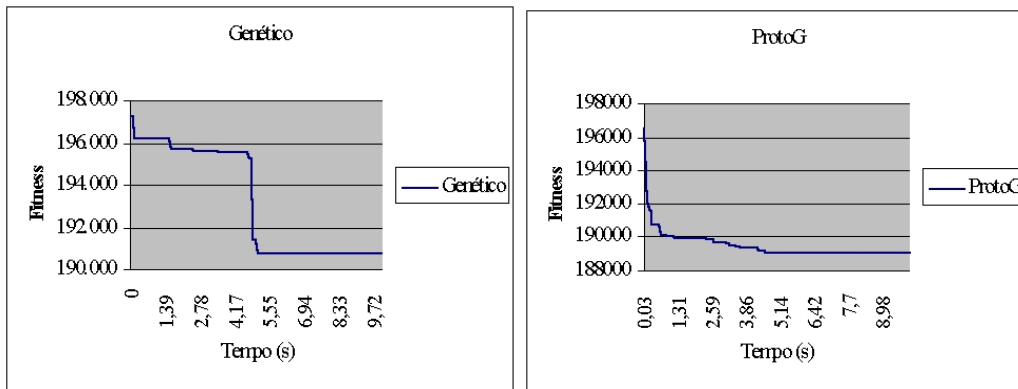


Figura 4: Desempenho na Instância REC01 – Horizonte de 10 Segundos

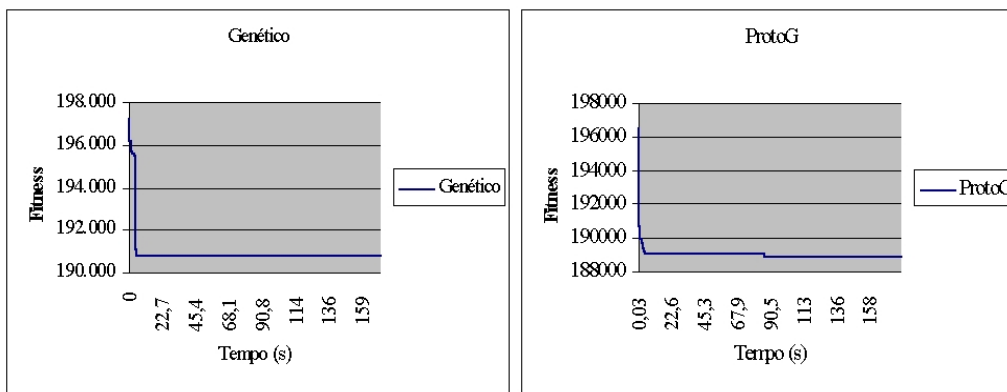
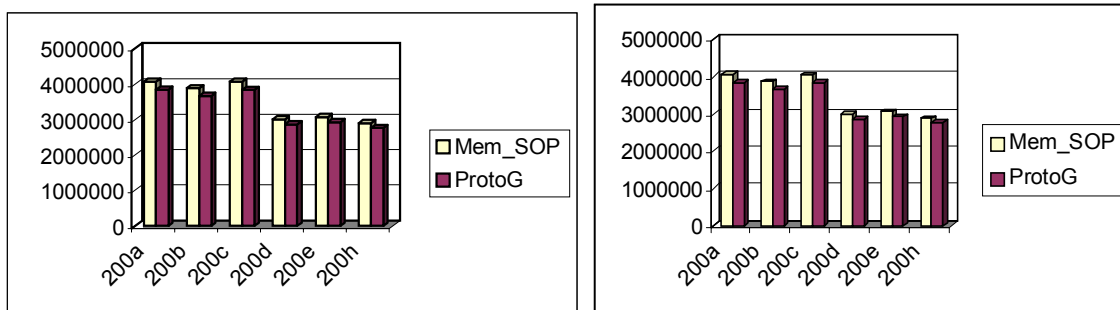


Figura 5: Desempenho na Instância REC01 – Horizonte de 180 Segundos

5.3 Experimento

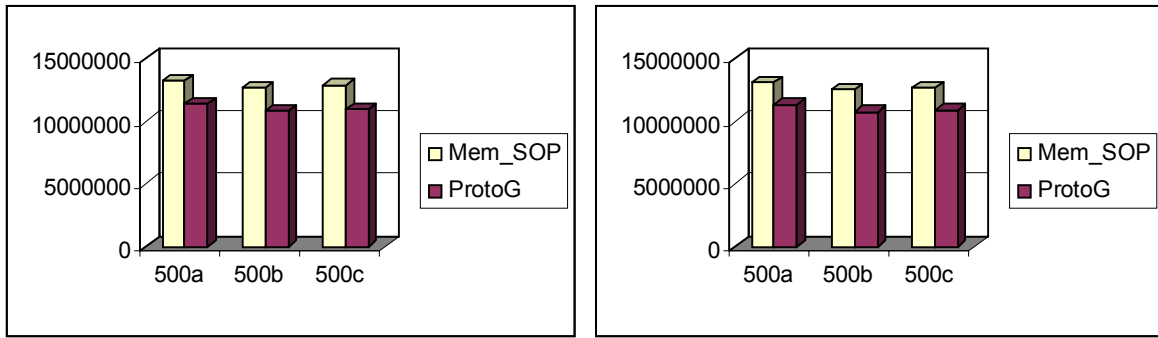
Os algoritmos ProtoG e Memético rodaram 30 vezes para cada instância durante três minutos em cada rodada, sendo os resultados resumidos na tabela I. Na coluna tipo consta o número de poços, o número de sondas da frota de atendimento e o número de sondas que podem ser alugadas em reforço.

Os gráficos da figura 6 e 7, cujas ordenadas representam os valores de fitness, mostram claramente a superioridade qualitativa da solução obtida pelo algoritmo ProtoG nesse experimento. O gráfico da figura 7 ressalta o fato de que os piores resultados das 30 rodadas ProtoG para as instâncias 200 e 500 são melhores que os melhores resultados das rodadas Mem_SOP.



Valores Mínimos - Instâncias 200 poços

Valores Médios - Instâncias 200 poços



Valores Mínimos - Instâncias 500 poços

Valores Médios - Instâncias 500 poços

Figura 6: Gráficos de Desempenho Relativo das Instâncias 200 e 500

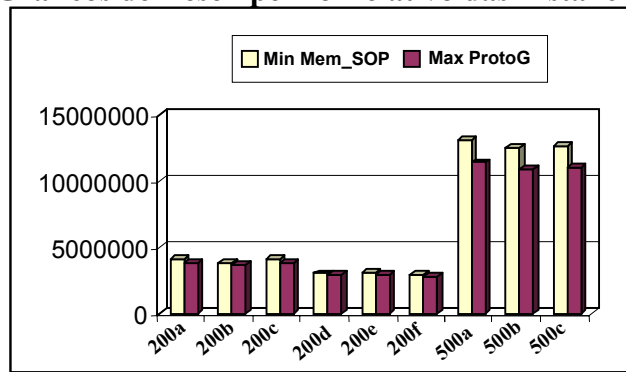


Figura 7: Gráfico de Desempenho Min x Max

Caso Teste	Tipo	Mem_SOP			ProtoG		
		Min	Média	Max	Min	Média	Max
rec01	20-2-3	188891	188891	188891	188884	188910,1	189091
rec03	20-2-3	159892	159892	159892	159881	159938,6	159983
rec05	20-2-3	159272	159272	159272	159249	159329,4	159336
rec07	20-2-8	148645	148645	148645	148586	148700	148973
rec09	20-3-7	199221	199221	199221	199191	199294,2	199373
rec11	20-8-2	143716	143716	143716	143708	143763,6	143828
rec37	75-4-16	768766	768789,7	769122	768715	769584,9	770618
rec39	75-2-18	893376	893542,8	893876	892961	894344,1	896297
rec41	75-11-9	797510	797605	797932	797326	798253,9	799289
050a	50-3-2	577689	577732,2	577787	577656	577858	578109
050b	50-4-1	690187	690279	690295	690167	690532,6	690893
050c	50-3-2	690356	690449,7	690587	690352	690754,7	691055
100a	100-6-4	1453706	1457594	1457702	1445080	1445785	1447000
100b	100-8-2	1275864	1277342	1279434	1269452	1270509	1272790
100c	100-9-1	1293391	1294525	1295629	1286148	1286736	1287816
100d	100-11-9	1217259	1219462	1221597	1211191	1213797	1218818
100e	100-11-9	1081539	1082825	1083919	1071401	1073764	1078306
100f	100-4-16	1299436	1301530	1302922	1290604	1292974	1296744
200a	200-6-4	4054425	4073714	4090028	3852468	3853995	3857171
200b	200-9-1	3864971	3888380	3910489	3675872	3677472	3679657
200c	200-5-5	4049651	4074927	4092817	3843305	3844851	3846969



200d	200-9-11	2993249	3014309	3025467	2867393	2872460	2875177
200e	200-6-14	3062105	3075650	3084940	2928537	2935511	2938231
200h	200-10-10	2885697	2904738	2919542	2762429	2767590	2771139
500a	500-3-17	13095483	13238926	13351544	11325860	11359092	11398609
500b	500-13-7	12549815	12638783	12712378	10761115	10808448	10934765
500c	500-7-13	12642354	12852138	12944094	10892088	10941988	10999748

Tabela I: Resultados do Experimento Computacional

É possível observar na coluna *Min* da Tabela 1 que o algoritmo ProtoG apresenta as melhores soluções para todas as instâncias do experimento. Além disso, à medida que o tamanho das instâncias cresce, as soluções obtidas pelo algoritmo ProtoG passam a apresentar uma melhoria bastante significativa em relação as soluções obtidas pelo outro algoritmo, chegando, essa diferença até 16,7%. Um comportamento semelhante pode ser observado para as demais colunas da tabela. Embora para instâncias pequenas, o algoritmo Mem_SOP apresente soluções médias e máximas, um pouco inferiores ao ProtoG, para instâncias maiores, o algoritmo transgenético obtém os melhores resultados chegando a uma diferença de 17% para as instâncias 500.

O experimento computacional realizado mostra que a confecção de um algoritmo evolucionário baseado na manipulação intencional e dirigida da informação, que não utiliza a reprodução sexual ou a busca local, apresenta um comportamento competitivo em relação ao algoritmo evolucionário híbrido desenvolvido segundo a abordagem dos algoritmos meméticos. Constata-se, igualmente, que a infiltração de boas informações obtidas através de meios não evolucionários pode acelerar e guiar o processo de evolução melhorando-o quantitativa e qualitativamente, e que a estratégia de transportar essas informações através de agentes semelhantes àqueles que operam no fluxo intracelular é operacionalmente viável. Conclui-se, ainda, que é possível manter a diversificação do processo evolucionário em um nível adequado, mesmo sem utilizar os recursos da mutação.

6. Agradecimentos

A presente pesquisa foi apoiada pelo programa de formação de recursos humanos, como um dos temas do projeto UFRN-PRH-22 – 2001, com participação da Agência Nacional de Petróleo.

7. Referências Bibliográficas

- BOYD, R., e P. RICHICHERSON, 1985, *Culture and Theory of Social Evolution*, Sage Publications, London.
- CHATTOE, E., 1998, Just How (Un)realistic are Evolutionary Algorithms as Representations of Social Processes?, *Journal of Artificial Societies and Social Simulation* 1 (3), disponível em: <http://www.soc.surrey.ac.uk/JASSS/1/3/2.html>.
- CHEN, S.-M., 2001, An Automatic Production Monitoring System - Design and Application, SPE72146, *Society of Petroleum Engineers*.
- COSTA, W. E., 2002, *Otimização das Sondas de Produção*, Monografia de Final do Curso de Engenharia de Computação, Universidade Federal do Rio Grande do Norte.
- D'ALMEIDA, A. L., 1996, Dimensionamento de Unidades de Serviço para Pescarias: Uma Aplicação de Pesquisa Operacional, *X Congresso Latino-Americano de Perforacion*.
- DUDLEY, W, 1990, *Genetic Engineering*, San Diego: Greenhave Press, Inc.
- GOLDBARG, M. C., e E. F. GOUVÊA, 2001, Extra-intracellular Transgenic Algorithm, In *Proceedings of GECCO 2001, Genetic and Evolutionary Computation Conference*, July, San Francisco, CA, USA.
- GOLDBERG, D., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley.
- JAIN, A., e S. MEERAN, 1998, *A State-of-the-art Review of Job-shop Scheduling Techniques*, <http://citeseer.nj.nec.com/jain98stateart.html>.



- KINZILISIK, Ö. B., 1999, *Predictive and Reactive Scheduling*. <http://benli.bcc.bilkent.edu.tr/~lors/>
- HOLLAND, J. H., 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.
- LEE, C. Y., 1996, Machine Scheduling with an Availability Constraint. *Journal of Global Optimization* 9, 395-416.
- LYNCH, A., 1998, Units, Events and Dynamics in Memetic Evolution, *Journal of Memetics - Evolutionary Models of Information Transmission*, 2.
- LUMSDEN, C. J., e O. E. WILSON, 1981, *Genes, Mind and Culture*, Harvard University Press, Cambridge.
- MANASSE, M. S. e L. A. McGEOCH, 1990, Competitive Algorithms for the Server Problem, *Journal of Algorithms* 11, 208-230.
- MARGULI, L., e D. SAGAN, 1991, *The Origins of Sex: Three Billions Years of Genetic Recombinations*, New Haven, Yale University Press.
- MERZ, P., e B. FREISLEBEN, 2000, Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem, *IEEE Transactions on Evolutionary Computation* 4 (4), 337-352. .
- MOROWITZ, H. J., 1992, *Beginning of Cellular Life*, New Haven, Conn, Yale University Press.
- OLIVEIRA, S. P., 1996, *O Problema dos K-Servos com Janela de Tempo*, Dissertação de Mestrado, Instituto Militar de Engenharia, IME, Rio de Janeiro.
- PAIVA, R. O., 1997, *Otimização do Itinerário de Sondas de Intervenção*, Dissertação de Mestrado, Universidade Estadual de Campinas, Brasil.

