

OTIMIZAÇÃO MULTI OBJETIVO DO TRANSPORTE DE PRODUTOS EM REDES DUTOVIÁRIAS ATRAVÉS DO ALGORITMO *Shuffled frog-leaping* MODIFICADO

Fabiany Lamboia

Programa de Pós Graduação em Engenharia Elétrica e em Informática Industrial - CPGEI,
Universidade Federal Tecnológica do Paraná – UTFPR
Avenida Sete de Setembro, 3165, 80230-901, Curitiba, PR, Brasil
fabianylamboia@yahoo.com.br

Lúcia Valéria Ramos de Arruda, Flávio Neves-Jr.

Programa de Pós Graduação em Engenharia Elétrica e em Informática Industrial - CPGEI,
Universidade Federal Tecnológica do Paraná – UTFPR
Avenida Sete de Setembro, 3165, 80230-901, Curitiba, PR, Brasil
lvrarruda@utfpr.edu.br, neves@utfpr.edu.br

RESUMO

Uma melhoria na eficiência do transporte de produtos através de redes de dutos pode ser obtida por uma melhor alocação dos recursos disponíveis, contudo além de ser este um problema combinatório de difícil solução, é também um problema de otimização multiobjetivo. Para resolver este tipo de problema, o uso de metaheurísticas, como os algoritmos evolucionários multiobjetivos tornam-se uma alternativa eficiente. Este trabalho apresenta a aplicação de dois algoritmos evolucionários, o Algoritmo Micro Genético (μ AG) e o *Shuffled frog-leaping* (SFL) modificado, na otimização multiobjetivo das operações de uma rede de dutos. A fim de facilitar a modelagem do problema, algumas restrições operacionais da rede são consideradas como objetivos de otimização. O algoritmo SFL apresentou uma melhor convergência para todos os objetivos na solução do modelo quando comparado ao algoritmo μ AG.

PALAVRAS CHAVE. Otimização Multiobjetivo, Algoritmo SFL, Algoritmo Micro Genético.

Área principal: Metaheurísticas (MH), Otimização Combinatória (OC), PO na Área de Petróleo & Gás (P&G).

ABSTRACT

An efficiency improvement of products transport of through pipeline networks can be obtained by a better allocation of available resources. However that is a hard solution combinatorial problem with multiobjective optimization characteristics. An alternative to efficient solve this type of problem is the use of metaheuristics such Multiobjective Evolutionary Algorithms (MOEA). This works applies two MOEA methods, the micro Genetic Algorithm and the *Shuffled frog-leaping* Algorithm in the multiobjective optimization of a pipeline network operation. A set of operational constraints are considered as objective and they are introduced in the problem objective function. The SFL Algorithm has presented a better performance than the μ GA for the tested instances.

KEYWORDS. Multiobjective Optimization, SFL Algorithm, Micro-Genetic Algorithm.

Main area: Metaheuristics (MH), Combinatorial Optimization (OC), OR in Oil & Gas (P&G)

1. Introdução

A modelagem de sistemas envolvidos no gerenciamento das operações de uma rede de dutos é um problema de otimização que envolve complexas restrições operacionais. Este tipo de transporte por meio de dutos mostra-se altamente confiável e econômico, principalmente para grandes volumes. Porém, é uma atividade complexa dada a elevada taxa de ocupação das redes de distribuição e a quantidade de diferentes produtos que devem ser transportados sob condições operacionais diferenciadas. Diante disso, há uma grande necessidade de desenvolver técnicas que garantam uma utilização mais eficiente das redes.

Um uso mais eficiente dessas redes pode ser alcançado através de uma melhor alocação dos recursos a partir de análises de planos de produção, consumo, estocagem e da análise das restrições da rede. Por outro lado, a alocação de recursos além de ser um problema combinatório de difícil solução é em geral um problema multiobjetivo. Para resolver este tipo de problema os métodos determinísticos são geralmente ineficientes, sendo portanto uma alternativa, o uso de metaheurísticas adaptadas para problemas multiobjetivos, como os algoritmos evolucionários multiobjetivos (Yamamoto, 2009).

Um problema multiobjetivo caracteriza-se pela existência de vários critérios conflitantes que não podem ser otimizados simultaneamente. Decorrente deste fato, admite-se como solução não apenas uma única solução (ótima), mas um conjunto de soluções ótimas em que nenhuma solução do conjunto pode ser considerada superior às outras quando todos os critérios são analisados separadamente. Este conjunto é denominado conjunto ótimo de Pareto e as soluções que o compõe são conhecidas como soluções não dominadas (Coello, 2003).

Apesar da variedade de métodos existentes e das melhorias alcançadas nos últimos anos, a eficiência computacional dos algoritmos evolucionários multiobjetivos (AEMO) tem limitado o seu uso na solução de problemas combinatórios de grande porte como é o caso do problema de transporte em rede de dutos (Westphal *et al.*, 2011). Uma solução corrente para reduzir o tempo computacional dos métodos AEMO, em especial dos algoritmos genéticos, é o uso de uma população pequena, associado a um processo de reinicialização dessa população a partir de uma memória externa que armazena as soluções não dominadas encontradas ao longo das gerações do método. Essa classe de algoritmo genético é denominada na literatura de algoritmo micro-genético (Coello e Pulido, 2001).

Neste trabalho dois algoritmos evolucionários com população reduzida, o Algoritmo micro Genético (μ AG) tal qual proposto por Westphal *et al.* (2011) e o *Shuffled frog-leaping* (SFL) modificado, são aplicados para otimizar as operações de uma rede de dutos que envolve um conjunto de restrições operacionais que influenciam significativamente em seu desempenho.

Na modelagem desta rede algumas destas restrições são consideradas como objetivos de otimização, configurando assim um problema multiobjetivo. Desta forma, os resultados obtidos a partir das simulações realizadas pelos dois algoritmos serão comparados considerando a minimização destes objetivos.

A seção 2 apresenta os algoritmos utilizados, a seção 3 descreve o problema em redes de distribuição, a metodologia utilizada se encontra na seção 4, em que uma ênfase especial é dada ao algoritmo SFL modificado, na seção 5 são apresentados os resultados obtidos e as conclusões são mostradas na seção 6.

2. Algoritmos Evolucionários

Os algoritmos evolucionários são métodos de busca estocástica que imitam a evolução biológica natural e/ou o comportamento social das espécies. Esses algoritmos têm sido desenvolvidos para se chegar a soluções ótimas ou próximas do ótimo para problemas de otimização complexos e de grande escala (Elbeltagi, 2005).

Neste trabalho foram utilizados dois algoritmos evolucionários, o Algoritmo micro Genético (μ AG) e o *Shuffled frog-leaping* (SFL) modificado.

2.1 Algoritmos Genéticos

Os princípios básicos dos Algoritmos Genéticos (AGs) foram desenvolvidos por

Holland (1975). Este algoritmo é um método de busca baseada em mecanismos de evolução natural e genética.

Nos AGs, uma população inicial é gerada aleatoriamente ou utilizando métodos heurísticos, essas soluções representam pontos espalhados no espaço de busca. Cada solução possível do espaço de busca é representada sob a forma de uma cadeia ou vetor (geralmente representada por bits ou números inteiros), chamado cromossomo, consistindo de um conjunto de elementos, chamados genes, que possuem um conjunto de valores para as variáveis de otimização (Goldberg, 1989).

Um AG padrão utiliza três operadores genéticos: a seleção, cruzamento ou recombinação (*crossover*) e a mutação. Os operadores genéticos são aplicados aos indivíduos da população com o objetivo de reproduzir novos e melhores indivíduos a partir dos já existentes. As operações são necessárias para permitir a diversidade dos indivíduos, bem como explorar outras regiões do espaço de busca.

Os principais parâmetros do AG são: o tamanho da população, número de gerações e a taxa de probabilidade de *crossover* e mutação. Uma breve descrição do algoritmo pode ser vista na Figura 1.

```
Inicio
Gerar uma população inicial de M indivíduos;
Calcular o fitness para cada indivíduo da população M;
Ordenar os M indivíduos por ordem decrescente do valor
de fitness;
  Repetir por um número I de gerações:
    Selecionar indivíduos;
    Cruzar indivíduos selecionados;
    Mutar indivíduos selecionados;
    Avaliar o fitness da população;
  Fim;
Fim.
```

Figura 1: Pseudo-código do Algoritmo Genético básico

A extensão do AG para aplicações em problemas multiobjetivos requer a inclusão de um mecanismo para ranqueamento da população que realiza uma comparação entre todos os indivíduos a fim de determinar sua dominância no sentido de Pareto e um segundo mecanismo para manter a diversidade a fim de evitar a convergência prematura do algoritmo devido à reprodução preponderante de super-indivíduos (Coello, 2003). Estes dois mecanismos são responsáveis pelo aumento da carga computacional dos métodos multiobjetivos baseados em AG.

Neste contexto, os métodos baseados em algoritmo microgenético permitem a redução do tempo de processamento enquanto mantém uma precisão aceitável, se comparado com soluções obtidas via AG padrão (Coello e Pulido, 2001; Westphal *et al.*, 2011). Essa redução do tempo computacional deve-se ao fato que o pequeno número de indivíduos na população reduz o esforço para avaliação e ranqueamento de cada indivíduo. Além disso, o uso de estratégias de reinicialização da população após poucas gerações garante a convergência do algoritmo microgenético e mantém a diversidade da população, evitando uma convergência prematura para um super-indivíduo.

Neste trabalho utiliza-se o algoritmo microgenético multiobjetivo com formação de nicho e casta apresentado em Westphal *et al.* (2011), o qual foi originalmente desenvolvido para um problema de escalonamento de rede de dutos semelhante ao problema apresentado neste trabalho.

O algoritmo microgenético citado combina técnicas de ranqueamento e elitismo para a formação de nicho e castas que garantem ao final da evolução, um conjunto de boas soluções em relação aos diferentes objetivos do problema de otimização. A utilização de uma pequena população com uma memória externa e uma casta interna garante a diversidade na população e permite alcançar um conjunto de soluções bem distribuído ao longo da fronteira de Pareto do problema multi-objetivo. Mais detalhes podem ser encontrados em Westphal *et al.* (2011).

2.2 Shuffled frog-leaping

O algoritmo *Shuffled frog-leaping* (SFL), desenvolvido por Eusuff *et al.* (2006), é uma metaheurística que visa encontrar uma solução ótima global através da realização de uma busca heurística local. O algoritmo é baseado na evolução de memes transmitidos por indivíduos e na troca global de informações entre a população.

O algoritmo usa a evolução memética sob a forma de infecção de idéias entre os indivíduos na busca local, e uma estratégia de embaralhamento que permite a troca de informações entre as buscas locais para se mover em direção a um ótimo global.

No SFL, a população consiste de um conjunto de sapos (possíveis soluções) que são particionados em subconjuntos chamados de *memeplexes*. Os diferentes *memeplexes* são considerados como diferentes culturas de sapos que realizam a busca local em cada *memeplex*. Os sapos mantêm idéias individuais, que podem ser influenciadas pelas idéias de outros sapos e assim evoluem utilizando um processo de evolução memética.

Depois de um número de passos evolutivos meméticos, as idéias são passadas entre os *memeplexes* em um processo de embaralhamento. A busca local e o processo global de embaralhamento continuam até que o critério de convergência seja satisfeito. A posição do sapo com a pior aptidão é ajustada conforme as equações 1 e 2.

$$S = \min \{ \text{int} [\text{rand}().C.(X_b - X_w)], S_{max} \} \quad (1)$$

$$\text{Nova Posição} = X_w + S \quad (2)$$

onde $\text{rand}()$ é uma função aleatória no intervalo $[0,1]$ e S_{max} é o tamanho máximo do passo permitido para mudar a posição do sapo. A nova posição do sapo é calculada pela equação 2. Se o processo produzir um sapo melhor, o sapo X_w é substituído. Caso contrário, os cálculos com as equações 1 e 2 são repetidos, mas utilizando o sapo com o melhor *fitness* global X_g .

Uma versão modificada do algoritmo SLF foi utilizada neste trabalho. No SLF padrão cada sapo em um *memeplex* está tentando mudar sua posição em relação ao melhor sapo dentro do *memeplex* ou do melhor sapo global. Como pode ser visto na equação 1, quando a diferença da posição entre o pior sapo X_w e o melhor sapo X_b torna-se pequena, a mudança da posição do sapo X_w será muito pequena e, portanto pode estagnar em um ótimo local e levar à convergência prematura. Para superar tal ocorrência, o lado direito da equação 1 é multiplicado por um fator C chamado de “fator de aceleração de busca” (Elbeltagi, 2007).

Os principais parâmetros do SFL são: o número de sapos F , número de *memeplexes* m e o número de passos evolutivos para cada *memeplex*. Uma breve descrição do algoritmo pode ser vista na Figura 2.

```

Início
Gerar população aleatória de F sapos;
Calcular a aptidão para cada sapo F;
Repetir por N iterações:
    Ordenar os F sapos pelo valor de aptidão;
    Dividir os F sapos em m memeplexes;
    Para cada memeplex:
        Repetir por um número I de evoluções:
            Determinar o melhor e o pior sapo do memeplex;
            Melhorar a posição do pior sapo;
            Calcular a nova aptidão;
            Se a posição for melhorada substituir o sapo;
            Senão gerar um novo sapo aleatoriamente e substituir o sapo;
        Fim;
    Fim;
Embaralhar os memeplexes evoluídos;
Fim;
Fim.
    
```

Figura 2: Pseudo-código do Algoritmo *Shuffled Frog Leaping*

3. Descrição do Problema

O transporte de produtos em uma rede de distribuição de petróleo mostra-se altamente

confiável e econômico, porém é uma atividade complexa. Diante disso, há uma grande necessidade de desenvolver técnicas para uma utilização mais eficiente das redes (Boschetto *et al.*, 2010).

Uma rede de distribuição de petróleo é composta por um número de refinarias e terminais, ou áreas, interligadas por um conjunto de oleodutos, ou trechos de dutos, os quais operam o transporte de um conjunto de produtos (petróleos, derivados de petróleo e produtos orgânicos) entre áreas adjacentes.

Geralmente um produto é transportado de refinarias, portos e/ou centro de armazenagens para pontos de destino. Os dutos podem ser unidirecionais ou bidirecionais, em uma conexão unidirecional, os produtos fluem somente em um sentido, de um terminal para a refinaria ou somente da refinaria para o terminal. Já na conexão bidirecional, os produtos podem fluir tanto do terminal para a refinaria quanto da refinaria para o terminal, mas nunca ao mesmo tempo.

A malha dutoviária da indústria brasileira de petróleo é composta basicamente por duas redes principais, a rede de escuros e a rede de claros, diferenciadas pelo tráfego de diferentes tipos de produtos. A rede de escuros realiza o transporte de diversos tipos de petróleos e derivados pesados e a rede de claros transporta diversos tipos de derivados leves de alto valor agregado. A programação das operações envolvidas nestas duas redes é realizada de forma separada, visto que não há uma interdependência operacional significativa entre essas malhas.

Neste trabalho foi desenvolvido um modelo de otimização para a alocação e sequenciamento operacional para a rede de escuros.

A Rede de Escuros é parte integrante do sistema dutoviário do estado de São Paulo, que é operado pela Transpetro. Este sistema, conforme pode ser visto na Figura 3, possui uma malha de oleodutos que interliga quatro refinarias (REPLAN, REVAP, RECAP e RPBC) a seis terminais de distribuição (Guararema, Guarulhos, Barueri, São Caetano, Cubatão e Santos).

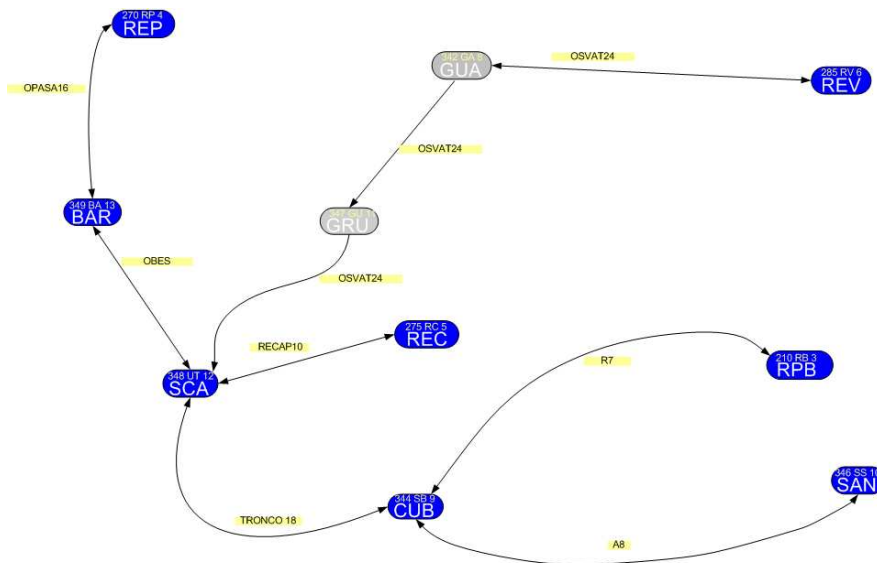


Figura 3: Rede de Escuros PETROBRAS

A Rede de Escuros é responsável pelo transporte dos derivados pesados do petróleo, como o óleo combustível, óleo combustível marítimo (*bunker*), gásóleo para craqueamento e resíduo atmosférico, que são produtos com coloração escura e alta viscosidade.

A modelagem da rede de escuros, utilizada neste trabalho, pode ser vista na Figura 4. A rede é composta por: oito nós, nos quais quatro deles são refinarias (nós N1, N2, N3 e N4), um porto (nó N8) e três centros de distribuição (nós N5, N6 e N7).

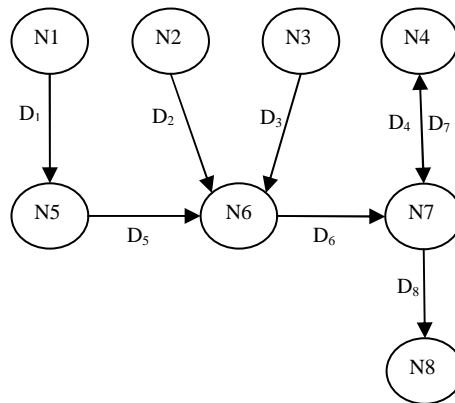


Figura 4: Modelo da Rede de Distribuição de Escuros

Os produtos produzidos nas refinarias N1, N2, N3 e N4 estão sendo designados de forma genérica e nomeados como produtos 1, 2, 3 e 4. O produto que cada fonte produz é configurado conforme desejado.

No modelo está sendo considerado que os produtos são entregues na forma de bateladas discretas. Uma batelada representa um volume mínimo de um produto a ser transportado em uma unidade de tempo. Uma batelada unitária é um volume mínimo que preenche um duto. Cada conexão possui uma distância normalizada em termos de unidades de tempo necessárias para que uma dada batelada seja transportada de um ponto a outro. O modelo assume a discretização do tempo.

Supõe-se que cada terminal ou nó intermediário possui a quantidade de tanques necessária para cada produto que ele possa receber. Todas as conexões têm as mesmas características como diâmetro, volume, velocidade do fluxo etc. As conexões D_1, D_2, \dots, D_8 possuem distâncias normalizadas em termos de unidades discretas de tempo necessárias para que uma dada batelada seja transportada de um ponto a outro.

4. Otimização Multiobjetivo

A metodologia aplicada para calcular o desempenho da solução utiliza a aproximação do método do critério global e do critério da ponderação, onde um peso é associado a cada objetivo (Coello *et al.*, 1995). A avaliação final do modelo, o *fitness*, será dada pela média da soma de todos os objetivos levando em consideração seus respectivos pesos, conforme mostrado na equação 4 (Westphal *et al.*, 2007).

$$\text{fitness} = \frac{\sum_{i=0}^{N_0} w_i \cdot f_i}{\sum_{i=0}^{N_0} w_i}, w_i \geq 0 \quad (4)$$

onde w_i é o peso para cada objetivo 'i', e f_i é a função de minimização para cada objetivo 'i'. Os resultados, para cada objetivo, ficam dentro da faixa $[0; 1]$, de tal forma que 0 é o valor ótimo, ou seja, todos os critérios foram cumpridos. Qualquer valor entre 0 e 1 mostra que o critério foi atendido parcialmente.

Os objetivos de otimização para o modelo desenvolvido são:

1. Demanda de fornecimento: definido como a quantidade mínima de bateladas de cada produto que deve ser enviada;
2. Demanda de recebimento: definido como a quantidade de bateladas de cada produto que deve ser entregue aos terminais;
3. Minimização do tempo para atendimento à demanda;
4. Minimização da fragmentação no envio das bateladas: evitar a alternância de produtos no envio das bateladas dos produtos.

O modelo está sujeito às seguintes restrições:

1. Não deve haver colisões de bateladas na conexão bidirecional.
 $NC = 0$ (5)

2. O número de bateladas nos tanques não pode violar o limite mínimo e máximo para cada nó 'i' e produto 'j'.
 $LCm_{ij} \leq C_{ij} \leq LCM_{ij}$ (6)

As restrições são implementadas por funções de reparação, conforme descrito em Westphal (2006). Os objetivos 1 e 2 são restrições que são tratadas como objetivos. Os objetivos são:

1. Satisfazer a produção mínima:

$$\min \left(\frac{\sum_{j=0}^{N_f} \sum_{i=0}^{Nt_j} \begin{cases} 1 - \frac{E_{ij}}{P_{ij}}, & E_{ij} \leq P_{ij} \\ 0, & E_{ij} > P_{ij} \end{cases}}{Nt_j} \right), \quad \text{para } j = 1,2,3,4 \quad (7)$$

onde Nt_j é a quantidade tanques, N_f é a quantidade de fontes, P_{ij} é a quantidade mínima de bateladas a serem enviadas pela fonte e E_{ij} é a quantidade de bateladas enviadas para cada fonte 'i' e produto 'j'.

2. Receber a quantidade de bateladas demandada (D_{ij}) pelos terminais:

$$\min \left(\frac{\sum_{j=0}^{Nt_j} \sum_{i=0}^{Nd} 1 - \frac{R_{ij}}{D_{ij}}}{Nt_j} \right), \quad \text{para } i = 1,2,3,4 \quad (8)$$

onde N_d é a quantidade de destinos, R_{ij} é a quantidade de bateladas recebidas do produto 'i' no terminal 'j'.

3. Minimizar o tempo de entrega das bateladas. O termo $(D_{ij} - R_{ij})$ é uma penalidade caso a demanda não seja cumprida.

$$\min \left(\frac{\sum_{j=0}^{Nt_j} \frac{Tchegada_{ij} - Tmim_j}{Tmax - Tmim_j} + (D_{ij} - R_{ij})}{(D_{ij} - R_{ij}) + 1} \right), \quad \text{para } i = 1,2,3,4 \quad (9)$$

onde $Tchegada_{ij}$ é o tempo de chegada de uma batelada do produto 'i' no terminal 'j', $Tmax$ é tempo de horizonte e $Tmim$ é o menor tempo para receber uma batelada em um terminal.

4. Minimizar a fragmentação das bateladas:

$$\min \left(\frac{\sum_{i=0}^{Nconex} \begin{cases} \frac{Frag_i - (Mask[i] - 1)}{Frag_i}, & Frag_i \geq Mask[i] \\ 0, & Frag_i < Mask[i] \end{cases}}{Nconex} \right) \quad (10)$$

onde $Nconex$ é quantidade de conexões, $Frag_i$ é a quantidade de fragmentações na conexão 'i' e $Mask$ é um vetor contendo a quantidade de produto em cada conexão em um dado instante de tempo.

A metodologia desenvolvida é mostrada na Figura 5 e foi implementada em linguagem C++ utilizando a ferramenta C++Builder 2007 versão 11.

Primeiramente uma população é inicializada com valores aleatórios. Em seguida, a função reparadora corrige as soluções infactíveis desta população de acordo com a violação de cada restrição. As violações referem-se aos limites máximo e mínimo dos tanques, a sobre produção e a existência de colisões. A população é em seguida avaliada e ranqueada de acordo com a função objetivo na equação 4. Para isto são calculados os quatro objetivos referentes à produção mínima, ao atendimento à demanda (restrições modeladas como objetivos), ao tempo deste atendimento e por fim à fragmentação (objetivos propriamente ditos).

Após o ranqueamento, para o algoritmo μ AG as castas elitistas são formadas e alguns indivíduos são selecionados para a aplicação dos operadores genéticos de recombinação e mutação (Westphal *et al.*, 2011). Já para o algoritmo SFL, a população é então dividida em *memeplexes* para ser realizada a busca local. Para ambos os algoritmos, a nova população é então formada e está sujeita novamente a função reparadora, pois soluções infactíveis podem surgir em qualquer ocasião após as operações genéticas e a evolução memética.

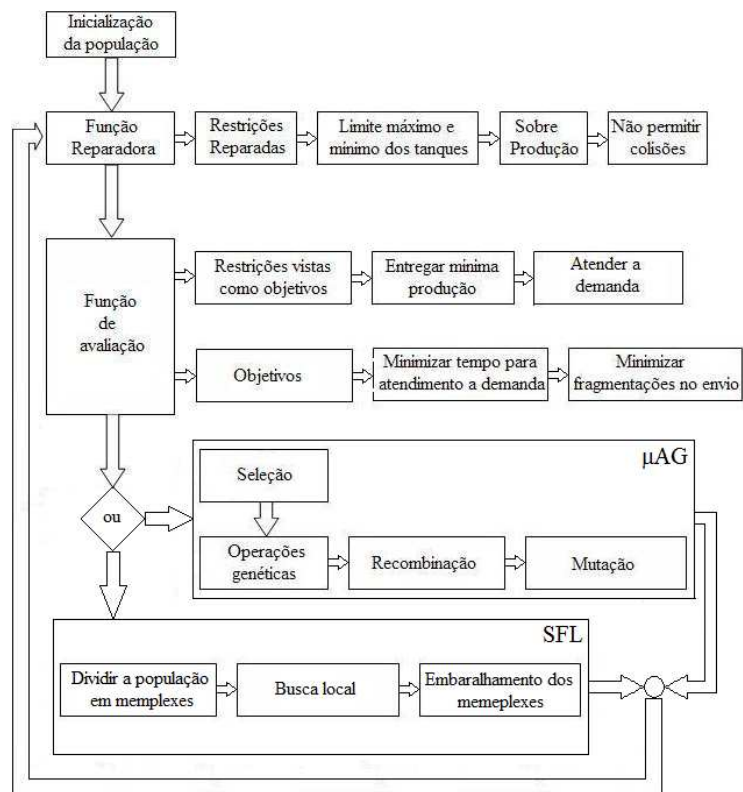


Figura 5: Metodologia para otimização multiobjetivo do transporte de produtos em rede de dutos

A partir da Figura 5, constata-se que o algoritmo microgenético multiobjetivo, a menos do mecanismo de formação de população, funciona exatamente como o algoritmo genético padrão dado na Figura 1 em que os operadores genéticos são responsáveis por criar novos e melhores indivíduos a partir dos já existentes.

5. Resultados e discussão

Para executar o modelo, é necessário configurar o limite superior e inferior dos tanques, a distância em termos de unidades discretas de tempo de cada conexão e o horizonte de programação. Esta configuração é uma decisão de projeto.

O horizonte de programação das simulações a serem apresentadas é de 48 unidades de tempo, ou seja, o último recebimento de produto possível será na unidade de tempo 48 e nenhum envio poderá ocorrer após esse tempo.

Os parâmetros de configuração do algoritmo μ AG são: P (população) = 20, taxa de mutação = 0.1, taxa de *crossover* = 0.8. Os parâmetros de configuração do algoritmo SFL são: F (população) = 20, m (memplexes) = 4, N (evoluções) = 5 e C (fator de aceleração) = 1.7. Os dois algoritmos são executados utilizando o mesmo número de gerações, que é igual a 1000.

Todas as simulações foram realizadas em um computador Intel Core2 Quad 2.83GHz, 4GB de memória RAM, sistema operacional Windows de 64 bits. Foram executados 20 experimentos de 1000 gerações, para cada um dos dois algoritmos. Os resultados estatísticos em termos de média, mediana, desvio padrão, máximo e mínimo de cada um dos objetivos (equações 7 a 10) e da função de avaliação (equação 4) estão mostrados nas Tabelas 1 e 2.

Tabela 1: Resultados para um conjunto de 20 simulações para o algoritmo μ AG.

Objetivo	Mínimo	Máximo	Média	Mediana	Desvio Padrão
1	0	0	0	0	0
2	0	0.0625	0.0125	0	0.0256
3	0.5486	0.7292	0.6130	0.6042	0.0366
4	0.1094	0.2240	0.1617	0.1576	0.0329
Global	0.1859	0.2321	0.2037	0.2013	0.0123

Tabela 2: Resultados para um conjunto de 20 simulações para o algoritmo SFL.

Objetivo	Mínimo	Máximo	Média	Mediana	Desvio Padrão
1	0	0	0	0	0
2	0	0.0625	0.0094	0	0.0229
3	0.5000	0.5833	0.5405	0.5417	0.0231
4	0.0313	0.0938	0.0641	0.0625	0.0250
Global	0.1531	0.1875	0.1714	0.1703	0.0095

A Figura 6 mostra a evolução do objetivo global do modelo, o *fitness* do melhor indivíduo para o experimento 1 (primeira simulação com 1.000 gerações), para ambos algoritmos durante as gerações.

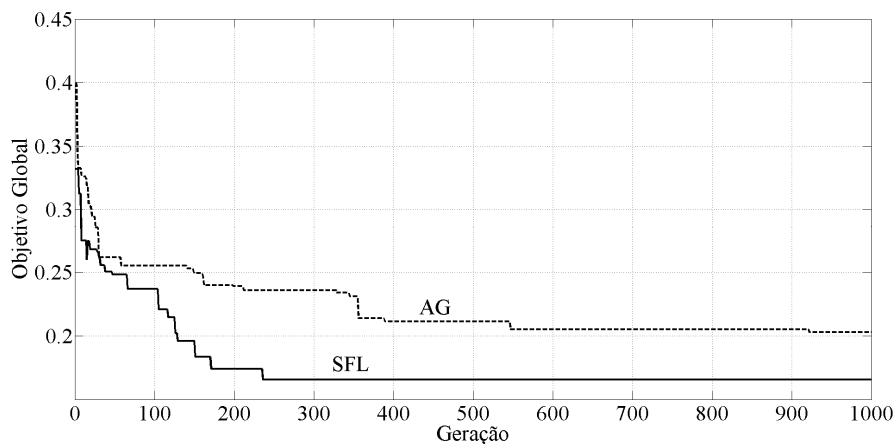


Figura 6: Objetivo Global

A Tabela 3 apresenta o desempenho por objetivo do modelo, para o melhor indivíduo do experimento 1, para o algoritmo μ AG e o algoritmo SFL modificado.

Os dois algoritmos utilizados apresentaram uma rápida convergência na solução do modelo. Pode se notar que para os objetivos 1 e 2, foi encontrada uma solução ótima, isto é, as demandas de recebimento e de fornecimento foram atendidas pelos dois algoritmos. Atender as

demandas é muito importante na solução do problema de redes de distribuição, pois é indesejável paralisar uma produção por falta de parque de armazenagem ou deixar faltar produtos nos pontos solicitantes (terminais).

Tabela 3: Valor do *Fitness* para todos os objetivos

Algoritmo/Objetivo	Global	1	2	3	4
μ AG	0.20	0	0	0.60	0.21
SFL	0.16	0	0	0.54	0.03

Atingir o valor zero para o objetivo 3, a minimização do tempo, não é possível, pois fisicamente é inviável que todos os produtos de um terminal sejam recebidos ao mesmo tempo com apenas uma conexão. Já para o objetivo 4, a minimização da fragmentação das bateladas, o algoritmo SFL chegou próximo de uma solução ótima.

O algoritmo SFL apresentou uma melhor convergência em todos os objetivos do modelo, alcançando melhores resultados na otimização do problema em questão. Esta rápida convergência pode ser associada a busca local utilizada por este algoritmo.

Como mencionado anteriormente, o SFL usa a evolução memética sob a forma de infecção de idéias entre os indivíduos na busca local, e uma estratégia de embaralhamento que permite a troca de informações entre as buscas locais para se mover em direção a um ótimo global.

No μ AG, a transmissão dos genes normalmente ocorre entre as gerações. Na evolução memética, se uma ideia melhorada é encontrado, ela pode ser incorporada em outros memes imediatamente, em vez de esperar para uma geração completa de genes a serem replicados. Além disso, a replicação do gene é limitada pelo número relativamente pequeno de descendentes que um único pai pode ter, ao passo que o número de indivíduos que podem assumir um meme a partir de um único indivíduo é quase ilimitado. Outra diferença entre memes e genes é que os memes são processados e possivelmente melhorados pelo indivíduo que os detém, algo que não pode acontecer com os genes (Eusuff *et al.*, 2006).

As Figuras 7 e 8 mostram a evolução de cada objetivo considerado no modelo, o valor do *fitness* do melhor indivíduo do experimento 1, para ambos algoritmos durante as gerações.

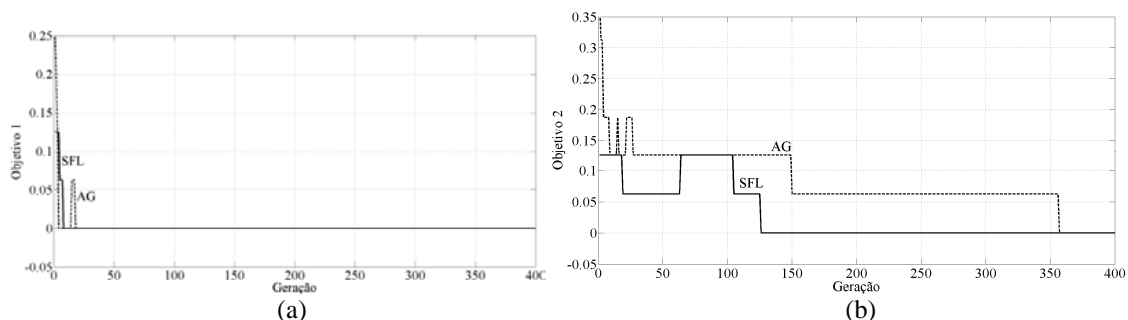


Figura 7: (a) Objetivo 1 e (b) Objetivo 2

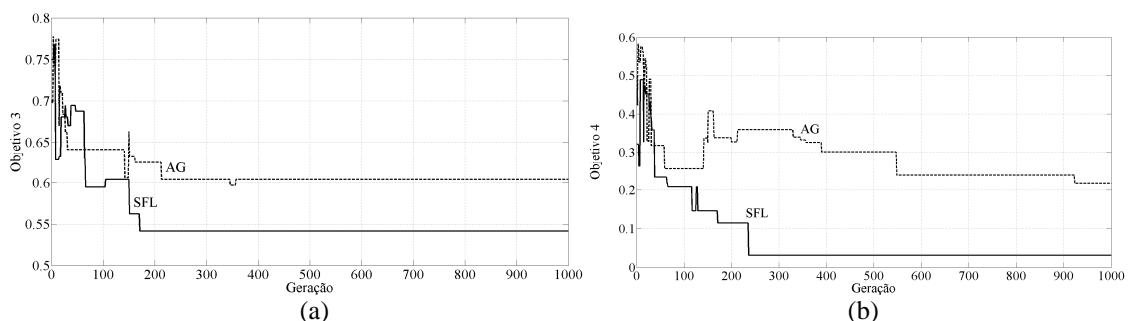


Figura 8: (a) Objetivo 3 e (b) Objetivo 4.

Para fins de comparação, a Figura 9 traz a carta de Gantt da solução do melhor indivíduo do experimento 1 ao longo do horizonte de programação utilizando o algoritmo SFL e o algoritmo μ AG, respectivamente. Nota-se que a conexão 8 é a mais ocupada durante todo o horizonte de tempo, pois para rede utilizada no modelo, o principal problema é atender a demanda no nó 8, que é o porto de Santos.

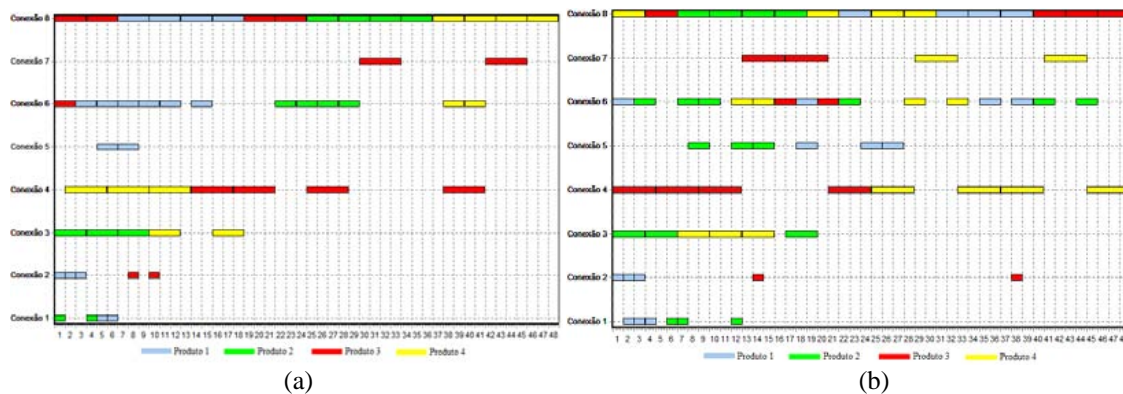


Figura 9: (a) *Shuffled Frog Leaping* e (b) Algoritmo micro genético

A minimização da fragmentação das bateladas alcançada pelo algoritmo SFL também pode ser notada na Figura 9(a). A fragmentação ocorre quando em uma sequência de bateladas diferentes produtos são enviados alternadamente, ao invés de ocorrer o envio de um mesmo produto sequencialmente. A alternância no tipo de produto a ser enviado pode provocar contaminações nas fronteiras, e por isso a fragmentação de bateladas em determinadas situações não é desejada.

O tempo de execução de ambos algoritmos para encontrar a solução do modelo utilizado neste trabalho, não foi apresentado devido a resposta da solução ser encontrada em poucos segundos.

6. Conclusão

Neste trabalho foram apresentados os resultados obtidos a partir da otimização multiobjetivo realizada por dois algoritmos evolucionários (μ AG e SFL). Foram aplicados os algoritmos em uma rede real de distribuição e feita uma análise dos resultados observando os objetivos de otimização.

Em dois objetivos ambos os algoritmos encontraram uma solução ótima. Por outro lado, nos objetivos relacionados à fragmentação dos produtos e minimização do tempo, o SFL modificado apresentou melhor desempenho do que o μ AG.

Na solução global do modelo, o SFL obteve uma melhor convergência em relação ao μ AG. Esta rápida convergência obtida pelo SFL pode ser associada à busca local que este algoritmo utiliza. Esta busca local pode ser uma alternativa para problemas mais complexos, como por exemplo, ao identificar um gargalo no sistema, pode-se aplicar a busca local para tentar solucionar este ponto em específico.

Por fim, os dois algoritmos conseguiram atender aos critérios do modelo, retornando soluções ótimas ou próximas do ótimo. Mas as metaheurísticas, tais como estes algoritmos evolucionários, são flexíveis e podem representar diversas classes de problemas. Esta flexibilidade deve-se principalmente ao fato de que cada metaheurística tem um ou mais parâmetros ajustáveis. Portanto, a “calibração” destes parâmetros dos algoritmos influencia na solução do modelo.

Agradecimentos

Este trabalho recebeu o apoio financeiro da Agência Nacional do Petróleo (ANP) e do CTPetro/Financiadora de Estudos e Projetos através do programa de recursos humanos para o

setor de petróleo e gás natural - UTFPR/PRH10, CENPES/PETROBRAS (0050.0044324.08.4).

Referências

Boschetto, S. N., Magatão, L., Brondani, W. M., Neves-Jr, F., Arruda, L.V. R., Barbosa-Póvoa, A. P. F. D. e Relvas, S. (2010), An operational scheduling model to product distribution through a pipeline network, *Industrial and Engineering Chemistry Research*, v.49, n.12, p.5661–5682.

Coello, C. A. C. (2003). Evolutionary multiobjective optimization: Current and future challenges. In F. Hoffmann J. Benitez, O. Cordon e R. Roy editores, *Advances in Soft Computing: Engineering, Design and Manufacturing*, p. 243 -256. Springer-Verlag, Heidelberg.

Coello, C. A. C. e Christiansen, A. D. (1995), An approach to multiobjective optimization using genetic algorithms, in *Intelligent Engineering Systems Through Artificial Neural Networks*, 1995, pp. 411–416.

Coello C.A.C. e Pulido, G.T. (2001). A micro-genetic algorithm for multiobjective optimization. In E. Zitzler, K. Deb, L. Thiele, C. Coello Coello, e D. Corne editores, *Evolutionary multi-criterion optimization (EMO 2001)*, p. 126 - 140. Springer-Verlag, Berlin, 2001.

Elbeltagi, E., Hegazy, T. e Grierson, D. (2005), Comparison among five evolutionary-based optimization algorithm, *Advanced Engineering Informatics*, Vol. 19, pp. 43–53.

Elbeltagi, E., Hegazy, T. e Grierson, D. (2007), A modified shuffled frog-leaping optimization algorithm: applications to project management, *Structure and Infrastructure Engineering*, Vol. 3, pp. 53–6.

Eusuff, M. e Lansey, K. (2003), Optimizing of water distribution network design using the shuffled frog-leaping algorithm. *Journal of Water Resources Planning and Management*, ASCE, Vol. 129, No. 3, pp. 210–225.

Eusuff, M., Lansey, K. e Pasha, F. (2006), Shuffled frog leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, Vol. 38, No. 2, pp. 129–154.

Goldberg, D. E. (1989). *Genetic algorithm in search, optimization and machine learning*. Reading, USA, Addison-Wesley Publishing.

Holland, J. H. (1975), *Adaptation in natural and artificial systems*. AnnArbor, University of Michigan Press.

Westphal, H. (2006), Algoritmo genético aplicado à otimização multiobjetivo em redes de distribuição de petróleos e derivados. Dissertação de Mestrado, Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Curitiba.

Westphal, H. e Arruda, L. V. R. (2007), Multiobjective Optimization Applied to the Distribution of Petroleum Products in Pipelines Networks, *Proceedings of 17th European Symposium on Computer Aided Process Engineering*. Amsterdam, Elsevier B.V. p. 1-6.

Westphal, H., Neves-Jr., F e Arruda, L. V. R. (2011), Algoritmo micro-genético aplicado ao scheduling de uma rede de distribuição de derivados de petróleo. In: H. S. Lopes; R. H. C. Takahashi. (Org.). *Computação Evolucionária em Problemas de Engenharia*. 1 ed. Curitiba: Omnipax, v. 1, p. 331-354.

Yamamoto, L. (2009), Um modelo baseado em metaheurística para o seqüenciamento de bateladas em redes dutoviárias. Tese de Doutorado, Universidade Tecnológica Federal do Paraná, Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Curitiba.