

Algoritmo Evolucionário para a Distribuição de Produtos de Petróleo por Redes de Polidutos

Thatiana Souza, Elizabeth Goldberg

Depto de Informática e Matemática Aplicada, UFRN,
Lagoa Nova, Natal, RN
thatinsouza@gmail.com, beth@dimap.ufrn.br

Marco Goldberg

Depto de Informática e Matemática Aplicada, UFRN,
Lagoa Nova, Natal, RN
gold@dimap.ufrn.br

Resumo: *Um importante problema quanto ao planejamento de produção em refinaria, é a determinação do que deve ser realizado em cada estágio da produção dado um determinado horizonte de tempo. Dentre tais problemas, a distribuição de produtos de petróleo através de redes de polidutos é um problema bastante significativo devido a sua importância econômica. Neste trabalho, um problema de distribuição de derivados de petróleo através de uma rede de dutos é modelado. A rede estudada é uma simplificação de uma rede real. Existem várias restrições a serem satisfeitas, tais como limites de estocagem, limites de envio ou recebimento e limites de transporte. O modelo adotado é bi-objetivo onde se deseja minimizar a fragmentação e o tempo de envio, atendendo as restrições de demanda e de capacidade de armazenamento. Considerando que a taxa de ocupação das redes está cada vez mais elevada, é de grande importância otimizar seu uso. Neste trabalho, a técnica de Otimização por Nuvem de Partículas é aplicada ao problema de distribuição de produtos de petróleo por redes de polidutos.*

Palavras-chave: Problema Multiobjetivo, Metaheurística, Otimização por Nuvem de Partículas.

Introdução

O propósito principal da programação da produção em refinarias é coordenar as operações de uma refinaria com seus objetivos de produção, de modo a otimizar a lucratividade do sistema de processamento ou a minimizar os seus custos. Vários são os fatores a serem considerados nas diretrizes de produção: alterações nas demandas, especificações dos produtos, datas de entrega, qualidade e quantidade das matérias primas, disponibilidade e desempenho das unidades de processo são alguns deles. Integrada fortemente a estratégias de planejamento, a atividade de programação da produção deve ser desenvolvida de maneira otimizada. Neste contexto, os principais fatores associados envolvem desde o suprimento de matéria prima até os sistemas de informação de mercado de produtos, passando pela operação da

planta e gerenciamento ótimo dos recursos disponíveis.

Uma rede de distribuição de petróleo é composta por refinarias, terminais e/ou parque de armazenagem, interligados por um conjunto de oleodutos, ou trechos de dutos, nos quais operam o transporte de petróleo e derivados entre áreas adjacentes. O transporte é realizado em sucessivos envios de bateladas. Uma batelada é a quantidade de um determinado produto transportado por um poliduto. Assim, o transporte destes produtos é motivado pelo cumprimento de uma demanda de mercado e/ou uma demanda de estoque e/ou por uma demanda de fornecimento de uma produção mínima ou por qualquer outro motivo operacional. Para a otimização da operação desta rede é proposta uma técnica metaheurística conhecida como Otimização por Nuvem de Partículas (*Particle Swarm Optimization*).

Modelo da Rede de Distribuição de Petróleo

O modelo tratado neste trabalho é uma simplificação de uma rede real. O modelo da rede tem um conjunto de nós composto por um conjunto de fontes (refinarias), um conjunto de terminais (pontos de destino ou pontos de entrega), e um conjunto de tanques intermediários (que podem ser receptores ou fornecedores de produtos (parque de armazenagem)).

A rede de distribuição de derivados em discussão pode ser vista na Figura 1. Nessa rede as refinarias estão representadas pelos nós $N1$ e $N2$. Os nós intermediários $N3$ e $N4$ representam parque de armazenagem de produtos. Os pontos de destino ou terminais estão representados pelos nós $N5$, $N6$ e $N7$. A rede apresenta 10 conexões representadas na figura pelas setas identificadas por D_1 , D_2 , D_3 , D_4 , D_5 , D_6 , D_7 , D_8 , D_9 e D_{10} . Uma das conexões é bidirecional e é representada por D_5 e D_6 .

As linhas que unem os nós representam os polidutos e as flechas representam a direção que flui o produto transportado. Ambos os nós, $N1$ e $N2$, fornecem produtos para $N3$ e $N4$ através das conexões D_1 , D_2 , D_3 e D_4 . Essas conexões são unidirecionais, ou seja, o produto flui somente na direção da fonte para

os nós intermediários. Os nós $N3$ e $N4$ são conectados através de um poliduto bidirecional $D5$ e $D6$. As conexões $D7$ e $D8$ conectam os nós $N5$ e $N6$ à $N3$ respectivamente. Já as conexões $D9$ e $D10$ conectam os nós $N6$ e $N7$ à $N4$ respectivamente.

Considera-se a tancagem agregada por produto, isto é, cada nó intermediário e/ou terminal possui a quantidade de tanques necessária para cada produto que ele possa receber. Ou seja, se em um nó é possível receber 4 tipos de produtos, então assume-se que nesse nó existem 4 tanques para a armazenagem desses produtos.

No modelo está sendo considerado que os produtos diferentes são entregues na forma de bateladas discretas. Poderá haver muitos diferentes tipos de bateladas como o número de diferentes produtos. Uma batelada representa um volume mínimo de um produto a ser transportado em uma unidade de tempo sobre o duto. Uma batelada unitária é um volume mínimo que preenche um duto. As bateladas também são utilizadas para medir o tamanho das conexões, ou seja, o número máximo possível de bateladas que podem ser transportadas por uma conexão. Na modelagem elas também são usadas para apresentar as capacidades de armazenamento, demandas e estoque atual. 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. A Tabela 1 mostra a distância de cada conexão.

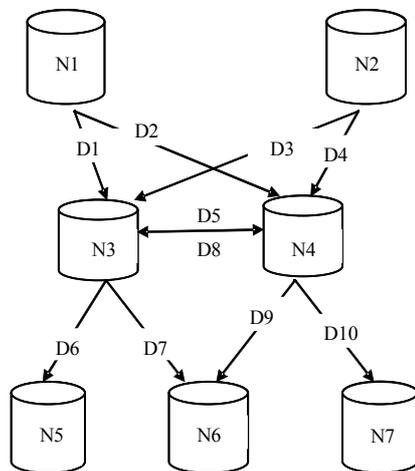


Figura 1: Modelo da rede de distribuição de derivados

Tabela 1: Distância entre as conexões.

Conexão	Distância
D_1	1
D_2	3
D_3	3
D_4	2
D_5	3
D_6	3
D_7	4
D_8	2
D_9	3
D_{10}	2

A fim de simplificar o problema, parte-se do princípio de que todos os polidutos são considerados com o mesmo diâmetro e mesmas características. Todos os produtos fluem com a mesma velocidade e ocupam o mesmo volume no duto. O envio consecutivo de diferentes produtos pode provocar contaminação, exigindo que o volume contaminado volte à refinaria para novo processamento. É dito que há fragmentação quando em uma seqüência de envios de bateladas, diferentes produtos são enviados alternadamente ao invés de ocorrer o envio de um mesmo produto e depois os outros produtos sequencialmente.

A solução para o problema é dada por cada tipo de batelada que é enviada em cada instante de tempo em cada conexão. Para armazenar a rede em memória foi criada uma matriz de tamanho $m \times m$, onde m é o número de nós da rede. Cada posição $[i, j]$ da matriz indica o tamanho da conexão ligando os nós i e j . Caso não exista ligação entre os nós o valor armazenado é zero. Para controle dos níveis dos produtos nos tanques foi construída uma matriz que armazena a quantidade mínima, atual e máxima de cada produto em cada nó. Essa matriz contém as quantidades iniciais de produtos dos nós no início da simulação e é alterada a cada nova solução sugerida. Para codificação de uma solução foi utilizada uma matriz que represente a solução onde as linhas representam as conexões da rede e as colunas os instantes de tempo. As conexões bidirecionais são desdobradas em duas conexões, onde cada conexão representa um possível caminho. O desdobramento das conexões bidirecionais se deve a necessidade de codificar o caminho de ida e de volta. Cada produto é codificado por um número inteiro. Em cada posição dessa matriz é preenchida um valor que indica o tipo de cada produto enviado no devido instante e conexão determinados pela linha e coluna. O valor zero significa que não existe produto alocado na conexão. Neste caso quando existe um produto na conexão $D5$ não existe produto em $D8$. Para a codificação é necessário saber o horizonte de tempo, que é uma definição de projeto. A Tabela 2 mostra um exemplo da matriz de solução da rede da Figura 1 para 12 instantes de tempo. Para exemplificar pode-se apontar

a conexão 2 nos instantes 1 e 2 enviando bateladas de produtos do tipo 2.

Tabela 2: Exemplo de matriz de solução.

instante \	1	2	3	4	5	6	7	8	9	1	1	1
Conx 1	1	1	0	3	3	1	0	2	2	2	0	0
Conx 2	2	2	0	1	1	1	1	1	1	0	0	0
Conx 3	0	0	3	3	3	3	1	1	1	1	0	0
Conx 4	1	1	1	1	1	2	2	2	2	0	1	1
Conx 5	0	0	0	1	1	1	2	2	2	0	0	0
Conx 6	1	1	1	3	3	3	3	3	3	0	0	0
Conx 7	3	3	3	2	2	2	2	0	0	1	1	1
Conx 8	1	1	1	1	1	2	2	2	2	3	3	3
Conx 9	0	0	0	0	0	1	1	1	1	2	2	0
Conx 10	0	0	1	1	1	2	2	2	3	3	3	3

Dessa forma, pode-se ainda exemplificar, que no instante 2 não haverá produto alocado na conexão 3, na conexão 4 haverá o produto 1, na conexão 5 não haverá produto, e assim por diante. Utilizar uma matriz como forma de representação em PSO acaba dificultando a implementação e com isso aumentando o esforço computacional. Para facilitar a implementação, a matriz é transformada em um vetor, ou seja, a solução será codificada em forma de um vetor, onde cada linha representará um indivíduo da população (iteração). O PSO é um algoritmo que possui um vetor de velocidades e outro de posição, a posição de cada partícula é atualizada de acordo com a velocidade atual. A solução em forma de matriz pode ser rearranjada em forma vetorial conforme mostrado na Tabela 3. Para associar o nó de origem e destino e a distância normalizada em termos de unidade de tempo de cada conexão, uma matriz é utilizada e pode ser vista na Tabela 4. A Tabela 5 mostra alguns parâmetros utilizados para implementação do algoritmo.

Tabela 3: Exemplo de vetor de solução.

Tempo	Instante 1	...	Instante 12
Conexão	1 ... 10	...	1 ... 10
Produto	110331022200	...	001112223333

Tabela 4: Matriz de conexão.

Conexão	Nó de origem	Nó de destino	Tempo
1	1	3	1
2	1	4	3
3	2	3	3
4	2	4	2
5	3	4	3
6	3	5	3
7	3	6	4
8	4	3	2
9	4	6	3
10	4	7	2

Tabela 5: Descrição de parâmetros utilizados no algoritmo.

Parâmetro	Descrição
nIndiv	Número de indivíduos da população
nConex	Número de conexões da rede
nInstantes	Número de instantes a ser programado pelo algoritmo
nGeracoes	Número de gerações

Algoritmo Proposto

Neste trabalho, utilizou a técnica de Otimização por Nuvem de Partículas (PSO) conforme sugerida no trabalho de Goldberg *et al.* (2006a e 2006b). São usados procedimentos de busca local e *path-relinking* como operadores de velocidade.

A busca local é um método tradicional de otimização que inicia com uma solução qualquer e procede procurando por uma melhor solução na vizinhança definida para a solução inicial. Se uma solução melhor é encontrada, então ela se torna a nova solução do problema e novamente o processo de busca na vizinhança é executado. Segundo Aarts e Lenstra (1997), este processo continua até que nenhuma solução melhor seja encontrada. O *Path-relinking* é uma técnica de intensificação cuja idéia foi originalmente proposta por Glover (1963).

Neste trabalho, trata-se de um problema de otimização bi-objetivo, onde temos duas funções a otimizar: minimizar a fragmentação no envio das bateladas e minimizar o tempo de envio.

O pseudocódigo do algoritmo multiobjetivo PSO é apresentado na Figura 3.

```

pr1 = x /* seguir o próprio caminho, v1 */
pr2 = y /* seguir pbest, v2 */
pr3 = z /* seguir gbest, v3 */
GeraPopInicial(Pop)
Repita
  reparaTempo();
  reparaDemanda();
  reparaCapacidade();
  para i = 1 até nro_part faça
    /* Avaliar as partículas */
    atualiza(pi, pbest[i])
    atualiza(pi, gbest)
  fim_para
  para i = 1 até nro_part faça
    altera_posição(pi, pr1, pr2, pr3)
  fim_para
  /* atualize as probabilidades */
  pr1 = pr1 * 0.95
  pr2 = pr2 * 1.01
  pr3 = pr3 * 100% - (pr1 + pr2)
  iter = iter + 1
Até que (numg < nGerações)
Retornar_soluções_não_dominadas(gbest)

```

Figura 3: Pseudocódigo do algoritmo Multiobjetivo PSO.

No início, o algoritmo define as probabilidades associadas com cada velocidade para as partículas, onde pr_1 , pr_2 e pr_3 correspondem, respectivamente, à probabilidade de que a partícula siga seu próprio caminho, siga em direção a sua melhor posição já atingida ($pbest$) e siga em direção a melhor posição encontrada até o momento ($gbest$). Então, os algoritmos procedem modificando a posição das partículas de acordo com o operador de velocidade aleatoriamente escolhido. No fim, as probabilidades são atualizadas.

Inicialmente, é atribuído um valor alto para pr_1 e valores baixos para pr_2 e pr_3 . O objetivo é que a partícula siga movimentos próprios mais frequentemente nas primeiras iterações. Durante a execução do algoritmo, esta situação vai sendo alterada e, nas iterações finais, pr_3 terá o maior valor. A idéia é intensificar a busca em boas regiões do espaço de busca nas iterações finais. A busca local é utilizada quando a partícula segue seu próprio caminho e o *path-relinking* é utilizado nas ocasiões onde a partícula segue em direção a outra partícula.

A posição $pbest$ trata-se de um arquivo ilimitado, é utilizado como soluções não dominadas. Assim, se for uma solução não dominada, essa entra no arquivo. Será também alterada a velocidade da partícula na posição p_i .

A função de geração da população inicial $GeraPopInicial(Pop)$ se dá de maneira aleatória, sorteando um produto para cada posição do vetor de soluções, de cada indivíduo. No pseudocódigo da Figura 3 $nGerações$ é o número de gerações definido pelo usuário que o algoritmo irá executar.

Caso a operação algoritmo faz uso de três funções reparadoras durante sua execução:

- Função Reparadora de Tempo: percorre cada conexão ligada a um nó destinatário retirando os pacotes que violem os limites de tempo máximo de chegada definida para cada produto em cada nó destinatário.
- Função Reparadora de Capacidade: verifica para cada instante de tempo se o pacote que está chegando viola a capacidade mínima do nó fornecedor ou a capacidade máxima do nó destinatário. Caso viole o pacote é retirado da solução, sendo adicionado novamente ao estoque do nó fornecedor.
- Função Reparadora de Demanda: tem funcionamento semelhante à reparação de capacidade máxima, a cada instante de tempo verifica se a demanda do nó destinatário foi atendida, caso já tenha sido atendida o pacote é retirado da conexão e colocado novamente no estoque do nó fornecedor.

Resultados Obtidos

Apesar das diferenças nas abordagens adotadas na modelagem e construção dos algoritmos algumas comparações podem ser feitas. O trabalho de (CRUZ,

2003) trata-se de um Algoritmo Evolucionário Multiobjetivo com Restrições. Para a sua implementação foi utilizada a modelagem proposta no seu artigo, assim como os operadores genéticos e os métodos de avaliação, cruzamento e substituição propostos, destacando ainda que nesse trabalho havia um grande número de fragmentações e a possibilidade de ocorrer colisões. Enquanto, neste trabalho utiliza-se da técnica PSO, verificando uma grande melhoria em relação aos objetivos propostos de minimização da fragmentação total e do tempo total da rede.

Os objetivos são as minimizações da fragmentação total e do tempo da rede. A fragmentação total da rede é a soma das fragmentações em todas as conexões da rede. A Figura 4 apresenta o gráfico contendo o número de fragmentações e o tempo. Pode ser visto também na Figura 4 a evolução do objetivo de minimização do tempo total gasto na rede na execução dos testes.

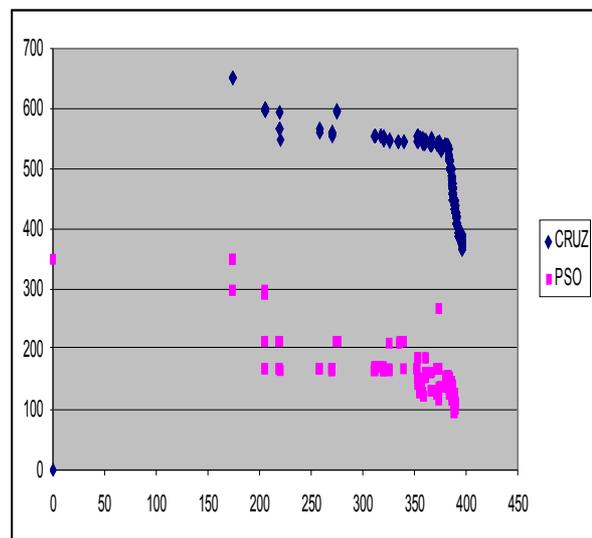


Figura 4: Fragmentação e tempo total da rede.

Conclusões

A programação da produção e alocação de tarefas continua sendo um problema atual, e sendo um mercado dinâmico em que a tomada de decisão precisa ser rápida, boas soluções devem ser encontradas em um curto espaço de tempo. A busca pela solução ótima demanda elevado esforço computacional e conseqüentemente tempo. PSO vem de encontro a esse problema já que é capaz de fornecer boas solução em tempo hábil. Muitos objetivos devem ser satisfeitos dentro da programação da produção e por isso faz-se necessário o uso da otimização multiobjetivo. A vantagem para o uso e desenvolvimento desse procedimento é o tratamento simultâneo de todos os objetivos identificados no problema buscando a solução de melhor compromisso neste cenário.

Este trabalho apresentou uma modelagem de um problema de distribuição de derivados de petróleo

através de uma rede de dutos. A rede proposta é uma simplificação de uma rede real. O método proposto foi baseado na utilização da técnica PSO, o qual garante ao final da evolução um conjunto de boas soluções em relação aos diferentes objetivos do problema de otimização. Ao término das gerações, selecionam-se as soluções ótimas de Pareto por serem consideradas as melhores soluções. O tratamento das restrições via função reparadora foi uma contribuição essencial para evitar soluções infactíveis.

Os resultados apresentados mostram que em um curto espaço de tempo o modelo é capaz de convergir e boas soluções são obtidas. Como contribuição final, o trabalho confirmou resultados da literatura, que estabelecem que PSO pode e deve ser aplicado na resolução de problemas complexos como a distribuição de petróleo e derivados. A utilização dessa metaheurística não demanda cálculos complexos e por isso proporciona relativa rapidez na obtenção de boas soluções, o que é primordial em operação de plantas reais como uma malha de dutos.

Agradecimentos

À Agência Nacional do Petróleo (ANP), pelo fundamental apoio financeiro indispensável para a realização deste trabalho.

Referências

- [1] AARTS, E., LENSTRA, J. K., “Local search in combinatorial optimization”. John Wiley & Sons, Chichester, England, 1997.
- [2] Coelho, C. A. C., Lechuga, M. S. MOPSO. “A proposal for Multiple Objective Particle Swarm Optimization”, Congress on Evolutionary Computation, pp. 1051-1056, 2002.
- [3] Cruz, J. M., Andrés-Toro, B., Herrán, A., Besada, E. P., Blanco, P. F., “Multiobjective optimization of the transport in oil pipeline networks”, *IEEE International Conference on Emerging Technologies and Factory Automation 1*, pp. 566-573, 2003.
- [4] Glover, F., “Parametric combinations of local job shop rules”. Chapter IV, ONR Research Memorandum N. 117, GSIA, Carnegie Mellon University, Pittsburgh, PA, 1963.
- [5] GOLDBARG, E.F.G, SOUZA, G.R., GOLDBARG, M.C. “Particle Swarm for the Traveling Salesman Problem”, EvoCOP, pp. 99-110, 2006a.
- [6] GOLDBARG, E.F.G., SOUZA, G.R., GOLDBARG, M.C. “Particle swarm optimization for the bi-objective degree-constrained minimum spanning tree”, *IEEE Congress on Evolutionary Computation 1*, pp. 1527-1534, 2006b.
- [7] KENNEDY, J., EBERHART, R. “Particle swarm optimization,” *IEEE International Conference on Neural Networks 4*, pp. 1942-1948, 1995.
- [8] Mostaghim, S., Teich, J. “Strategies for finding good local guides in multiobjective particle swarm optimization (mopso)”, pp. 26-33, 2003.
- [9] Pareto, V. *Manuel D’Économie Politique*. Paris: Marcel Giard, 1927.