

# Otimização do Processo de Transferência e Estocagem de Diesel em uma Refinaria de Petróleo aplicando Transgenética Computacional e Programação Linear Inteira Mista

Angela Olandoski Barboza (UTFPR) [angelabarboza@utfpr.edu.br](mailto:angelabarboza@utfpr.edu.br)

Flavio Neves Junior (UTFPR) [neves@utfpr.edu.br](mailto:neves@utfpr.edu.br)

Rosely Antunes de Souza (UFSC) [roselypr@gmail.com](mailto:roselypr@gmail.com)

Silvana Ligia Vincenzi Bortolotti (UFSC) [sligia@globo.com](mailto:sligia@globo.com)

## RESUMO

*Hoje, as empresas brasileiras vivem uma realidade em que ganhos de produtividade significam sobrevivência. Para tanto, busca-se a criação de técnicas que proporcionem aos empresários e gerentes a certeza de que estão utilizando um sistema de produção que, se bem administrado, pode gerar vantagens competitivas. Na indústria petrolífera, um dos processos que pode ser aperfeiçoado, usando técnicas da Pesquisa Operacional, é o de transferência e estocagem de diesel. Este trabalho descreve primeiramente, a modelagem deste problema usando Programação Linear Inteira Mista (PLIM) com abordagem de tempo discreto. O modelo encontrado foi resolvido com o aplicativo computacional LINGO 8.0, por meio do algoritmo Branch and Bound. Como o tempo computacional para a resolução deste modelo mostrou-se excessivo, desenvolveu-se uma nova metodologia usando-se algoritmo transgenético ProtoG integrado à Programação Linear (PL). Quando comparada à modelagem usando PLIM, pode-se observar que o resultado obtido após vários testes com a nova metodologia mostrou que esta pode encontrar bons resultados em um tempo computacional aceitável.*

Palavras-Chave: Programação da produção (*scheduling*), PLIM, Transgenética Computacional

## 1. INTRODUÇÃO

O mercado mundial vem se modificando em decorrência dos avanços tecnológicos, da globalização, das grandes fusões de empresas e da maior conscientização ecológica. As transformações econômicas, a dinâmica dos mercados e a crescente competitividade fazem parte da globalização mundial, a qual intensifica o comércio internacional de produtos e serviços, promove intercâmbio cultural acentuado e a constante troca de informações. Tais mudanças implicam em um aumento da competitividade, obrigando as organizações a criarem soluções inovadoras para se manterem no mercado. No contexto das indústrias, um dos problemas é o de otimização do sistema de produção no qual, o planejamento e a programação de produção são de fundamental importância. O uso de técnicas de Pesquisa Operacional na modelagem de processos das empresas tem-se mostrado um fator decisivo para o desenvolvimento de políticas otimizadas de operação industrial. O desenvolvimento de modelos, em especial os que empregam técnicas de otimização, tem possibilitado que procedimentos operacionais complexos sejam avaliados de forma criteriosa, fazendo com que recursos críticos possam ser utilizados da melhor maneira possível.

Este trabalho aborda um problema de programação da produção (*scheduling*) envolvendo estocagem e distribuição de óleo diesel em uma refinaria de petróleo. A solução ótima para este tipo de problema é difícil de ser encontrada devido à sua característica combinatorial. Em geral, este é modelado como problema de Programação Linear Inteira Mista (PLIM) e resolvido com auxílio de pacotes computacionais comerciais (MORO, 2000). Contudo, pela sua natureza, o aumento do número de variáveis inteiras torna impraticável o

uso destes aplicativos, por exigirem um tempo computacional excessivo. Desta forma, não são aplicáveis em sistemas *just in time*, pois a tomada de decisão nestes casos deve ser imediata.

Diante do exposto, para solucionar este problema foram utilizados, a princípio, modelos de Programação Linear Inteira Mista (PLIM) com abordagem para a representação no tempo discreto. O modelo desenvolvido foi resolvido com o uso do aplicativo computacional LINGO 8.0, por meio do algoritmo *Branch and Bound* com o objetivo de obter-se médias para o tempo computacional despendido e número de iterações para a resolução do mesmo, para fins de comparação com a nova metodologia. A nova metodologia desenvolvida utiliza o Algoritmo Transgenético ProtoG integrado à Programação Linear (PL).

O objetivo geral deste trabalho é o desenvolvimento de uma metodologia mais eficiente que o algoritmo *Branch and Bound*, em termos de tempo computacional despendido, para resolver um problema de Programação Linear Inteira Mista (PLIM) modelado a partir de um problema de transferência e estocagem de diesel em uma indústria petrolífera. Como objetivos específicos, tem-se a construção do modelo em PLIM e o desenvolvimento da nova metodologia usando Algoritmo Transgenético ProtoG integrado à Programação Linear (PL).

Este artigo está estruturado da seguinte forma: A seção 1 descreve os problemas que motivaram esta pesquisa, justificando o desenvolvimento de nova metodologia para a otimização do problema de transferência e estocagem de diesel. Na seção 2 é feita uma breve revisão bibliográfica dos seguintes assuntos: Programação Linear (PL), Programação Linear Inteira Mista (PLIM) e Transgenética Computacional. A seção 3 mostra a modelagem do problema analisado. O modelo em PLIM pode ser encontrado na seção 4 e a nova metodologia foi descrita na seção 5. A seção 6 foi dedicada à implementação e resultados e finalmente, na seção 7, são apresentadas a análise dos resultados e conclusões obtidas.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1 PROGRAMAÇÃO LINEAR

Programação Linear (PL) é uma ferramenta da Pesquisa Operacional aplicada à solução de problemas de otimização que buscam a distribuição eficiente de recursos limitados para atender um determinado objetivo, em geral, maximizar lucros ou minimizar custos. É uma das mais importantes e mais utilizadas técnicas de Pesquisa Operacional. Sua modelagem é simples e a resolução é feita utilizando-se o método Simplex descrito por Dantzig (1963). Esta técnica é amplamente utilizada, pois possui habilidade para modelar importantes e complexos problemas de decisão e o método Simplex pela capacidade de produzir soluções rapidamente. A descrição do método Simplex pode ser encontrada em Zionts (1974).

### 2.2 PROGRAMAÇÃO LINEAR INTEIRA MISTA (PLIM)

Alguns problemas reais requerem o uso de variáveis que assumem somente valores inteiros. Quando isto acontece tem-se um problema de Programação Inteira (PI). Se um problema envolve variáveis inteiras e contínuas então é denominado Problema de Programação Linear Inteira Mista. Existem muitos problemas de programação de produção (*scheduling*) que podem ser colocados como problemas de Programação Linear Inteira Mista, pois os modelos matemáticos de otimização correspondentes envolvem variáveis contínuas e discretas que devem satisfazer um conjunto de restrições lineares de igualdade e desigualdade (MORO, 2000).

A obtenção de uma solução ótima para problemas de otimização linear inteira mista, pode ser difícil, pela sua natureza combinatorial. Num primeiro contato com este tipo de problema, a abordagem seria a de resolver o problema para todas as combinações de variáveis inteiras utilizando a PL e extrair a solução como o menor valor da função objetivo (problemas

de minimização). Porém, o número de combinações cresce exponencialmente com o número de variáveis binárias. Logo, para problemas práticos onde se tem um grande número de variáveis inteiras, esta abordagem é inviável. Uma outra alternativa seria a de relaxar as restrições de integralidade e tratar as variáveis inteiras como contínuas, mas não se tem a garantia de que, resolvendo o problema com esta relaxação, encontre-se uma solução com valores inteiros para as variáveis discretas. O arredondamento para valores mais próximos também não leva, em geral, ao resultado correto. Taha (1975) mostra, através de um exemplo, que esta abordagem não é conveniente. Atualmente, o método mais utilizado na resolução deste tipo de problema é o *Branch and Bound*.

### 2.3 TRANSGENÉTICA COMPUTACIONAL

Transgenética Computacional (TC) é uma nova técnica da Computação Evolucionária (CE). É considerada uma abordagem evolucionária, pois emprega populações de indivíduos, as populações trocam informações dinamicamente e somente os mais aptos sobrevivem em cada ciclo de troca de informações. O paradigma da TC está contextualizado na evolução dos micro-organismos e células e fundamentam-se no processo da simbiogênese (GOLDBARG, GOLDBARG e MEDEIROS NETO, 2005). A simbiogênese é uma teoria evolucionária onde indivíduos de naturezas distintas se unem para formar um novo indivíduo (MOROWITZ, 1992). Esta teoria destaca mais os efeitos positivos resultantes das inter-relações entre indivíduos do que a sobrevivência e reprodução do mais apto.

Dentro do contexto da simbiogênese, a aplicação da metáfora aos algoritmos transgenéticos é feita de forma que:

- processo evolucionário dos algoritmos transgenéticos esteja baseado em interações simbióticas desenvolvidas entre uma população de cromossomos e uma população de vetores do fluxo intracelular;
- o resultado da simbiose seja avaliado pela adequação alcançada pelos cromossomos;
- os mecanismos evolucionários empregados nos algoritmos transgenéticos mimetizem os processos de conjugação e de recombinação (processos de transferência horizontal);
- os vetores empregados mimetizem os plasmídios, vírus e transposons;
- o processo evolucionário seja gerenciado por três tipos de regras, descritas a seguir;
- o paradigma da transgenética possua os seguintes princípios gerais: a evolução pode ocorrer sem reprodução sexual ou mutações, os vetores do fluxo intra-extracelular cooperam com a evolução dos cromossomos e o fluxo de informações do processo evolucionário percorre vários níveis de decisão.

A metáfora transgenética possui, três níveis onde as informações são armazenadas:

- Primeiro nível: é composto pela população de cromossomos;
- Segundo nível: é composto por uma população de indivíduos de natureza distinta da dos cromossomos. Estes indivíduos são chamados de vetores transgenéticos e operam na diversificação e intensificação da busca. Os vetores transgenéticos são detalhados a seguir.
- Terceiro Nível: é formado por regras que direcionam a interação entre as duas populações (cromossomos e vetores transgenéticos).

Um cromossomo é composto por uma cadeia de informação e tem associados um ou mais valores de adequação.

Com o objetivo de explorar o espaço de busca, os algoritmos transgenéticos obtêm informação e a inserem nos cromossomos através dos vetores transgenéticos.

Um vetor transgenético  $\lambda$  é uma dupla  $\lambda = (I, \Phi)$ , onde  $I$  é uma cadeia de informação e  $\Phi$  é um método de manipulação. A cadeia de informação  $I$  pode ser *a priori* ou *a posteriori*. A informação *a priori* pode vir de diversas fontes, tais como conhecimento teórico e heurístico. A informação *a posteriori* é obtida através do próprio processo de busca. Estas cadeias podem ser reunidas em um banco de informações genéticas. Este banco de informações visa acelerar o processo de evolução.

O método de manipulação  $\Phi = (p_j)$  com  $j = 1, \dots, s$  é composto pelos procedimentos  $p_j$  que definem a atuação do vetor. A Tabela 1 resume os procedimentos que compõem o método de manipulação dos vetores transgenéticos.

Tabela 1: Procedimentos Gerais de um Vetor Transgenético

Procedimento	Denominação	Descrição
Procedimento 1 ( $p_1$ )	Ataque (A)	Define o critério de avaliação que estabelece quando um cromossomo é suscetível à manipulação do vetor. $A : S_i \rightarrow \{\text{falso, verdadeiro}\}, i = 1, \dots, N$ .
Procedimento 2 ( $p_2$ )	Operador de Transcrição ( $\Gamma$ )	Se $A(S_i) = \text{"verdadeiro"}$ , o procedimento define como a informação $I$ , transportada pelo vetor, será transferida para o cromossomo.
Procedimento 3 ( $p_3$ )	Bloqueio da informação transcrita ( $\Psi$ )	Torna o resultado da manipulação inviolável por um certo período de tempo – número de iterações, gerações de cromossomos, etc.
Procedimento 4 ( $p_4$ )	Desbloqueio da informação transcrita ( $\bar{\Psi}$ )	Torna o resultado da manipulação sem restrições.
Procedimento 5 ( $p_5$ )	Operador de identificação ( $\Lambda$ )	Identificam uma cadeia ou um trecho da cadeia.
Procedimento 6 ( $p_6$ )	Operador de recombinação ( $\Pi$ )	Recombinam os genes da cadeia.

Em geral, quando um vetor transgenético manipula um cromossomo, modifica sua estrutura e desta forma um novo ponto do espaço de busca é investigado (GOUVÊA e GOLDBARG, 2001). Esta manipulação ocorre de acordo com o procedimento A (ataque), definido em  $p_1$ , que pode ser determinístico ou probabilístico. Um vetor  $\lambda$  transcreve sua informação  $I$  usando o operador de transcrição definido em  $p_2$  somente se  $A(S_i) = \text{"verdadeiro"}$ . Caso contrário, o cromossomo não é manipulado por  $\lambda$  e diz-se que  $S_i$  resistiu ao ataque de  $\lambda$ .

Os vetores  $\lambda = (I, \Phi)$  com  $\Phi = (p_1, p_2, \dots, p_s)$  podem possuir um dos seguintes processos:

- Vírus (V):  $\Phi = (p_1, p_2, p_3, p_4)$ ;
- Plasmídeo (PI):  $\Phi = (p_1, p_2)$ ;
- Transposon (T):  $\Phi = (p_1, p_2, p_5, p_6)$

- Plasmídeo Recombinado (Pr):  $\Phi = (p_1, p_2, p_6)$

As regras de administração (regras transgenéticas) gerenciam o processo de infiltração intra-extracelular. Existem três tipos de regras que administram o relacionamento entre cromossomos e vetores transgenéticos (GOLDBARG, GOLDBARG e MEDEIROS NETO, 2004).

- Regra Tipo 1: Direciona a constituição da cadeia de informação  $I$  que irá ser carregada no vetor transgenético. Pode usar fontes teóricas ou heurísticas para obter o conhecimento para formação do vetor. Podem existir várias regras deste tipo;
- Regra Tipo 2: Define como a cadeia de informação  $I$  será transcrita em um cromossomo ou como o conteúdo do cromossomo será rearranjado.
- Regra Tipo 3: Determina como utilizar as regras do Tipo1 e Tipo2 e gerenciar a simbiose estabelecendo: como escolher um número  $q$  de cromossomos para manipulação; como escolher um número  $r$  de vetores de manipulação; quais as regras Tipo1 e Tipo2 serão utilizadas em cada manipulação; como organizar e obter informações genéticas; como realimentar o processo evolucionário e quando parar a evolução.

Entre os algoritmos existentes na Transgenética Computacional tem-se o algoritmo ProtoG. Este algoritmo é baseado em população na qual os cromossomos não trocam informações de forma direta. Ainda, os cromossomos são expostos a um ataque direto pelos vetores do fluxo intracelular para concretizarem a transcrição de seus códigos.

### 3. MODELAGEM DO PROBLEMA

Neste trabalho foi desenvolvida uma técnica para solução de um problema de *scheduling* de curto termo em um sub-sistema de uma refinaria de petróleo. Em geral, problemas de *scheduling* em refinarias de petróleo são formulados como modelos de otimização inteira mista e resolvidos utilizando-se técnicas exatas, mas o tempo despendido para chegar à solução é inviável para a implementação dos mesmos. O sub-sistema estudado neste trabalho envolve as operações de transferência e estocagem de diesel. São considerados o desenvolvimento e a solução de modelos de otimização para o *scheduling* de um conjunto de operações que incluem: o recebimento do diesel nos tanques, armazenamento e envio deste produto a clientes.

A modelagem desenvolvida para este sistema tem como objetivo a obtenção da seqüência para o fluxo e armazenamento do diesel de forma a atender às restrições de operações e de demandas a um custo mínimo. Os parâmetros envolvidos na modelagem deste problema incluem a configuração do parque de tancagem, as restrições de operação e a demanda de diesel como dados de entrada e o gerenciamento de atividades como dados de saída. Para a resolução do modelo foram utilizadas as seguintes técnicas: *Branch and Bound*, Método Simplex e AG.

As seguintes premissas e restrições operacionais foram consideradas na modelagem do problema:

- Os volumes de diesel estocados nos tanques são conhecidos;
- Os tanques são dedicados, ou seja, armazenam um único tipo de diesel;
- Todos os tanques devem estar disponíveis durante o horizonte de planejamento, ou seja, não devem sofrer manutenção ou qualquer operação que impossibilite seu uso;
- Os tempos de transição entre tarefas não são considerados, por serem desprezíveis em relação às demais operações;
- Cada um dos tanques não pode efetuar operações de carga e descarga simultaneamente;
- Cada tanque pode receber de uma única origem num determinado intervalo de tempo;
- Cada tanque pode enviar para um único destino num determinado intervalo de tempo;

- As restrições operacionais de vazão mínima/máxima de fluxo de recebimento e envio de produtos devem ser respeitadas;
- Os tanques não poderão estar com volume abaixo do mínimo e nem acima do máximo estipulados;
- Quando um tanque iniciar o recebimento de carga, deverá fazê-lo até o enchimento total do mesmo;
- Após o enchimento, o tanque deverá ficar parado por um período mínimo estipulado para repouso e análise do produto estocado. Após este período o tanque estará disponível para envio;
- Todo o lote requerido pelo cliente deve ser bombeado de forma ininterrupta;
- As demandas de recebimento e envio podem ser satisfeitas em qualquer instante do horizonte de planejamento.

#### 4. MODELO COM DISCRETIZAÇÃO UNIFORME DE TEMPO

O primeiro modelo matemático para o problema de transferência e estocagem utiliza a modelagem PLIM com discretização uniforme no tempo. São utilizadas variáveis contínuas, inteiras e binárias. As variáveis binárias representam as decisões a serem tomadas. Eventos de qualquer tipo devem começar e terminar no início de cada intervalo de tempo. Como já citado, o objetivo deste modelo é o de minimizar custos operacionais. As restrições foram construídas respeitando-se os procedimentos operacionais, as restrições físicas do processo e a demanda.

Nomenclatura utilizada no Modelo

Índices:

- $Tq$  - representa o número do tanque:  $tq = 1, 2, \dots, TQ$   
 $C$  - representa o número do cliente:  $c = 1, 2, \dots, C$   
 $T$  - representa o número do intervalo de tempo:  $t = 1, 2, \dots, T$

Conjuntos:

- $CTQ$  - conjunto formado pelos tanques  $tq$   
 $CC$  - conjunto formado pelos clientes  $c$   
 $CT$  - conjunto formado pelos intervalos de tempo  $t$

Vale observar o uso de índices subscritos, para representar os tanques, os clientes ou os intervalos de tempo envolvidos nos parâmetros e variáveis.

Parâmetros:

- $CB_c$  - custo de bombeio para o cliente  $c$  (unidades monetárias/mil  $m^3$ );  
 $CA_{tq}$  - custo de armazenamento no tanque  $tq$  (unidades monetárias/mil  $m^3$ );  
 $CTR_{tq}$  - custo de troca do tanque  $tq$  que recebe da produção (unidades monetárias);  
 $QR_{min}$  - vazão mínima de recebimento pelos tanques  $tq$  (mil  $m^3/h$ );  
 $QR_{max}$  - vazão máxima de recebimento pelos tanques  $tq$  (mil  $m^3/h$ );  
 $QE_{min_c}$  - vazão mínima de envio ao cliente  $c$  (mil  $m^3/h$ );  
 $QE_{max_c}$  - vazão máxima de envio ao cliente  $c$  (mil  $m^3/h$ );  
 $Vol_{min_{tq}}$  - volume mínimo permitido no tanque  $tq$  (mil  $m^3$ );  
 $Vol_{max_{tq}}$  - volume máximo permitido no tanque  $tq$  (mil  $m^3$ );  
 $Vol_{ini_{tq}}$  - volume no tanque  $tq$  no início do processo (mil  $m^3$ );  
 $DEM_c$  - demanda de diesel do cliente  $c$  (mil  $m^3$ ).

Variáveis Binárias:

$$Rec_{tq,t} = \begin{cases} 1, & \text{se o tanque } tq \text{ recebe diesel da produção no intervalo de tempo } t \\ 0, & \text{caso contrário} \end{cases}$$

$$Env_{tq,c,t} = \begin{cases} 1, & \text{se o tanque } tq \text{ envia diesel para o cliente } c \text{ no intervalo de tempo } t \\ 0, & \text{caso contrário} \end{cases}$$

$$XI_{c,t} = \begin{cases} 1, & \text{se o cliente } c \text{ começa a receber diesel no intervalo de tempo } t \\ 0, & \text{caso contrário} \end{cases}$$

$$XF_{c,t} = \begin{cases} 1, & \text{se o cliente } c \text{ termina de receber diesel no intervalo de tempo } t \\ 0, & \text{caso contrário} \end{cases}$$

$$TR_{tq,tq',t} = \begin{cases} 1, & \text{se o tanque } tq \text{ termina de receber e o tanque } tq' \text{ começa a receber no intervalo de tempo } t \\ 0, & \text{caso contrário} \end{cases}$$

Variáveis Contínuas:

$QR_{tq,t}$  - volume de diesel recebido pelo tanque  $tq$  no intervalo de tempo  $t$ ;

$QE_{tq,c,t}$  - volume de diesel enviado pelo tanque  $tq$  ao cliente  $c$  no intervalo de tempo  $t$ ;

$Vol_{tq,t}$  - volume de diesel estocado no tanque  $tq$  no tempo  $t$ ;

$TI_c$  - marca o momento de início do recebimento do cliente  $c$ ;

$TF_c$  - marca o momento de término do recebimento do cliente  $c$ ;

Função Objetivo:

Para este modelo assume-se que a operação ótima do problema de transferência e estocagem é aquela que minimiza os custos operacionais do processo, definidos pelos custos de bombeamento de envio adicionado ao custo de armazenamento de produto nos tanques e, finalmente, adicionado ao custo de transição por troca de tanques na operação de recebimento do processo.

$$Custos = \sum_{tq \in CTQ} \sum_{c \in CC} \sum_{t \in CT} CB_c \cdot QE_{tq,c,t} + \sum_{tq \in CTQ} \sum_{t \in CT} CA_{tq} \cdot Vol_{tq,t} + \sum_{tq \in CTQ} \sum_{tq' \neq tq \in CTQ} \sum_{t \in CT} CTR_{tq} \cdot TR_{tq,tq',t} \quad (1)$$

Restrições: As seguintes restrições são utilizadas no modelo:

1) Restrições de operação:

Um tanque não pode receber e enviar diesel ao mesmo tempo.

$$Rec_{tq,t} + \sum_{c \in CC} Env_{tq,c,t} \leq 1 \quad \forall tq = 1, \dots, TQ; t = 1, \dots, T \quad (2)$$

Haverá sempre um tanque recebendo diesel em cada intervalo de tempo.

$$\sum_{tq \in CTQ} Rec_{tq,t} = 1 \quad \forall t = 1, \dots, T \quad (3)$$

Somente um tanque poderá enviar diesel para um cliente em cada intervalo de tempo.

$$\sum_{tq \in CTQ} Env_{tq,c,t} \leq 1 \quad \forall c = 1, \dots, C; t = 1, \dots, T \quad (4)$$

Quando um cliente iniciar seu recebimento de diesel, deverá fazê-lo sem interrupção.

$$\sum_{i \in CT} XI_{c,t} \leq 1 \quad \forall c = 1, \dots, C \quad \text{e} \quad \sum_{i \in CT} (XI_{c,t} - XF_{c,t}) = 0 \quad \forall c = 1, \dots, C \quad (5) \text{ e } (6)$$

O momento de início e término de bombeio são armazenados nas variáveis auxiliares  $TI_c$  e  $TF_c$ .

$$\sum_{i \in CT} (t \cdot XI_{c,t}) = TI_c \quad \forall c = 1, \dots, C \quad \text{e} \quad \sum_{i \in CT} (t \cdot XF_{c,t}) = TF_c \quad \forall c = 1, \dots, C \quad (7) \text{ e } (8)$$

## 2) Restrições de Fluxo:

O fluxo de recebimento deve estar entre o fluxo mínimo e máximo estipulados se a variável  $Rec_{iq,t}$  for igual a “1”. Se o valor da variável  $Rec_{iq,t}$  for “0”, o fluxo  $QR_{iq,t}$  também será igual a “0”.

$$QR_{min} \cdot Rec_{iq,t} \leq QR_{iq,t} \leq QR_{max} \cdot Rec_{iq,t} \quad \forall tq = 1, \dots, TQ; t = 1, \dots, T \quad (9)$$

O fluxo de envio deve estar entre o fluxo mínimo e máximo estipulados se a variável  $Env_{iq,c,t}$  for igual a “1”. Se o valor da variável  $Env_{iq,c,t}$  for “0”, o fluxo  $QE_{iq,c,t}$  também será igual a “0”.

$$QE_{min_c} \cdot Env_{iq,c,t} \leq QE_{iq,c,t} \leq QE_{max_c} \cdot Env_{iq,c,t} \quad \forall tq = 1, \dots, TQ; c = 1, \dots, C; t = 1, \dots, T \quad (10)$$

## 3) Balanço de Massa:

O volume de um tanque num certo intervalo de tempo deve ser igual ao volume inicial, mais o volume recebido do processo menos o volume enviado a clientes até este intervalo de tempo.

$$Vol_{iq,t} = Volini_{iq} + \sum_{\substack{t' \in CT \\ t' \leq t}} \left( QR_{iq,t'} - \sum_{\substack{c \in CC \\ t' \leq t}} QE_{iq,c,t'} \right) \quad \forall tq = 1, \dots, TQ; t = 1, \dots, T \quad (11)$$

O volume dos tanques deve estar sempre entre o volume mínimo e o máximo determinados.

$$Vol_{min_{iq}} \leq Vol_{iq,t} \leq Vol_{max_{iq}} \quad \forall tq = 1, \dots, TQ; t = 1, \dots, T \quad (12)$$

## 4) Restrição de Demanda: A demanda dos clientes deverá ser cumprida em sua totalidade.

$$\sum_{tq \in CTQ} \sum_{i \in CT} QE_{iq,c,t} = DEM_c \quad \forall c = 1, \dots, C \quad (13)$$

## 5) Restrições de Transição:

A transição ocorre quando num intervalo de tempo um certo tanque está recebendo do processo e no intervalo seguinte um outro tanque é que recebe.

$$TR_{iq,tq',t} \leq Rec_{iq,t-1} \quad \forall t = 2, \dots, T; tq, tq' = 1, \dots, TQ, tq \neq tq' \quad (14)$$

$$TR_{iq,tq',t} \leq Rec_{iq',t} \quad \forall t = 2, \dots, T; tq, tq' = 1, \dots, TQ, tq \neq tq' \quad (15)$$

$$TR_{iq,tq',t} \geq Rec_{iq,t-1} + Rec_{iq',t} - 1 \quad \forall t = 2, \dots, T; tq, tq' = 1, \dots, TQ, tq \neq tq' \quad (16)$$

As restrições a seguir fazem com que a variável  $XI_{c,t}$  assuma valor “1” se no intervalo (t-1) o cliente não está recebendo e passa a receber no intervalo t e assume valor “0” para qualquer outra situação.

$$XI_{c,t} \leq \sum_{tq \in CTQ} Env_{iq,c,t} \quad \forall t = 2, \dots, T; c = 1, \dots, C \quad (17)$$

$$XI_{c,t} \leq 1 - \sum_{tq \in CTQ} Env_{iq,c,t-1} \quad \forall t = 2, \dots, T; c = 1, \dots, C \quad (18)$$

$$XI_{c,t} \geq \sum_{tq \in CTQ} Env_{iq,c,t} - \sum_{tq \in CTQ} Env_{iq,c,t-1} \quad \forall t = 2, \dots, T; c = 1, \dots, C \quad (19)$$



As restrições a seguir fazem com que a variável  $XF_{c,t}$  assumam valor “1” se no intervalo  $(t - 1)$  o cliente está recebendo e passa a não receber no intervalo  $t$  e assume valor “0” para qualquer outra situação.

$$XF_{c,t} \leq \sum_{tq \in CTQ} Env_{tq,c,t-1} \quad \forall t = 2, \dots, T; c = 1, \dots, C \quad (20)$$

$$XF_{c,t} \leq 1 - \sum_{tq \in CTQ} Env_{tq,c,t} \quad \forall t = 2, \dots, T; c = 1, \dots, C \quad (21)$$

$$XF_{c,t} \geq \sum_{tq \in CTQ} Env_{tq,c,t-1} - \sum_{tq \in CTQ} Env_{tq,c,t} \quad \forall t = 2, \dots, T; c = 1, \dots, C \quad (22)$$

As restrições (17) a (22) não contemplam o primeiro intervalo de tempo. Se no início do período algum tanque estiver bombeando, este será considerado o início do bombeio. Nenhum bombeio poderá encerrar no primeiro intervalo.

$$XI_{c,1} = \sum_{tq \in CTQ} Env_{tq,c,1} \quad \forall c = 1, \dots, C \quad \text{e} \quad XF_{c,1} = 0 \quad \forall c = 1, \dots, C \quad (23) \text{ e } (24)$$

## 5. PROTOG – PLIM: MODELAGEM E METODOLOGIA

O modelo utilizado para o problema é gerado a partir do modelo de PLIM com discretização uniforme do tempo (Seção 4). Para a resolução, este modelo em PLIM é modificado para ser resolvido por PL. Esta modificação inclui a retirada de algumas restrições e variáveis binárias, e relaxação linear das variáveis binárias e inteiras restantes. Algumas das variáveis retiradas do modelo são tratadas por com Algoritmo ProtoG e outras através da aplicação de um procedimento. Os valores encontrados são inseridos como dados no modelo em PL, que é então resolvido pelo aplicativo LINGO 8.0. O valor da função objetivo resultante é utilizado como valor da função de aptidão para o algoritmo ProtoG. Depois de terminado o processo do algoritmo ProtoG, o resultado encontrado é inserido como dados no modelo em PLIM. Este modelo é então resolvido pelo LINGO 8.0 usando a técnica de *Branch and Bound*.

As etapas para o desenvolvimento do processo mostrado na Figura 1, foram separadas em oito módulos, descritos a seguir:

Módulo 1: Construtor de indivíduos

Este módulo gera aleatoriamente vetores compostos pelos dígitos binários “0” e “1” para formar os indivíduos da população. Estes dígitos representam os valores das variáveis  $Rec_{tq,t}$ ,  $tq = 1, \dots, TQ$ ,  $t = 1, \dots, T$ . Um procedimento foi desenvolvido para gerar estes cromossomos de forma a satisfazer a restrição (3) que impõe que haverá sempre um, e somente um tanque recebendo da produção em cada intervalo de tempo. Logo, para a construção de um indivíduo deve-se ter em cada intervalo de tempo  $t \in \{1, \dots, T\}$ , um e somente um dos tanques  $tq \in \{1, \dots, TQ\}$  com valor da variável binária igual a “1”. A Figura 2 ilustra o exemplo de um cromossomo gerado pelo Módulo 1 com  $T = 5$  e  $TQ = 4$ .

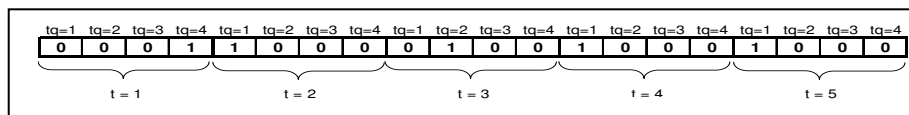


Figura 1: Exemplo de cromossomo

Módulo 2: Procedimento para encontrar as variáveis binárias  $TR_{tq,t}$

Como estas variáveis dependem exclusivamente das variáveis  $Rec_{tq,t}$ , um procedimento foi implementado para encontrá-las, satisfazendo as restrições (14), (15) e (16).

O procedimento verifica, para todas as combinações duas a duas dos tanques, se as variáveis  $Rec_{tq,t}$  no intervalo de tempo  $t$  e  $(t - 1)$  são iguais a “1”. Em caso afirmativo, a variável  $TR_{tq,tq',t}$  assumirá valor “1” no intervalo  $t$ . Para qualquer outro caso, a variável assume valor “0”. Esta verificação é feita para os intervalos de tempo  $t \in \{2, \dots, T\}$ .

### Módulo 3: Avaliador dos indivíduos

Este módulo é composto pelas etapas: extração dos dados do cromossomo (variáveis  $Rec_{tq,t}$ ,  $tq = 1, \dots, TQ$ ,  $t = 1, \dots, T$ ), aplicação do Módulo 2 para gerar as variáveis  $TR_{tq,tq',t}$ , inserção dos dados obtidos no modelo do LINGO 8.0 em PL, resolução deste modelo e obtenção do valor da função objetivo para ser utilizado como valor da função aptidão.

### Módulo 4: Ordenador da população

Ordena os indivíduos, no caso de minimização, em ordem crescente de acordo com o valor da função de aptidão. Após operações do algoritmo ProtoG, se um novo cromossomo vier a ser inserido na população, este módulo será acionado para reordenar a população.

### Módulo 5: Construtor da população inicial com $N$ indivíduos

Para formar este módulo, foram utilizados os Módulos 1, 3 e 4 da seguinte forma. Gera-se um indivíduo usando-se o construtor de indivíduos (Módulo 1). Avalia-se este indivíduo utilizando o Módulo 3. Se este indivíduo é viável e diferente de todos os indivíduos já na população, então aplica-se o Módulo 4 para fazer a inserção do indivíduo, respeitando-se a ordenação. Este procedimento é efetuado até que a população tenha o número de indivíduos desejados.

### Módulo 6: Construtor das partículas genéticas móveis

Para formar as partículas genéticas móveis  $\lambda = (I, \Phi)$ , deve-se definir a cadeia de informação  $I$  e o método de manipulação  $\Phi$ , com  $\Phi = \{p_1, p_2\}$ . O procedimento  $p_1$  estabelece quando um cromossomo é suscetível à manipulação ou ataque pelas partículas genéticas móveis. Na aplicação deste trabalho, um cromossomo é suscetível à manipulação se a operação promover melhoria no valor de aptidão. Já o procedimento  $p_2$ , em caso de ataque, define como a informação  $I$ , que é transportada pela partícula genética móvel, é transferida para o cromossomo. Como citado na Regra Tipo 1, a constituição da cadeia de informação  $I$  que será carregada na partícula genética móvel pode fazer uso de conhecimento sobre o problema. Neste caso, sabe-se que o número de trocas entre tanques é reduzido se cada tanque que recebe da produção o faz por vários intervalos de tempo consecutivos. Portanto, a cadeia  $I$  para este problema será formada por um certo número de intervalos consecutivos com o mesmo tanque recebendo da produção. O número de intervalos consecutivos é considerado como parâmetro variável do modelo. A Figura 3 ilustra uma cadeia  $I$  com número de intervalos  $T = 3$  e  $TQ = 4$ .

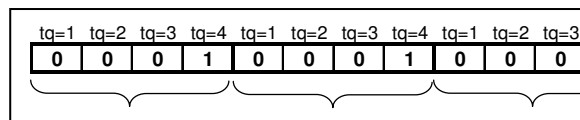


Figura 2: Exemplo de cadeia  $I$  para o algoritmo ProtoG

A Regra Tipo 2 define como a cadeia de informação é transcrita em um cromossomo. Para esta aplicação, o operador de transcrição primeiramente gera aleatoriamente um ponto de corte. A seguir retira do cromossomo, a partir do corte, uma cadeia de mesmo comprimento da cadeia  $I$ . A cadeia  $I$  é então inserida na mesma posição da cadeia retirada.

## Módulo 7: Algoritmo ProtoG

Este módulo é constituído pelo Algoritmo Transgenético ProtoG propriamente dito. Utiliza o resultado obtido pelo Módulo 5 para dados de entrada e os Módulos 3, 4 na execução do algoritmo. Fazendo uso da Regra Tipo 3, define-se o tamanho da sub-população a ser extraída da população. O operador de seleção utilizado para a escolha dos indivíduos da sub-população foi o proposto por Mayerle (1994) através da fórmula (25).

$$Select(R) = \left\{ r_j \in R / j = m + 1 - \left\lceil \frac{-1 + \sqrt{1 + 4 \cdot rnd \cdot (m^2 + m)}}{2} \right\rceil \right\} \quad (25)$$

onde:  $R$  é o conjunto dos  $m$  cromossomos;  $r_j$  é o  $j$ -ésimo cromossomo;  $rnd \in [0,1)$  é um número aleatório uniformemente distribuído;  $\lceil x \rceil$  é a função que retorna o menor inteiro maior que  $x$ .

A partícula genética móvel  $\lambda = (I, \Phi)$  que fará o ataque a um determinado indivíduo da população é escolhida aleatoriamente entre todas as construídas pelo Módulo 6. Um parâmetro representando o número máximo de iterações do algoritmo é utilizado como critério de parada.

O algoritmo transgenético ProtoG utilizado neste módulo e seus passos estão descritos a seguir:

- Carregue\_Regras\_Transgenéticas (*Tipo1, Tipo2, Tipo3, q, r*): Determina como as partículas genéticas móveis serão formadas, informando o conjunto de regras de administração que serão utilizadas. Define também o número de partículas genéticas ( $r$ ) a serem formadas e quantos serão os indivíduos da sub-população ( $q$ );
- Construir\_Partículas\_Genéticas\_Móveis ( $r$ ): gera  $r$  partículas genéticas móveis de acordo com as regras definidas no Módulo 6;
- Escolha\_ $\lambda$ : escolhe aleatoriamente uma das  $r$  partículas genéticas móveis;
- Escolha\_ $S$ : escolhe os indivíduos (cromossomos) para formar a sub-população;
- ataque ( $S_j, \lambda$ ): implementa o procedimento  $p_1$ .
- manipular ( $S_j, \lambda$ ): Caso o ataque seja permitido, a partícula  $\lambda$  efetua a manipulação  $p_2$  no cromossomo  $S_j$ ;
- critério\_parada: o número máximo de iterações  $itMax$  foi o critério de parada utilizado nesta aplicação.

**Algoritmo ProtoG**

**Gerar** população ( $S_1, S_2, \dots, S_N$ )

Carregue\_Regras\_Transgenéticas (*Tipo1, Tipo2, Tipo3, q, r*)

Construir\_Partículas\_Genéticas\_Móveis ( $r$ )

**Repita**

Escolha\_ $S$  = seleção\_população ( $S_1, S_2, \dots, S_N, q$ )

**Para**  $j = 1$  até  $q$ , **faça**

Escolha\_ $\lambda$  = Partícula\_genética\_móvel ( $r$ );

**Se** ataque ( $S_j, \lambda$ ) é verdadeiro, **então**

$S_j$  = manipular ( $S_j, \lambda$ )

Inserir  $S_j$  na população

**fim**\_Se

**fim**\_Para\_ $j$

**até** critério\_parada ser satisfeito

**Fim**

## Módulo 8: Obtenção do resultado final

O melhor indivíduo obtido ao final da execução do Módulo 7 é utilizado para gerar as variáveis  $Rec_{tq,t}$ . Através do Módulo 2, obtém-se as variáveis  $TR_{tq,tq',t}$ . Estas variáveis são inseridas no modelo PLIM sem as restrições (3), (14), (15) e (16) e o problema é resolvido pelo LINGO 8.0.

## 6. IMPLEMENTAÇÃO E RESULTADOS

Nesta seção são apresentados os resultados dos testes obtidos através da implementação computacional das metodologias descritas. Estes resultados são apresentados para os modelos com representação de tempo discreto com a implementação das metodologias: algoritmo *Branch and Bound* com o uso do aplicativo LINGO 8.0 e ProtoG-PLIM. Os programas computacionais implementados foram desenvolvidos com o aplicativo Visual Basic 6.0 (VB 6.0) na versão Enterprise Edition. Na implementação da abordagem AGEEH – PLIM foram utilizados dois módulos em VB, fornecidos pela LINDO Systems Inc., para integração entre VB 6.0 e LINGO 8.0.

Para o modelo com discretização uniforme de tempo, foram efetuados dois grupos de testes distintos para a obtenção de resultados: resolução do modelo em PLIM utilizando o aplicativo LINGO 8.0 e aplicação da metodologia ProtoG – PLIM.

Os dados utilizados para estes grupos de testes foram:

- Número de tanques:  $TQ = 4$ ;
- Número de clientes:  $C = 2$ ;
- Número de intervalos de tempo:  $T = 24$ ;
- Custo de bombeio para os clientes:  $CB_1 = 0,15$  unidades monetárias/mil m<sup>3</sup> e  $CB_2 = 0,2$  unidades monetárias /mil m<sup>3</sup> de produto enviado;
- Custo de armazenamento nos tanques:  $CA_{tq} = 0,01$  unidades monetárias/mil m<sup>3</sup> de produto estocado em qualquer tanque;
- Custo de troca:  $CTR_{tq} = 2,00$  unidades monetárias para cada troca efetuada entre tanques na operação de recebimento;
- Vazão mínima de recebimento pelos tanques:  $QRmin = 0,6$  mil m<sup>3</sup>/h;
- Vazão máxima de recebimento pelos tanques:  $QRmax = 0,7$  mil m<sup>3</sup>/h;
- Vazão mínima de envio ao cliente  $c$ :  $QEmin_1 = 0,5$  mil m<sup>3</sup>/h e  $QEmin_2 = 0,9$  mil m<sup>3</sup>/h;
- Vazão máxima de envio ao cliente  $c$ :  $QEmax_1 = 0,6$  mil m<sup>3</sup>/h e  $QEmax_2 = 1$  mil m<sup>3</sup>/h;
- Volume mínimo permitido nos tanques:  $Volmin_{tq} = 1$  mil m<sup>3</sup> para todos os tanques;
- Volume máximo permitido nos tanques:  $Volmax_{tq} = 16$  mil m<sup>3</sup> para todos os tanques;
- Volume no tanque  $tq$  no início do processo:  $Volini_1 = 7$  mil m<sup>3</sup>,  $Volini_2 = 1$  mil m<sup>3</sup>,  $Volini_3 = 1$  mil m<sup>3</sup> e  $Volini_4 = 1$  mil m<sup>3</sup>;
- Demanda de diesel para cada cliente  $c$ :  $DEM_1 = 5$  mil m<sup>3</sup> e  $DEM_2 = 6$  mil m<sup>3</sup>;

### 6.1 RESULTADOS PARA A MODELAGEM PLIM COM REPRESENTAÇÃO DE TEMPO DISCRETO

Os dados acima aplicados ao modelo PLIM, resultaram em 390 variáveis contínuas, 766 inteiras e um total de 2149 restrições. O modelo foi executado 25 vezes pelo LINGO 8.0, até encontrar-se o valor ótimo igual a 6,285 unidades monetárias. A Tabela 1 mostra a média e o desvio padrão para o número de iterações efetuadas pelo LINGO 8.0 e o tempo despendido no processo.

Tabela 2: Resultados para o modelo PLIM

Estatística	Nº de Iterações	Tempo Computacional (s)
Média	743124,8	1359,4
Desvio Padrão	273218,69	480,52

## 6.2 RESULTADOS PARA A METODOLOGIA PROTOG – PLIM

Para a obtenção dos resultados para a metodologia algoritmo ProtoG - PLIM foram efetuadas 405 rodadas de testes, variando-se os parâmetros: tamanho da população, tamanho da sub-população e tamanho da partícula genética móvel. Os dados referentes ao número de tanques, clientes e intervalos de tempo devem ser fornecidos pelo usuário. Os resultados gerados pelo programa para a fase de resolução do Algoritmo ProtoG são: número de iterações do algoritmo ProtoG, número de iterações do LINGO, tempo computacional e valor de aptidão (*fitness*). Em cada teste foram salvos todos os indivíduos com valor da função de aptidão menor que 6,9, sendo este valor 10% maior que o valor ótimo (resolução em PL) de 6,266523 unidades monetárias.

Os resultados das rodadas de testes foram sintetizados para populações com tamanho igual a 20, 30 e 40 indivíduos. Os outros parâmetros variáveis que foram combinados com o tamanho da população são:

- Tamanho da Sub-população: 30% do tamanho da população;
- Tamanho da cadeia do vetor transgenético (partícula genética móvel): 5, 6 e 7;

Na Tabela 3 são apresentadas estatísticas descritivas do número de iterações do algoritmo ProtoG, do número de iterações do LINGO, do tempo computacional e da função objetivo resultantes dos 405 resultados obtidos.

Tabela 3: Estatísticas descritivas dos testes da metodologia algoritmo ProtoG – PLIM

Intervalos da função $f$	Nº de Iterações do ProtoG	Nº de Iterações do LINGO	Tempo Computacional (s)	Valor da Função de Aptidão
$f = 6,285$	$147 \pm 86,5$	$753340 \pm 464587$	$888 \pm 537$	$6,285 \pm 0$
$6,285 < f \leq 6,47$	$108 \pm 62$	$565021 \pm 360228$	$660 \pm 375$	$6,354 \pm 0,053$
$6,47 < f \leq 6,66$	$85 \pm 51$	$442470 \pm 252712$	$515 \pm 288$	$6,521 \pm 0,025$
$6,66 < f \leq 6,9$	$74 \pm 50$	$399271 \pm 253122$	$463 \pm 284$	$6,77 \pm 0,069$

Média  $\pm$  desvio padrão

A Tabela 4 mostra as médias do número de iterações do Algoritmo ProtoG, do número de iterações do LINGO e do tempo computacional para cada combinação dos parâmetros tamanho da população, tamanho da sub-população e tamanho da partícula genética móvel, quando o resultado ótimo igual a 6,285 foi encontrado.

Tabela 4: Média para o resultado ótimo da metodologia Algoritmo ProtoG – PLIM

Tamanho População	Tamanho Sub-População	Tamanho da Partícula Genética Móvel	Nº de Iterações do ProtoG	Nº de Iterações do LINGO	Tempo Computacional (segundos)
20	6	5	207	706078	804,4
20	6	6	107	382817	467,8
20	6	7	123	445386	539,9
30	9	5	188	973764	1131,5
30	9	6	147	781346	929,7
30	9	7	111	616410	745,9
40	12	5	205	1414862	1614,7
40	12	6	119	837106	1006,5
40	12	7	73	535616	655,3

## 7. ANÁLISE DOS RESULTADOS E CONCLUSÕES

O processo de transferência e estocagem de óleo diesel da refinaria de petróleo estudado foi analisado de forma a obter-se um modelo com representação discreta no tempo, com características lineares. Por ser um modelo em PLIM com discretização do tempo, um problema que surge na resolução através do aplicativo LINGO 8.0 é que a redução do tamanho do intervalo de tempo implica em um número de intervalos (T) maior e conseqüentemente um número maior de variáveis binárias, fazendo com que o tempo computacional aumente. Este acréscimo no tempo se deve ao fato do problema ser do tipo combinatorial. Neste caso, aumentando-se o número de variáveis inteiras, aumenta-se o número de combinações possíveis para o vetor solução, o que pode tornar o problema inviável, em termos de tempo computacional, na resolução com o algoritmo *Branch and Bound*.

A Tabela 5 exibe as médias das iterações do LINGO 8.0 e do tempo computacional das metodologias PLIM e ProtoG-PLIM de acordo com os intervalos da função de aptidão (f). Com base nestes dados foram efetuadas comparações entre as duas metodologias em relação ao desempenho.

Tabela 5: Resumo das médias para as metodologias PLIM e ProtoG – PLIM

Intervalos da função f	Metodologia	Iterações do LINGO	Tempo Computacional (segundos)
$f = 6,285$ (resultado ótimo)	PLIM	743125	1359,4
	ProtoG – PLIM	753340	888
$6,285 < f \leq 6,47$	ProtoG – PLIM	565021	660
$6,47 < f \leq 6,66$	ProtoG – PLIM	442470	515
$6,66 < f \leq 6,9$	ProtoG – PLIM	399271	463

A partir dos dados resumidos na Tabela 5 pode-se comparar os resultados das metodologias PLIM e ProtoG - PLIM, da seguinte forma: se o resultado ótimo para o valor da função de aptidão (f) for alcançado na metodologia ProtoG – PLIM, tem-se um número de iterações apenas 1,4% maior com tempo computacional 34,7% menor. Se  $6,285 < f \leq 6,47$ , a metodologia ProtoG – PLIM resulta num número de iterações 24% menor com tempo computacional 51,4% menor. Se  $6,47 < f \leq 6,66$ , a metodologia ProtoG – PLIM resulta num

número de iterações 40,4% menor com tempo computacional 62,1% menor. Ainda, se  $6,66 < f \leq 6,9$ , a metodologia ProtoG – PLIM resulta num número de iterações 46,3% menor com tempo computacional 65,9% menor

De modo geral observa-se que o desempenho da metodologia ProtoG – PLIM com relação ao tempo computacional foi melhor. Conclui-se, portanto, que deve ser analisada a vantagem em reduzir-se o número de iterações do LINGO e tempo computacional em detrimento do aumento no valor da função de aptidão.

Para os testes realizados neste trabalho, analisando-se os resultados da Tabela 4 observa-se que o melhor desempenho em termos de número de iterações do LINGO e tempo computacional ao atingir o resultado ótimo, foi observado para tamanho de população igual a 20, tamanho da sup-população igual a 6 e tamanho da partícula genética móvel igual a 6.

Finalmente, conclui-se que a metodologia ProtoG-PLIM pode contribuir significativamente para o problema de transferência e estocagem de produtos em refinarias de petróleo. Os resultados obtidos mostraram que a metodologia é adequada, levando-se em conta a redução do tempo computacional, sem grande perda na qualidade da solução.

## 8. REFERÊNCIAS

- BARBOZA, A. O. Simulação e Técnicas da Computação Evolucionária Aplicadas a Problemas de Programação Linear Inteira Mista. Tese de Doutorado. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, 2005.
- DANTZIG, G. B. Linear Programming and Extensions. Princeton University Press, Princeton, New Jersey, United States of America, 1963.
- GOLDBARG, M. C., GOLDBARG, E. F. G., MEDEIROS NETO, F. D. An Evolutionary Approach for the Piston Pump Mobile Unit Problem. Proceedings of MCO'04 Fifth International Conference on Computer Sciences. V. 1, p. 281-290, 2004.
- GOLDBARG, M. C., GOLDBARG, E. F. G., MEDEIROS NETO, F. D. Algoritmos evolucionários na determinação da configuração de custo mínimo de sistemas de co-geração de energia com base no gás natural. Pesquisa Operacional. v. 25, n. 2. p. 231-253, 2005.
- GOUVÊA, E. F., GOLDBARG, M. C. An Intracellular Transgenetic Algorithm applied to the Quadratic Assignment Problem. In: IV Congresso Chileno de Investigación Operativa – anais do IV Congresso Chileno de Investigación Operativa, v. 1 p. 39-45, 2001.
- MAYERLE, S. F. Um Algoritmo Genético para Solução do Problema do Caixeiro Viajante. Artigo de circulação interna do departamento de Engenharia de Produção e sistemas da UFSC, 1994.
- MORO, L. F. L. Técnicas de Otimização Mista-Inteira para o Planejamento e Programação de Produção em Refinarias de Petróleo. Tese de Doutorado. Escola Politécnica da Universidade de São Paulo, SP, Brasil, 2000.
- MOROWITZ, H. Beginning of Cellular Life. New Haven, Conn, Yale University Press, 1992.
- TAHA, H. A. Integer Programming. Theory, Applications, and Computations. Academic Press, Inc. Orlando, Florida, United States of America, 1975.
- ZIONTS, S. Linear and Integer Programming. Prentice-Hall, Inc, Englewood Cliffs, New Jersey, United States of America, 1974.