

KALIANNE ALMEIDA ROSA

**AVALIANDO ALGORITMOS GENÉTICOS COM DIFERENTES
ESTRUTURAS POPULACIONAIS EM UM PROBLEMA INTEGRADO
DE DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DA
PRODUÇÃO**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Prof. Claudio Fabiano Motta Toledo

LAVRAS
MINAS GERAIS – BRASIL
2009

KALIANNE ALMEIDA ROSA

**AVALIANDO ALGORITMOS GENÉTICOS COM DIFERENTES
ESTRUTURAS POPULACIONAIS EM UM PROBLEMA INTEGRADO
DE DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DA
PRODUÇÃO**

Aprovada em 26 de Novembro de 2009

Prof. Me. Cristiano Leite de Castro

Prof. Dr. Ricardo Silveira Souza

Prof. Dr. Claudio Fabiano Motta Toledo
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2009

*Dedico esse trabalho aos meus pais, Messias e Vanda, e minha irmã Djamila
por sempre terem me apoiado em todos os momentos da minha vida,
proporcionando que essa conquista fosse realizada. Amo vocês!*

AGRADECIMENTOS

Agradeço primeiramente a Deus por sua grandeza, infinita bondade e compaixão, sem o qual nada seria possível.

Aos meus pais e minha irmã por todo apoio e amor incondicional. Por serem tão solidários e afetuosos, por sempre acreditarem em mim e estarem presentes em cada momento dessa conquista.

Ao meu cunhado Marcelo, pelo carinho e a amizade.

Ao meu grande amigo Rogger, pessoa com quem convivi dias felizes e dias tristes, amigo de apenas 3 anos e alguns meses, porém sua amizade é única, sempre estarei torcendo por você, onde quer que esteja.

Agradeço a minha turma da computação, meus amigos e companheiros de trabalho, por tudo.

A todos os meus amigos que engrandecem minha vida, fazendo com que as dores passem rapidamente e as alegrias se eternizem em cada sorriso.

Ao meu orientador, professor e amigo Claudio Fabiano Motta Toledo, por toda a paciência, disponibilidade e principalmente por me apoiar e auxiliar na construção desse trabalho.

Aos demais professores e funcionários do Departamento de Ciência da Computação da UFLA.

Enfim, agradeço de coração a cada um que participou desta conquista. Só enquanto eu respirar vou me lembrar de vocês!

**AVALIANDO ALGORITMOS GENÉTICOS COM DIFERENTES
ESTRUTURAS POPULACIONAIS EM UM PROBLEMA INTEGRADO
DE DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DA
PRODUÇÃO**

RESUMO

Algoritmos Genéticos foram avaliados utilizando diferentes estruturas populacionais e operadores genéticos de recombinação na resolução do Problema Integrado de Dimensionamento de Lotes e Programação da Produção. Estruturas uni e multipopulacionais com indivíduos organizados hierarquicamente ou não em árvores, foram aplicadas em instâncias encontradas em fábricas de refrigerantes. Nestas indústrias, o processo de produção é separado em dois níveis interdependentes envolvendo tanques, que armazenam as matérias-primas das bebidas, e linhas de produção, onde as bebidas são engarrafadas. O dimensionamento e a programação dos lotes de matérias-primas em tanques e das bebidas nas linhas devem ser estabelecidos simultaneamente. Praticamente todas as abordagens avaliadas apresentaram melhores resultados que as soluções estimadas pela fábrica, com destaque para as abordagens multipopulacionais hierarquicamente estruturadas em árvores.

Palavras-chave: Algoritmo Genético; estruturas populacionais; dimensionamento de lotes; programação da produção.

***EVALUATING GENETIC ALGORITHMS WITH DIFFERENT
POPULATION STRUCTURES ON A INTEGRATED LOT SIZING AND
SCHEDULING PROBLEM***

ABSTRACT

Genetic Algorithm were evaluated using different population structures and genetic operators of recombination to solve the Integrated Lot Sizing and Scheduling Problem. Uni and multi-population structures with individuals hierarchically organized or not in trees, were applied in instances found in soft drink companies. In these industries the production process has two interdependent levels, one concerning tanks which storage raw material and other concerning production lines where the beverage is bottling. The lot sizing and scheduling of raw materials in tanks and beverages in lines have to be simultaneously determined. Almost all approaches presented better results than estimated cost by factory, standing out the multi-population approaches hierarchically structured in trees.

Keywords: *Genetic Algorithm; population structures; lot sizing, scheduling.*

SUMÁRIO

LISTA DE FIGURAS.....	viii
LISTA DE TABELAS	x
1 INTRODUÇÃO	1
1.1 Contextualização.....	1
1.2 Objetivos do Trabalho.....	3
1.3 Estrutura do Trabalho.....	4
2 REFERENCIAL TEÓRICO	5
2.1 Introdução	5
2.2 Produção de Bebidas no Brasil	5
2.3 Conceitos Básicos	9
2.4 Problemas de Programação da Produção	11
2.4.1 Tipos de Problema.....	11
2.4.2 Métodos de Resolução	13
2.5 Estudo de Caso.....	14
2.6 Trabalhos Relacionados ao PIDLPP	20
2.7 Algoritmos Genéticos	23
2.7.1 Algoritmo Genético Canônico.....	23
2.7.2 Algoritmo Genético Multipopulacional	26
2.7.3 Algoritmos Genéticos Híbridos.....	28
3 MÉTODO DE RESOLUÇÃO	29
3.1 Introdução	29
3.2 Algoritmo Genético Multipopulacional	29
3.3 Representação dos Indivíduos.....	32
3.3.1 Codificação e Inicialização	32
3.3.2 Decodificação.....	36
3.4 Operadores Genéticos	40
3.4.1 Recombinação	40
3.4.2 Mutação.....	43
3.5 Estruturas Populacionais	45

3.5.1	População Não Estruturada	45
3.5.2	População Estruturada em Hierarquia	46
3.5.3	Abordagens Uni e Multipopulacional	49
4	RESULTADOS COMPUTACIONAIS	51
5	CONSIDERAÇÕES FINAIS	60
5.1	Conclusões	60
5.2	Contribuições	61
5.3	Trabalhos Futuros	62
	REFERÊNCIAS BIBLIOGRÁFICAS	63

LISTA DE FIGURAS

Figura 2.1 - Volume de Venda de Refrigerante no Brasil	7
Figura 2.2 - Participação do Volume de Vendas em 2008.....	8
Figura 2.3 - Divisão dos Problemas de Programação da Produção	11
Figura 2.4 - Processo de Produção de Bebidas	15
Figura 2.5 - Programação Não Sincronizada da Produção	17
Figura 2.6 - Programação Sincronizada da Produção	18
Figura 2.7 - Novas Restrições ao Processo de Produção	22
Figura 2.8 - Algoritmo Genético Canônico	26
Figura 3.1 - Algoritmo Genético Multipopulacional	30
Figura 3.2 - Migração	32
Figura 3.3 - Indivíduo	33
Figura 3.4 - Inicialização dos Indivíduos.....	35
Figura 3.5 - Exemplo de Dois Indivíduos Diferentes	36
Figura 3.6 - Decodificação do Primeiro Gene em T_2	37
Figura 3.7 - Decodificação do Segundo Gene em T_2	38
Figura 3.8 - Decodificação do Primeiro Gene em T_1	38
Figura 3.9 - Decodificação do Segundo Gene em T_1	39
Figura 3.10 - Decodificação do Terceiro Gene em T_1	40
Figura 3.11 - <i>Crossover</i> Uniforme.....	41
Figura 3.12 - <i>Crossover</i> de Um Ponto	42
Figura 3.13 - Primeiro Tipo de Mutação	43
Figura 3.14 - Segundo Tipo de Mutação	44
Figura 3.15 - Terceiro Tipo de Mutação.....	45
Figura 3.16 - População Não Estruturada	46
Figura 3.17 - Estruturas com <i>Clusters</i> de Tamanho 3 e 4.....	46
Figura 3.18 - População Antes da Inserção do Novo Indivíduo	47

Figura 3.19 - População Após Inserção do Novo Indivíduo.....	48
Figura 3.20 - Restauração da População – Passo 1.....	48
Figura 3.21 - Restauração da População – Passo 2.....	49
Figura 3.22 - Execução da Migração Entre Populações	50
Figura 4.1 - Gráfico das Médias por Quantidade de Indivíduos	54
Figura 4.2 - Gráfico das Melhores Soluções com População Não Estruturada ..	55
Figura 4.3 - Gráfico das Melhores Soluções com População Estruturada.....	56
Figura 4.4 - Distribuição das Vitórias por Tipo de <i>Crossover</i>	58
Figura 4.5 - Média das Soluções.....	59

LISTA DE TABELAS

Tabela 3.1 - Demandas dos Produtos.....	35
Tabela 3.2 - Demandas dos Indivíduo 1	44
Tabela 4.1 - Parâmetros das instâncias industriais.....	51
Tabela 4.2 - Resultados com 60 Indivíduos.....	53
Tabela 4.3 - Resultados com 120 Indivíduos.....	53
Tabela 4.4 - Resultados com 180 Indivíduos.....	53
Tabela 4.5 - Resultados de S1 e S2.....	57
Tabela 4.6 - Resultados de S3 e S4.....	58

1 INTRODUÇÃO

1.1 Contextualização

O Brasil tem se tornado um grande produtor de bebidas nos últimos anos com destaque para a produção de refrigerantes, que vem aumentando fortemente desde a década de 90. O país possui um volume anual de consumo maior do que 14 bilhões de litros ficando atrás apenas dos Estados Unidos e do México. O setor é disputado tanto por empresas nacionais quanto por multinacionais. As empresas nacionais estão aumentando sua parcela no mercado graças a sua competitividade, quando o assunto é o preço mais baixo repassado ao consumidor final.

O aumento no consumo de refrigerante pelos brasileiros e também pelo mundo é caracterizado pela introdução de novas tecnologias de envase como a garrafa PET e novas tecnologias de produção com máquinas mais rápidas e eficientes. Isso permitiu uma maior diversidade de produtos e queda nos custos de produção. A concorrência e os processos de produção cada vez mais modernos trouxeram também a necessidade da utilização de *softwares* para o controle e planejamento da produção, evitando prejuízos e visando o lucro pelas empresas produtoras da bebida.

Técnicas de otimização e sistemas baseados em modelagem matemática têm sido largamente utilizados em sistemas de controle e planejamento produtivo. Decisões sobre a capacidade limitada dos recursos de produção, tempos de preparação e custos envolvidos, como o de estocagem devem ser levados em conta. Entretanto, o grande número de variáveis e restrições específicas do processo produtivo em fábricas de bebidas dificultam essa tomada de decisões.

Os *softwares* de controle e planejamento da produção vieram para facilitar o processo de tomada de decisão, dentro de um contexto

estratégico/operacional de curto a longo prazo. Antes as dificuldades relacionadas à tomada de decisão eram resolvidas manualmente apenas pelo gerente da produção, que muitas vezes era o único com o conhecimento do processo produtivos da empresa como um todo.

A presente monografia propõe o uso de técnicas de otimização como Algoritmos Genéticos (AGs) dentro do contexto de sistemas de controle e planejamento da produção em fábricas de refrigerantes. AG é um método baseado em uma analogia à Teoria Evolucionista das Espécies proposta por Darwin, onde uma possível solução do problema é obtida através de uma “competição” entre diversas soluções dentro de uma população de soluções. Esse método será detalhado neste trabalho e aplicado ao Problema Integrado de Dimensionamento de Lotes e Programação da Produção (PIDLPP).

O problema em questão trata de uma situação real encontrada em indústrias que fabricam refrigerantes onde o processo de produção é separado em dois níveis principais: nível dos tanques, onde são tomadas decisões sobre a quantidade e o tempo em que as matérias-primas (xaropes) serão armazenadas, e o nível das linhas, onde as mesmas decisões são tomadas em relação à quantidade e ao tempo de engarrafamento das bebidas. Esses dois níveis são interdependentes, pois a produção de um tipo de bebida nas linhas não pode ocorrer sem que a matéria-prima correspondente esteja programada nos tanques.

Outro aspecto importante, que interfere na programação da produção, é o tempo de troca entre xaropes e bebidas. Trata-se do tempo gasto na limpeza e no abastecimento de um tanque, momento em que nenhuma matéria-prima pode ser escoada para as linhas de produção (engarrafamento). Há também um tempo de troca gasto para se alterar o tipo de bebida (produto) engarrafada por uma linha. Esse intervalo de tempo de troca em ambos os casos é dependente da seqüência de troca. O presente trabalho propõe o uso de Algoritmos Genéticos como ferramenta de otimização no processo de tomada de decisão para o

PIDLPP no contexto da fabricação de refrigerantes. Diversas abordagens são propostas para o AG com foco principal na avaliação do uso de diferentes estruturas populacionais e de diferentes tipos de operadores de recombinação.

1.2 Objetivos do Trabalho

O objetivo principal deste trabalho é estudar e avaliar métodos de otimização para a tomada de decisão no processo de fabricação de refrigerantes, ou seja, solucionar o PIDLPP. O foco principal será a avaliação de diferentes estruturas populacionais para o Algoritmo Genético. Além disso, novos operadores genéticos de recombinação, específicos para o problema, serão implementados e avaliados. Outro objetivo é contribuir com o enriquecimento do estado da arte sobre o assunto, enfatizando o uso de Algoritmos Genéticos com populações hierarquicamente estruturadas.

Todos os testes computacionais considerarão instâncias reais fornecidas por fábricas de bebidas. Com isso, será possível fazer comparações entre os diferentes testes e propor o melhor conjunto de variações para o Problema Integrado de Dimensionamento de Lotes e Programação da Produção.

Outros objetivos:

- Estudos relacionados ao entendimento de Problemas de Programação da Produção e Dimensionamento de Lotes.
- Estudos relacionados ao entendimento de métodos heurísticos com ênfase em novas estruturas populacionais.
- Implementação e testes das estruturas populacionais nas instâncias do PIDLPP.
- Implementação e testes dos operadores genéticos de recombinação nas instâncias do PIDLPP.
- Elaboração e análise dos resultados obtidos.

1.3 Estrutura do Trabalho

Capítulo 1: descreve uma breve introdução do presente trabalho apresentando a contextualização em que o problema de pesquisa está inserido no mundo real, fornecendo detalhes sobre o problema estudado.

Capítulo 2: apresenta o levantamento bibliográfico que embasa a pesquisa desenvolvida. Um histórico do setor de bebidas é brevemente descrito, além da apresentação de conceitos básicos e outros tipos de problemas de produção com seus respectivos métodos de resolução.

Capítulo 3: são descritos os métodos de resolução utilizados no presente trabalho com o detalhamento de cada uma das diferentes abordagens aplicadas durante os experimentos.

Capítulo 4: apresenta e compara os resultados obtidos no decorrer da execução dos variados conjuntos de testes.

Capítulo 5: as contribuições e conclusões sobre o trabalho são relatadas, assim como as expectativas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Introdução

Neste capítulo serão apresentados inicialmente dados estatísticos da indústria brasileira de refrigerantes e os conceitos básicos envolvendo problemas de programação da produção, onde algumas de suas classificações serão comentadas. Contudo, o principal destaque será dado aos tipos de problemas e métodos de resolução relacionados ao problema utilizado como estudo de caso, o Problema Integrado de Dimensionamento de Lotes e Programação da Produção.

2.2 Produção de Bebidas no Brasil

O refrigerante é caracterizado como uma bebida não alcoólica e não fermentada, obtida pela dissolução de extratos vegetais em água, açúcar e gás carbônico. A criação do refrigerante não foi de maneira intencional e o seu consumo tinha propósitos diferentes do que se conhece atualmente. Em 1772, um químico inglês chamado Joseph Priestley descobriu como adicionar gás carbônico à água, surgia então a água gaseificada. No entanto, essa mistura começou a ser vendida em 1830 apenas para fins farmacêuticos. Nessa época, ervas e extratos de frutas eram adicionados para dar propriedades medicinais à bebida, como auxílio para digestão, por exemplo. A primeira indústria de bebidas com marca registrada a surgir no mundo nasceu nos Estados Unidos, no ano de 1871, era a Lemon's Superior Sparkling Ginger Aleã (ABIR, 2009).

Entre as maiores fabricantes de refrigerantes do mundo estão Coca-Cola e Pepsi, pioneiras na história do refrigerante, suas fórmulas tiveram a mesma finalidade: ajudar na digestão. Em 1886 na cidade de Atlanta, o farmacêutico John Pemberton criou uma mistura de cor caramelo e a juntou com água

gaseificada. Batizada de Coca Cola, por seu contador, Frank Robinson, eles começaram a vendê-la na farmácia. Um pouco mais tarde, em 28 de agosto de 1898, outro farmacêutico, Caleb Bradham, criou a Pepsi-Cola na Carolina do Norte com a finalidade de rejuvenescer, revigorar e ajudar na digestão. Um reflexo disso foi o próprio nome do produto “Pepsi”, que veio da palavra pepsina, principal enzima que atua no processo da digestão e que estava presente na composição da bebida junto com nozes de cola. A Coca-cola chegou ao Brasil apenas no ano de 1941, mais de meio século depois, e a Pepsi chegou ao Brasil em 1953. Desde então, o consumo de refrigerantes cresceu e se diversificou com novas embalagens, novos sabores e tecnologias utilizadas pelas indústrias de refrigerantes em todo o mundo (About.com, 2009).

Segundo Lafis (2005), no Brasil existem mais de 800 indústrias de bebidas, entre elas estão as que fabricam vinhos, sucos, cervejas, águas envasadas e refrigerantes. Graças ao clima tropical e a grande população, o mercado brasileiro de bebidas vem crescendo nos últimos anos. De acordo com a Associação Brasileira das Indústrias de Refrigerantes e Bebidas não Alcoólicas - ABIR (2009), o consumo de refrigerantes no país foi superior a 14 bilhões de litros em 2008, com um faturamento de mais de 20 bilhões de reais ao ano. O Brasil destaca-se como o terceiro maior mercado do mundo nesse setor, apenas perdendo para os Estados Unidos e México. A Figura 2.1 apresenta o volume do consumo de refrigerantes entre os anos de 1986 e 2008. Há um aumento expressivo do consumo a partir de 1995 devido à implantação do Plano Real que possibilitou o aumento do poder aquisitivo dos brasileiros. Em 2008 houve alta no volume de vendas de 3,57% comparado ao ano de 2007.

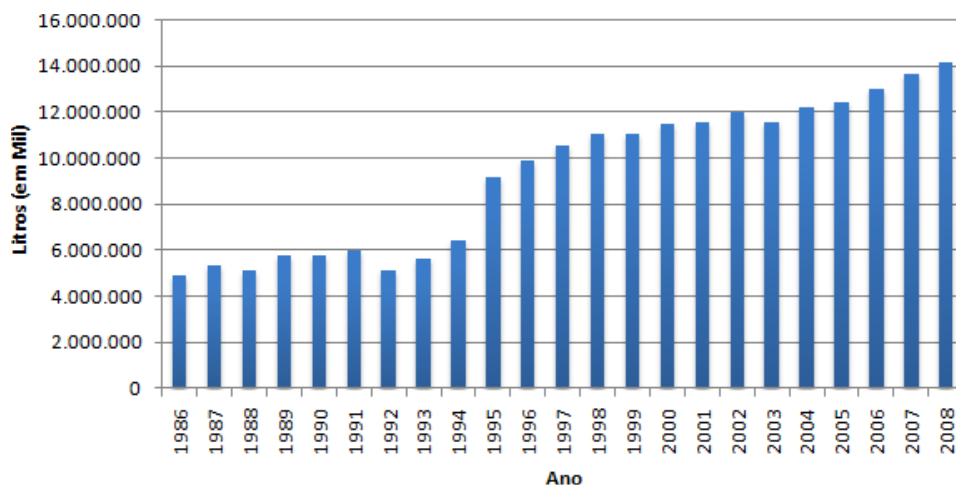


Figura 2.1 - Volume de Venda de Refrigerante no Brasil (Fonte: ABIR, 2009)

No mercado de refrigerantes, podem-se distinguir dois segmentos. O primeiro é constituído pelas grandes empresas que produzem em larga escala com âmbito nacional e internacional. Segundo a ABIR (2009), em 2008 elas foram detentoras de 80,8% do mercado no Brasil com 56,2% a Coca Cola, 17,7% a Companhia de Bebidas das Américas (AmBev) e 6,9% a Pepsi. Essa última passou a fazer parte da AmBev em 1997 e tem se destacado individualmente dentro da companhia por sua alta participação no mercado quando comparada as outras marcas de refrigerantes. O segundo segmento é formado pela Schincariol com participação de 3,4% e por empresas regionais que produzem tubaínas ou refrigerantes populares. Essas empresas juntas constituem 15,8% do mercado nacional, mas se trata de um segmento que vem apresentando uma crescente participação no Brasil. A explicação para tal crescimento se deve a competitividade no preço oferecido aos clientes. A Figura 2.2 ilustra a divisão da participação de vendas no ano de 2008 por esses segmentos.

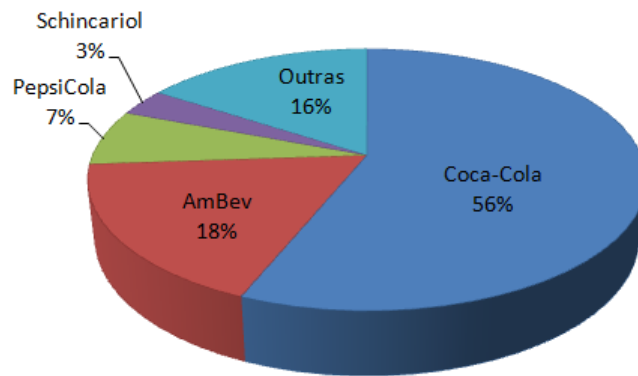


Figura 2.2 - Participação do Volume de Vendas em 2008 (Fonte: ABIR, 2009)

O baixo preço repassado ao consumidor pelas empresas regionais é explicado pela proximidade dessas empresas com os centros consumidores de seus produtos, o que diminui o custo de transporte até os supermercados, padarias, mercearias e barzinhos das periferias das grandes cidades e dos municípios interioranos. Segundo o Banco Nacional de Desenvolvimento Econômico Social – BNDES (2006), um levantamento realizado pelo Instituto Brasileiro de Planejamento Tributário (IBPT), a pedido da Associação dos Fabricantes de Refrigerantes Regionais do Brasil (Afrebras), há mais de 230 fábricas brasileiras de bebidas, com 89 marcas diferentes de refrigerante.

Outro fator que vem influenciando ao longo dos anos o aumento do consumo de refrigerantes no Brasil e no mundo é o uso de embalagens de plástico descartável PET (Polietileno Tereftalato), a implantação de novas tecnologias, novas máquinas e processos produtivos cada vez melhores. Esses fatores aumentaram a concorrência e as exigências do mercado, fazendo com que cada vez mais empresas utilizem softwares para o planejamento e controle da produção.

2.3 Conceitos Básicos

Segundo Karimi *et al.* (2003), o planejamento da produção é uma atividade que consiste no melhor uso dos recursos de forma a satisfazer a produção em um determinado horizonte de tempo. O planejamento da produção envolve decisões de longo, médio e curto prazo. No longo prazo, o foco está em antecipar necessidades agregadas e tomar decisões estratégicas. No médio prazo, o planejamento estabelece quantidades (dimensionamento de lotes), otimiza alguns fatores e satisfaz restrições. No curto prazo, são tomadas decisões do dia-a-dia da programação. Decisões de programação da produção e dimensionamento de lotes nos remetem ao problema de identificar quando e quanto de um produto deve ser produzido. A solução obtida para esse problema pode afetar diretamente o desempenho e a produtividade do sistema produtivo. Por isso, são medidas decisivas para que uma indústria consiga concorrer com outras dentro de um mercado competitivo.

Alguns conceitos (ou características) envolvendo modelos de dimensionamento de lotes irão determinar sua classificação e complexidade. O primeiro é o horizonte de planejamento, um intervalo de tempo que pode ser finito ou infinito, observado continuamente (sistemas contínuos) ou durante períodos discretos de tempo (sistemas discretos). Segundo Belvaux e Wolsey (2001) existem duas terminologias relacionadas a período de tempo: problemas *big bucket* e problemas *small bucket*. Nos problemas chamados *big bucket*, o período de tempo é longo o suficiente para produzir vários itens ao mesmo tempo. Problemas *small bucket* permitem que apenas um item possa ser produzido em cada período. Outro conceito é o número de níveis, um sistema de produção pode ter um só nível ou vários níveis. No primeiro caso, o produto final é simples, sua produção é executada diretamente a partir da matéria-prima e a demanda é estimada a partir do pedido do cliente (demanda independente).

No segundo caso, a matéria-prima precisa ser processada por várias operações antes de ser convertida em produto final, e a demanda de um nível depende da demanda do nível anterior (demanda dependente).

O processo de produção também varia de acordo com o número de produtos, podendo ser classificados pelo planejamento da produção de um único item final, ou então, pelo planejamento da produção de vários itens finais. Os recursos ou capacidades do sistema produtivo, como equipamentos e máquinas, também caracterizam o processo de produção. Um sistema produtivo pode ser considerado mono ou multimáquinas e, quando não existem restrições de capacidade, esse processo de produção é caracterizado como não-capacitado ou capacitado, caso contrário. A demanda por produtos ou itens pode ser estática ou dinâmica, caso seu valor mude durante a produção, ou não. A demanda também pode ser classificada como determinística, quando seu valor é conhecido previamente, ou probabilística, caso o valor seja baseado em alguma medida de probabilidade. Outra característica que impacta na resolução do problema é a inserção de tempos ou custos de troca, que geralmente ocorrem quando há permutação entre diferentes produtos. Se o tempo e o custo de troca forem independentes da seqüência e das decisões tomadas nos períodos anteriores, a estrutura de troca é caracterizada como simples, caso contrário é chamada complexa. Um processo produtivo pode tolerar a ocorrência de algum déficit de demanda em períodos futuros ou até mesmo o seu não atendimento (KARIMI *et al.*, 2003).

Todos esses conceitos e restrições estão relacionados diretamente com a dificuldade de se resolver o problema. Quanto mais complexo, mais restrições estarão envolvidas e decisões cada vez mais elaboradas devem ser tomadas, durante o dimensionamento dos lotes e a programação da produção.

2.4 Problemas de Programação da Produção

2.4.1 Tipos de Problema

Programar a produção consiste em planejar e executar um plano que deverá ser seguido em um pequeno período de tempo, que pode ser medido em horas, dias ou turnos. Os problemas envolvendo programação da produção são separados levando-se em conta qual parte da produção é tratada por eles. Na literatura há trabalhos que tratam apenas do dimensionamento de lotes, como Kuik *et al.* (1994) e Brahimi *et al.* (2006), outros da programação da produção, como Pinedo (1995) e Cheng *et al.* (2004) e trabalhos que envolvem tanto o dimensionamento como a programação da produção, entre eles Drexl e Kimms (1997) e Belvaux e Wolsey (2001). A Figura 2.3 ilustra esta divisão, levando-se em conta que esses problemas podem ser capacitados e não capacitados.

Existem ainda outras maneiras para se classificar os problemas de programação da produção, como pelo número de máquinas, número de itens, tipo de demanda, etc., de acordo com os conceitos apresentados na Seção 2.3.

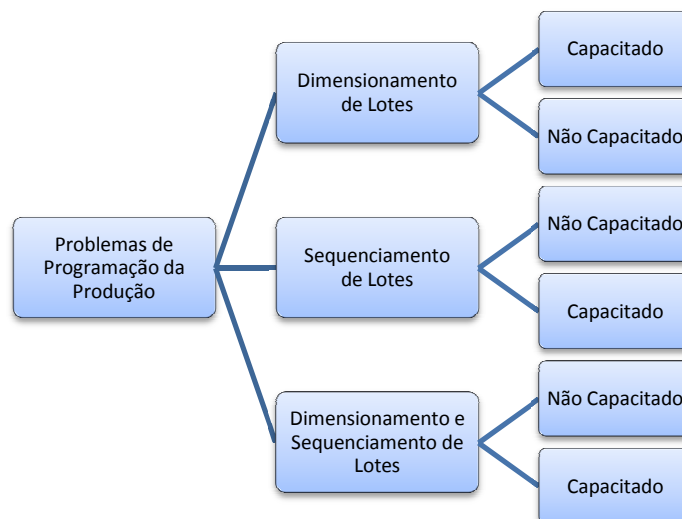


Figura 2.3 - Divisão dos Problemas de Programação da Produção

Um dos primeiros estudos de dimensionamento de lotes foi realizado por Wagner e Whitin em 1958, o *Economic Order Quantity* (EOQ) com escala de tempo contínua, demanda constante e horizonte de tempo infinito.

Existem algumas variações do problema de dimensionamento de lotes, são elas: o *Economic Lot Scheduling Problem* (ELSP), o *Capacitated Lot Sizing Problem* (CLSP), *Discrete Lot Sizing and Scheduling Problem* (DLSP), o *Continuous Setup Lot Sizing Problem* (CSLP), o *Proportional Lot Sizing and Scheduling Problem* (PLSP) e o *General Lot Sizing and Scheduling Problem* (GLSP).

O ELSP é um problema com um único nível, multi-itens e demanda estática, sua escala de tempo é contínua e o horizonte de tempo infinito. A resolução desse problema considerando restrições de capacidades é NP-difícil (GALLEGO e SHAW, 1997).

O CLSP também é um problema NP-difícil que apresenta um único nível de produção, horizonte de planejamento finito, demandas dinâmicas e restrições de capacidade sobre a quantidade de produção em cada período. Seus custos de produção podem variar para cada produto e para cada período, não dependendo da seqüência em que são programados, porém existem variações do CLSP quanto a essa independência na seqüência de produção (KARIMI, 2003).

Outro problema NP-difícil é o DLSP, onde os macro-períodos do CLSP são subdivididos em vários micro-períodos. No DLSP apenas um item pode ser produzido por micro-período, funcionando como se o lote de produção estivesse usando toda a capacidade. Por isso, trata-se de um problema do tipo *small bucket* (Seção 2.3) (BRUGGEMANN e JAHNKE, 2000). O CSLP, quando comparado ao DLSP, é mais geral e mais próximo da realidade, pois não existe a suposição de que um item utilize toda a capacidade disponível. O produto pode utilizar parte da capacidade disponível em um micro-período, todavia, continua-se com apenas um produto produzido por micro-período. (DREXL e KIMMS, 1997).

O PLSP amplia a idéia do DLSP ao usar a capacidade disponível em um micro-período para a produção de um segundo item, caso essa capacidade não seja totalmente ocupada. Sendo assim, o PLSP assume que o estado de uma máquina pode ser mudado mais de uma vez por micro-período (DREXL e HAASE, 1995).

2.4.2 Métodos de Resolução

A maior parte dos problemas de dimensionamento de lotes são difíceis de serem resolvidos. Florian *et al.* (1980) provaram que o problema capacitado de apenas um item é resolvido em tempo polinomial. Por causa da dificuldade desses problemas, diferentes métodos de resolução foram criados e utilizados para resolvê-los. Métodos exatos como *Branch and Bound* (BB), ou métodos não-exatos como heurísticas e meta-heurísticas, ou ainda métodos híbridos que combinam essas abordagens são utilizados para resolver problemas de programação da produção.

Métodos exatos são aqueles cuja solução encontrada em qualquer uma das execuções do método, sob as mesmas condições, será sempre a mesma. Proposto por Land e Doig (1960), o método exato, *Branch and Bound* consiste de uma enumeração sistemática de todas as soluções candidatas, onde um amplo conjunto de soluções é descartado em massa pelo uso de limitantes superiores e inferiores. Fleischmann (1990), por exemplo, utiliza esse método junto com a Relaxação Lagrangiana para resolver o DLSP. Barany *et al.* (1984) e Leung *et al.* (1989) utilizam o BB junto com planos de corte para resolver o CLSP. Esses métodos não são tão utilizados em problemas integrados de dimensionamento de lotes e programação da produção devido à complexidade do problema. Métodos exatos podem levar muito tempo para resolverem instâncias complexas do problema, justificando desta forma o uso de heurísticas, segundo Karimi *et al.* (2003).

Métodos heurísticos são técnicas que resolvem problemas por aprendizado e descoberta, são usados para encontrar soluções rápidas com esperança de que elas sejam próximas de uma resposta possível ou da solução ótima. Heurísticas podem ser baseadas em observações da natureza, como analogias, em regras básicas ou então pelo senso comum (MICHALEWICZ e FOGEL, 2004). Segundo Jans e Degraeve (2007), a principal razão para o uso desses métodos é a sua flexibilidade e habilidade para lidar com problemas complexos.

Problemas como o CLSP, DLSP, PLSP e o GLSP são resolvidos por diversos autores utilizando diferentes heurísticas e meta-heurísticas. Gupta e Magnusson (2005) propõem uma heurística para o CLSP com custos e tempos de preparo dependentes da seqüência. A heurística estabelece inicialmente uma solução factível para em seguida executar passos de melhoria desta solução. Özdamar e Birbil (1998) utilizam um Algoritmo Genético (AG) associado a uma busca local híbrida para o CLSP considerando máquinas paralelas. Haase e Kimms (2000) também utilizam um AG para resolver o PLSP. Meyr (2002) estende o GLSP com tempo de preparação para o contexto de máquinas paralelas, utilizando heurísticas baseadas em buscas locais como Simulated Annealing e Threshold Accepting.

2.5 Estudo de Caso

O estudo de caso do presente trabalho é o Problema Integrado de Dimensionamento de Lotes e Programação da Produção (PIDLPP), os parágrafos seguintes descrevem em detalhes as características do problema e as restrições envolvidas.

O processo de fabricação de bebidas envolve o uso de diferentes matérias primas que misturadas a outros componentes resultam nos xaropes. Os

xaropes ficam armazenados em tanques de onde escoam para as linhas de produção. A combinação do tipo de xarope que escoam dos tanques com o tipo de envase realizado nas linhas resulta na bebida ou produto final. A Figura 2.4 representa o processo de produção considerado.

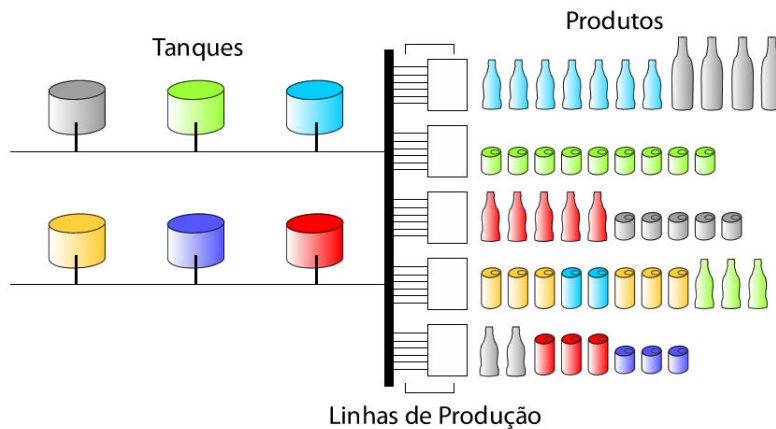


Figura 2.4 - Processo de Produção de Bebidas

As demandas por bebidas são planejadas geralmente em um horizonte de tempo mensal e devem ser atendidas ao final de cada semana de trabalho. A planta industrial apresenta dois níveis de produção: tanques e linhas de engarrafamento. Nos tanques, o responsável pela produção precisa realizar o dimensionamento e a programação (seqüenciamento) dos xaropes, onde se deve decidir quanto e quando determinado xarope será armazenado em cada um dos tanques disponíveis. Nas linhas de engarrafamento, o responsável pela produção estabelece o dimensionamento e a programação das bebidas ou produtos finais, decidindo quanto e quando determinado produto será colocado em produção nas linhas existentes.

Diversas restrições e decisões estão presentes em cada nível de produção considerado. Todos os tanques envolvidos têm uma capacidade máxima de

armazenagem que não pode ser ultrapassada. Os tanques também devem ser abastecidos com uma quantidade mínima de xarope e podem armazenar um único xarope por vez, mas o mesmo xarope pode estar armazenado em diversos tanques. Um tanque é reabastecido apenas quando o xarope nele armazenado for completamente escoado para as linhas de produção. Há um custo relacionado à troca de um xarope por outro nos tanques. Também há um gasto de tempo durante esta troca. Isto ocorre porque o armazenamento de um novo xarope no tanque exige a limpeza do tanque. Tanto o tempo quanto o custo de troca dependem do produto que estava previamente no tanque. Desta forma, deve-se decidir a programação dos xaropes nos tanques que atenda às demandas, minimizando os custos e o tempo gasto nas trocas realizadas.

As linhas podem realizar mais de um tipo de envase, por exemplo, uma mesma linha pode ser ajustada para engarrafar o xarope em uma embalagem de 600ml, 1l ou 2l, dependendo da programação estabelecida. O tempo e o custo envolvidos no preparo da linha dependem do tipo de bebida previamente produzida, por exemplo, a seqüência “refrigerante normal - refrigerante *diet*” leva a custos e tempos de preparo maiores do que a seqüência inversa. O tempo de processamento depende de cada linha, onde uma mesma linha em geral tem diferentes tempos de processamento para diferentes tipos de envase. Trocas ou reposições de xarope em um tanque exigem a interrupção da produção nas linhas que o utilizam. A demanda semanal deve ser atendida e o excedente produzido deve ser estocado.

As decisões relativas às linhas de produção envolvem a programação dos produtos nas linhas que atenda às demandas, minimizando os custos e os tempos gastos nas trocas realizadas. As decisões e as restrições relativas aos tanques e às linhas de engarrafamento são interdependentes, pois as quantidades de xaropes armazenadas nos tanques devem ser suficientes para atender a produção nas linhas. Além disso, a programação dos xaropes nos tanques deve

estar sincronizada à programação das linhas. Não poderá ocorrer o envase de determinada bebida, se o xarope correspondente não estiver armazenado em um tanque.

A Figura 2.5 ilustra um exemplo da falta de sincronização durante o seqüenciamento da produção. Suponha quatro produtos (P1, P2, P3 e P4) que podem utilizar três tipos de xaropes distintos (X1, X2 e X3). Os retângulos na parte inferior representam o tempo de produção dos lotes de produtos na linha. Os retângulos na parte superior da figura apresentam o tempo de permanência dos lotes de xaropes no tanque. Os espaços entre esses retângulos são os tempos de troca, ou seja, o tempo de espera para que a linha seja preparada antes de começar a fabricar um produto diferente, ou então, o tempo de espera para que o tanque seja limpo e esteja abastecido com o xarope necessário.

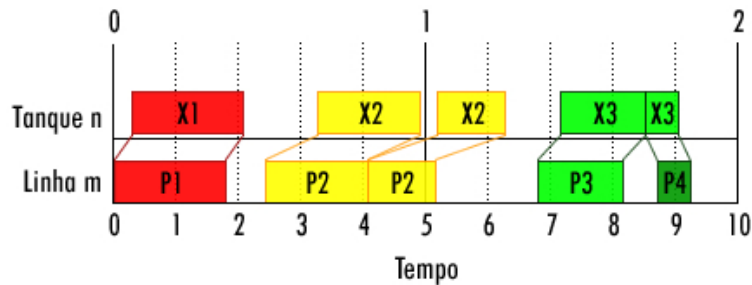


Figura 2.5 - Programação Não Sincronizada da Produção

Neste exemplo, o produto 1 (P1) começa a ser produzido no instante inicial do horizonte de tempo, porém, observe que o xarope 1 (X1) ainda não está completamente preparado, o que na prática é inconsistente. Não é possível começar a produção sem que a matéria-prima, nesse caso o xarope, esteja disponível. Observe que o tempo de troca de X1 para X2 é diferente do tempo de troca de P1 para P2, pois é gasto mais tempo para preparar o tanque do que para preparar a linha. Ainda que esses tempos fossem iguais, a produção de P2 não

seria viável. Isso ocorre porque a produção de P1 foi iniciada antes do preparo do tanque com X1, adiantando também o início da produção de P2. Além disso, X2 ainda não estaria completamente pronto para ser utilizado, ocorrendo novamente a falta de sincronização.

Na produção do segundo lote de P2 foi necessário o reabastecimento do tanque com X2, isso pode ocorrer já que os tanques têm capacidade limitada. Todavia, o reabastecimento do tanque não foi levado em conta na linha, ou seja, não foi interrompida a produção de P2.

Na última troca de xaropes (X2 para X3) o tempo de espera para o preparo do tanque foi menor que o da linha entre P2 e P3. Na produção de P4, o tanque possuía anteriormente uma quantidade suficiente de xarope para produzir os dois lotes, mas mesmo assim, existe o tempo de troca na linha de P3 para P4. Nesse caso, P4 começa sua produção atrasada.

A programação mostrada na Figura 2.6 ilustra o que ocorreria se fossem consideradas todas as esperas mencionadas anteriormente e representadas aqui pelos retângulos riscados.

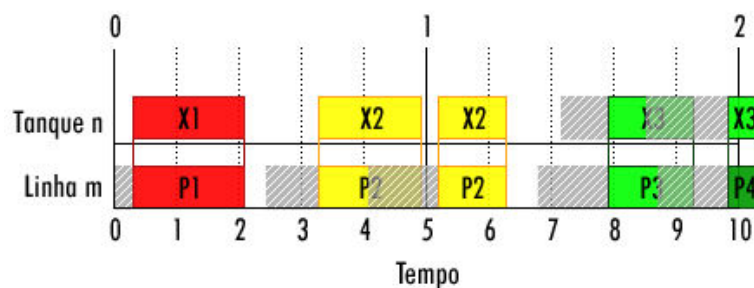


Figura 2.6 - Programação Sincronizada da Produção

A sincronia entre os dois estágios permite a integração da armazenagem e da produção dos lotes em ambos os cenários (tanque e linha). Porém, a produção ainda se mostra inviável, pois ultrapassa a capacidade disponível no

horizonte de tempo apresentado. Dessa forma, a sincronia deve ser feita no momento em que o dimensionamento dos lotes e a programação da produção estão sendo realizados para ambos os cenários.

O principal objetivo em problemas industriais é a minimização dos custos de troca, de estoque e de produção, respeitando restrições previamente determinadas. Toledo *et al.* (2007) apresenta uma formulação para a minimização desses custos, descrita pela seguinte função objetivo em (1):

$$\begin{aligned}
& \sum_{i=1}^J \sum_{j=1}^J \sum_{l=1}^L \sum_{s=1}^{T \cdot S} s_{ijl} z_{ijls} + \sum_{j=1}^J \sum_{t=1}^T h_j I_{jt} + \sum_{j=1}^J \sum_{l=1}^L \sum_{s=1}^{T \cdot S} v_{jl} q_{jls} + \\
& + \sum_{i=1}^{\bar{J}} \sum_{j=1}^{\bar{J}} \sum_{k=1}^{\bar{L}} \sum_{s=1}^{\bar{T} \cdot \bar{S}} s_{ijk} z_{ijks} + \sum_{j=1}^{\bar{J}} \sum_{k=1}^{\bar{L}} \sum_{t=1}^{\bar{T}} h_j \bar{I}_{jk,t \cdot T^m} + \sum_{j=1}^{\bar{J}} \sum_{k=1}^{\bar{L}} \sum_{s=1}^{\bar{T} \cdot \bar{S}} v_{jk} q_{jks} + \quad (1) \\
& + M \sum_{j=1}^J q_j^0
\end{aligned}$$

Na primeira linha da expressão estão os custos envolvidos nas linhas de produção, na segunda os custos nos tanques e na última há uma penalização, caso alguma demanda não seja atendida. A primeira parcela da expressão, tanto para segunda quanto para a primeira linha da equação (1), representa a soma de todos os custos quando ocorre a troca de um produto ou xarope i para j em uma linha l ou em um tanque k , respectivamente. Nos dois casos, o custo de troca é proporcional ao seu tempo, ou seja, quanto mais tempo se leva para fazer a troca de um produto para outro ou de um xarope para outro, maior é o custo envolvido no processo. Na segunda parcela da soma na primeira linha, estão dimensionados os custos da quantidade de um produto j em estoque ao final de um macro-período t . Há uma diferença em relação aos tanques nessa segunda parcela quando comparada à segunda linha da função objetivo. No custo de

estoque dos tanques, deve ser levado em conta a quantidade de xarope j em estoque para cada tanque k no final do último micro-período T^m do macro-período considerado. Os macro-períodos são períodos maiores de tempo, em um horizonte de tempo mensal, por exemplo, eles corresponderiam às semanas enquanto os micro-períodos representariam os dias dentro de cada semana. Na última parcela das duas primeiras linhas da equação (1) estão os custos de produção nas linhas e nos tanques. Este custo irá depender da quantidade de produto ou xarope j em cada lote s , respectivamente. Uma descrição detalhada da notação utilizada, das restrições e da função objetivo propostas por Toledo em sua modelagem matemática podem ser encontradas em Toledo (2005).

O problema apresentado caracteriza-se como multinível ao apresentar a programação de dois níveis interdependentes que devem ser totalmente sincronizados. Há também restrições de capacidades e as demandas devem ser respeitadas. Além disso, são necessárias decisões envolvendo quantidades e seqüências. É um problema que utiliza máquinas paralelas já que tanques e linhas atuam como processadores em paralelo tanto para armazenagem de xaropes (tanques) quanto para produção de bebidas (linhas). A programação de xaropes nos tanques e produtos nas linhas é o que determina os custos envolvidos em toda a produção.

2.6 Trabalhos Relacionados ao PIDLPP

O PIDLPP descreve uma situação bastante geral envolvendo o processo de produção em uma fábrica de refrigerantes. Toledo *et al.* (2007) propuseram um modelo matemático inteiro-misto que introduziu diversas restrições combinadas que até então costumavam ser tratadas separadamente na literatura. A não existência de testes com modelos similares levou os autores a criarem um conjunto de instâncias para avaliar o modelo. As instâncias foram solucionadas

por meio do pacote computacional GAMS/Cplex, um sistema de modelagem em alto nível para programação matemática e otimização. Todavia, a solução exata se mostrou viável apenas em instâncias de pequena dimensão devido à complexidade do PIDLPP considerado.

Segundo Toledo *et al.* (2009), isso se explica pelo fato do PIDLPP englobar a resolução integrada de dois Problemas de Programação e Dimensionamento de Lotes Capacitado (PPDLC) como subproblemas, um nos tanques e outro nas linhas de produção. Bitran e Yanasse (1982) e Meyr (2002) provaram que o PPDLC é um problema NP-Difícil. O PIDLPP também apresenta custos de troca dependentes da seqüência em cada nível de produção, logo há custos de troca dependentes da seqüência em cada PPDLC presente no PIDLPP. Maes *et al.* (1991) demonstraram que determinar uma solução factível, ou seja, que consiga atender todas as restrições do problema (como por exemplo, atenda as demandas, não estoure as capacidades e etc.), para o PPDLC com tempos de troca também é um problema NP-Difícil.

Ferreira *et al.* (2009) também propuseram modelos matemáticos para descrever o PIDLPP. Todavia, os modelos propostos são mais simples que em Toledo *et al.* (2007) e foram estabelecidos a partir de algumas mudanças nas restrições do PIDLPP. Primeiro, foi considerado que o número de linhas é igual ao número de tanques e que cada um deles é dedicado a uma única linha. A Figura 2.7 ilustra esta situação.

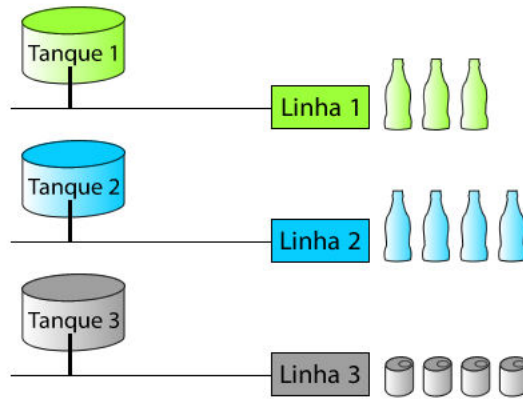


Figura 2.7 - Novas Restrições ao Processo de Produção

Isto se justificou pelo fato de que considerar o número de linhas diferente do número de tanques pode inviabilizar a resolução do modelo por ferramentas de apoio a decisão, já que a complexidade do problema é aumentada. Além disso, em um cenário real de uma fábrica de bebidas, um tanque normalmente fica dedicado a uma determinada linha. O modelo proposto por Toledo *et al.* (2007) é mais geral e pode resolver problemas em que o número de tanques é diferente do número de linhas, ao contrário do modelo proposto por Ferreira *et al.* (2009) que não se aplica a esse caso.

Ferreira *et al.* (2009) também propôs um modelo para o cenário em que há uma única linha de engarrafamento e outro modelo para um cenário em que a programação dos tanques é desconsiderada. Nesse caso, os autores supõem que o gargalo de produção ocorra apenas nas linhas de engarrafamento. Os autores também estabeleceram instâncias para esses modelos a partir de dados coletados junto a três fábricas de bebidas.

2.7 Algoritmos Genéticos

Segundo Linden (2006), Algoritmos Genéticos (AGs) são técnicas heurísticas inspiradas em processos biológicos de evolução natural e que fazem parte da classe dos algoritmos evolucionários. Os AGs utilizam o mesmo princípio da seleção natural e sobrevivência do mais apto apresentado por Charles Darwin, naturalista e fisiologista inglês, em seu livro “A Origem das Espécies” (Darwin, 1872).

O princípio da seleção natural, de acordo com Atamar (1992), indica que os indivíduos cujas variações se adaptam melhor ao ambiente terão maior probabilidade de sobreviver e também de se reproduzir. Este processo faz com que os descendentes sejam variações de seus pais. Essas variações podem ser positivas ou negativas mostrando que não se trata de um processo com a finalidade de maximizar algumas características da espécie. Na verdade, não existe nenhuma garantia de que se os pais são bem adaptados seus descendentes também serão (LINDEN, 2006).

Segundo Von Zuben (2009), a seleção natural aponta os tipos de variações que devem conduzir os organismos a uma melhor adaptação ao meio e não apenas prevê essas variações, esse processo ocorre de maneira probabilística.

2.7.1 Algoritmo Genético Canônico

O primeiro trabalho conhecido envolvendo AGs foi apresentado por John Holland no ano de 1975 (Holland, 1975) e, segundo Vose (1993), até hoje a maior parte dos algoritmos genéticos criados são variações do algoritmo descrito por Holland. Esse algoritmo ficou conhecido como algoritmo genético canônico, porém o trabalho de Holland só veio a ser popularizado mais tarde por um de seus alunos, David Goldberg (1989).

De acordo com Gen e Cheng (1997), o grande diferencial dos AGs, se comparados a outras técnicas de pesquisa, é a utilização de um conjunto inicial de soluções gerado de maneira aleatória, chamado de população. É na população onde serão feitas todas as simulações dos processos naturais da evolução de uma espécie, aplicando-se métodos de seleção, reprodução e mutação em seus indivíduos. Cada indivíduo gerado recebe uma avaliação de sua aptidão (*fitness*) que nada mais é do que uma quantificação da sua qualidade para a resolução do problema em questão. Os operadores genéticos utilizados, como recombinação (*crossover*) e mutação atuarão de forma a permitir a sobrevivência dos mais aptos, ou seja, aqueles que melhor resolvem o problema.

Cada indivíduo da população é representado por um cromossomo e sua aptidão. Cromossomos são geralmente codificados na forma de listas de atributos, onde cada atributo é conhecido como gene. Os possíveis valores que um determinado gene pode assumir são denominados alelos. Cada indivíduo de uma população representa um candidato em potencial à solução do problema em questão. A representação do cromossomo consiste em traduzir a informação do problema estudado de uma maneira viável a ser tratada pelo computador. Ela deve ser o mais simples possível, onde restrições impostas pelo problema devem estar implícitas dentro da representação adotada. Segundo Von Zuben (2009), a forma como um indivíduo é representado, ou seja, sua codificação é uma das etapas mais críticas na construção de um algoritmo genético.

O *fitness* é uma função de avaliação, também conhecida como função objetivo ou de custo, e determina a qualidade do indivíduo (cromossomo) enquanto solução para o problema. Essa função é uma parte não genérica dos AGs que deve ser capaz de identificar todas as restrições e objetivos, ou seja, ela é específica para cada problema. A função objetivo fornece uma medida de desempenho com respeito a um conjunto particular de parâmetros, através dela é possível obter a aptidão de um indivíduo. Essa aptidão é definida em relação a

outros membros da atual população. A função aptidão pode ser igual à função objetivo, ou resultado do escalonamento da função objetivo, ou então baseada no *ranking* do indivíduo em sua população.

Operadores genéticos são mecanismos que buscam perturbar o sistema, ao exemplo do que ocorre na natureza, e possuem a função de buscar nos AGs regiões desconhecidas do espaço de soluções. Esses operadores são a recombinação (*crossover*) e a mutação. Tais operadores ocorrem no algoritmo genético segundo taxas de probabilidade percentual de ocorrência. A recombinação consegue através da troca de informações codificadas no indivíduo, encontrar novas e, se possível, melhores soluções. De acordo com Bessaou e Siarry (2001), a recombinação permite criar novos indivíduos (filhos) a partir daqueles já existentes (pais), onde filhos podem herdar de seus pais os melhores atributos. Após a recombinação, é aplicado um operador genético de mutação que altera algumas informações codificadas no indivíduo, o que permite incorporar maior diversidade à população. Isso faz com que a população não fique presa em determinados pontos dentro de um espaço de busca por soluções.

Seguida da aplicação da recombinação e da mutação ocorre a seleção. Segundo Linden (2006), o método de seleção de pais deve simular o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os pais mais capazes, ou seja, mais bem adaptados, geram mais filhos. Ao mesmo tempo, os pais menos aptos também devem poder gerar descendentes para que a população não tenha somente indivíduos semelhantes, sem uma determinada característica que no futuro possa vir a ser importante.

A descrição padrão dos AGs pode ser vista no pseudocódigo da Figura 2.8, onde t é o contador de gerações, P a população geral, C os reprodutores selecionados e C' os novos indivíduos resultantes da recombinação e mutação.

```

1  t = 0
2  inicializar P(t)
3  avaliar P(t)
4  enquanto não atingir o critério de parada faça
5      t = t + 1
6      selecionar pais C(t) a partir de P(t - 1)
7      C'(t) = recombinação e mutação em C(t)
8      avaliar C'(t)
9      selecionar e substituir em P(t) a partir de C'(t) e P(t -
    1)
10 fim do enquanto

```

Figura 2.8 - Algoritmo Genético Canônico

O passo mais importante do algoritmo é a busca na população pelas melhores soluções (linha 6) para que essas sejam recombinadas e gerem outras soluções ainda melhores (linha 7). O algoritmo para quando o critério de parada for encontrado (linha 4). Ele pode ser estabelecido por tempo, número de gerações, convergência da população ou outro tipo de informação desejada. Como é possível notar, os algoritmos genéticos não são processos dirigidos à obtenção da solução ótima, na verdade é a competição entre os indivíduos que determina as melhores soluções, ou seja, eles dependem de fatores probabilísticos. Isso faz com que um AG dificilmente repita os mesmos resultados de uma execução anterior. Logo, são considerados métodos heurísticos por não assegurarem a obtenção do melhor resultado em todas as suas execuções.

2.7.2 Algoritmo Genético Multipopulacional

De acordo com Linden (2006), os biólogos foram capazes de perceber que em ambientes isolados como, por exemplo, ilhas, frequentemente são geradas espécies de animais melhor adaptadas às características de seus

ambientes do que em áreas de maior extensão. Analogamente, esta observação forneceu subsídios para a utilização de um novo modelo de AG com múltiplas populações.

Neste modelo, conhecido por multipopulacional, uma população completa é dividida em subpopulações que podem, ou não, ter a mesma quantidade de indivíduos. Em um algoritmo genético padrão, decidir a quantidade de indivíduos pode ser problemático, enquanto que em um AG multipopulacional essa decisão é composta pela necessidade de selecionar a quantidade de subpopulações. Segundo Bäck *et al.* (2000), o número de indivíduos em cada subpopulação é crucial para a dinâmica da evolução e é totalmente dependente da natureza do problema. A próxima decisão está relacionada à ocorrência de uma intercomunicação entre as subpopulações que pode ser importante para a evolução do AG multipopulacional. Assim, é possível realizar migrações, com uma certa frequência, determinando um maior isolamento ou interação entre as subpopulações. Um processo de migração poderia ser feito eliminando, ou não, o indivíduo da sua população original antes de movê-lo para outra.

Ao final de cada evolução isolada de uma subpopulação, chamada de época, ocorre a migração. O tamanho de cada época pode ser determinado pelo número de gerações, porém a formulação mais fiel a teoria da evolução natural é permitir que cada subpopulação alcance um equilíbrio ou convergência. Nesse caso, a carga de trabalho pode se tornar desbalanceada, pois as populações que convergirem mais rapidamente serão forçados a ficarem ociosas.

Tratando-se de Algoritmos Genéticos, segundo o conceito de *genetic drift* (Weiner, 1995), a abordagem multipopulacional no AG permite maior exploração do espaço de busca, pois indivíduos em populações distintas seguem caminhos evolutivos diferentes. Enquanto que a abordagem uni populacional intensifica a busca num espaço mais restrito e isso diminui a possibilidade de se

encontrar soluções de melhor qualidade ou até mesmo soluções factíveis. A abordagem multipopulacional prestigia o aspecto exploratório na busca de soluções (TOLEDO, 2007). As populações evoluem em paralelo, possibilitando compartilhar os bons resultados entre as múltiplas populações, através de migrações de indivíduos, criando uma sinergia e tornando assim as abordagens multipopulacionais mais eficientes.

O presente trabalho utiliza um algoritmo genético multipopulacional que será explicado em detalhes no Capítulo 3.

2.7.3 Algoritmos Genéticos Híbridos

Os algoritmos genéticos tem se mostrado uma versátil e efetiva ferramenta para ser utilizada em situações de otimização, mas existem muitas situações em que os algoritmos genéticos não são particularmente bons, e vários métodos de hibridização vem sendo propostos. Goldberg (1989) incluiu melhorias nos operadores locais dos algoritmos genéticos puros para otimização de problemas NP-difícil.

Gen e Cheng (1997) mencionam uma das formas mais comuns de algoritmos genéticos híbridos, a de incorporar uma otimização local como um complemento extra no simples laço de recombinação e seleção. A otimização local é aplicada a cada novo filho gerado na população. Dessa forma, a abordagem híbrida muitas vezes melhora o desempenho destes métodos.

3 MÉTODO DE RESOLUÇÃO

3.1 Introdução

Neste capítulo o Algoritmo Genético multipopulacional é descrito com detalhes a respeito da representação dos indivíduos, função objetivo, operadores genéticos utilizados e estruturas populacionais consideradas na resolução do problema.

3.2 Algoritmo Genético Multipopulacional

O Algoritmo Genético multipopulacional utilizado segue a estrutura já implementada no ambiente Np-Opt que permite um alto grau de reutilização de código para problemas de otimização. Este ambiente foi proposto por Mendes (2003) e está fundamentado na orientação a objetos, utilizando a linguagem de programação Java.

De forma geral, se um pesquisador deseja reutilizar em um problema diferente, um mesmo método de otimização já implementado, ele terá de refazer grande parte do código. Isso acontece especialmente se o código não estiver bem estruturado ou não utilizar orientação a objetos. No ambiente Np-Opt, a reutilização de código permite, por exemplo, a adição de novos problemas sem alterações críticas no código. O ambiente também possui uma interface gráfica de utilização simplificada para o usuário, onde ele pode fazer mudanças nos valores de diferentes parâmetros (como taxas de mutação, recombinação, tipo de estrutura e etc.), escolher instâncias, criar lotes de testes e uma série de outras opções disponíveis, além disso, também é possível a adição de novos métodos.

O pseudocódigo do Algoritmo Genético multipopulacional pode ser visto na Figura 3.1.

```

Método AGMultiPopulacional
início
  para i = 1 até númeroDePopulações faça
    inicializaPopulação(pop(i))
    avaliaFitnessPopulação(pop(i))
    estruturaPopulação(pop(i))
  fim para
  repetir
    para i = 1 até númeroDePopulações faça
      repetir
        para j = 1 até númeroDeRecombinações faça
          selecionaPais(individuoA, individuoB)  $\subseteq$  pop(i)
          novoInd = recombina(individuoA, individuoB)
          se (efetuaMutaç o) ent o
            novoInd = mutaç o(novoInd)
          fim se
          avaliaFitnessIndiv duo(novoInd)
          inserePopulaç o(novoInd, pop(i))
        fim para
        estruturaPopulaç o(pop(i))
      at (populaç oConvergiu pop(i))
    fim para
    reinicializaPopulaç es
    efetuaMigraç o
  at (tempo limite)
fim

```

Figura 3.1 - Algoritmo Gen tico Multipopulacional

Na primeira fase do algoritmo os indiv duos de cada populaç o s o inicializados (inicializaPopulaç o), esse procedimento   feito de maneira aleat ria utilizando v rios dados do problema. Ent o toda a populaç o   avaliada (avaliaFitnessPopulaç o) e organizada (estruturaPopulaç o) de acordo com a estrutura determinada pelo usu rio que pode ser ou n o hier rquica. A avaliaç o

do *fitness* de cada indivíduo é feita para que esses possam ser diferenciados e classificados como mais aptos ou não, em termos evolutivos. A seleção (selecionaPais), que é o próximo passo do algoritmo, tende a preservar os mais aptos para obter combinações de indivíduos melhores. Todavia, não há garantias de que isso venha a ocorrer, pois nada impede que existam bons cruzamentos entre indivíduos menos aptos da população.

Com a escolha dos dois pais (individuoA, individuoB) na seleção, a recombinação é realizada e gera um filho (novoInd). A mutação pode ocorrer ou não dependendo de uma taxa pré-definida. Finalmente o novo indivíduo é avaliado (avaliaFitnessIndivíduo) e inserido (inserePopulação) na população, se for melhor que um de seus pais. As operações de seleção, recombinação e mutação podem ocorrer várias vezes e terminam com a re-organização de toda a população, visando manter a estrutura populacional após a inserção dos novos indivíduos. O número de vezes em que esses operadores genéticos são executados (numeroDeRecombinações) é determinado pelo produto entre taxa de recombinação e o tamanho da população. Por exemplo, para uma taxa de 1,5 e uma população de tamanho 100, serão realizadas 150 recombinações. A mutação é aplicada depois do cruzamento entre os pais selecionados, porém ela ocorre de maneira probabilística sob uma taxa pré-definida. Os processos descritos anteriormente ocorrem até que a população em questão tenha convergido. Isto ocorre quando nenhum novo indivíduo é inserido e então o processo evolutivo continua na próxima população.

A segunda fase do algoritmo é a reinicialização de todas as populações (reinicializaPopulações) preservando apenas o melhor indivíduo. Esse indivíduo também é migrado (efetuaMigração), ou seja, aquele que possuir o melhor *fitness* é inserido na próxima população. Se essa população for a última, ele é inserido na primeira população. A migração entre três populações organizadas

em árvores ternárias (as estruturas populacionais serão descritas na Seção 3.5) pode ser vista na Figura 3.2.

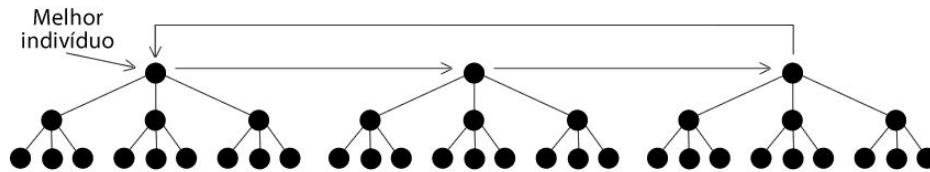


Figura 3.2 - Migração

O critério de parada do algoritmo é definido por um tempo limite de execução determinado pelo usuário.

3.3 Representação dos Indivíduos

Um dos fatores mais importantes que irão influenciar no desempenho de um algoritmo genético é como os indivíduos, também chamados de cromossomos, são representados. Afinal, são eles que irão conter a solução para o problema em questão. Não existe uma regra a ser seguida que diga a melhor maneira de representá-los.

3.3.1 Codificação e Inicialização

Toledo (2005) utilizou uma representação complexa capaz de determinar a programação dos lotes de produtos nas linhas e dos xaropes nos tanques. A Figura 3.3 ilustra a esquematização da codificação dos cromossomos, os quais são formados por pequenos genes que contém as informações dos lotes que serão produzidos.

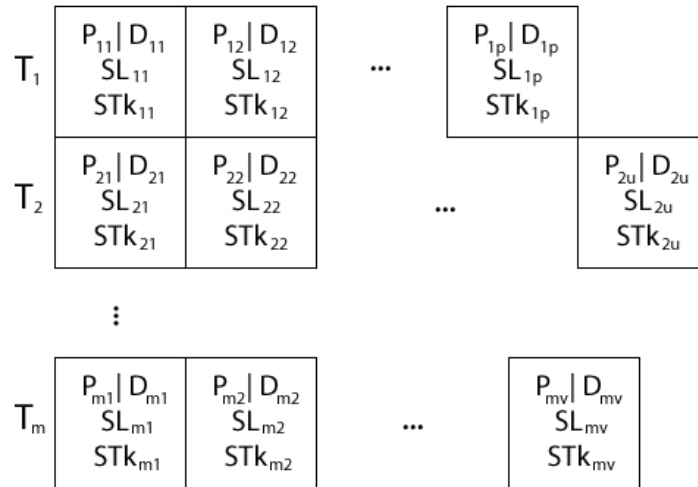


Figura 3.3 - Indivíduo (Toledo, 2005)

Cada um dos indivíduos é uma lista com um número de macro-períodos fixo, T_1, T_2, \dots, T_m , com quantidades de genes variadas. No exemplo da Figura 3.3 no primeiro macro-período existem p genes, u genes no segundo e v genes no último, cada um dos genes nos macro-períodos possuem quatro atributos:

- P_{mn} : n -ésimo produto a ser produzido no macro-período m
- D_{mn} : tamanho do lote do produto P_{mn}
- SL_{mn} : seqüência de possíveis linhas onde D_{mn} pode ser produzido.
 - $SL_{mn} = (\alpha_1, \dots, \alpha_k)$, k é a quantidade de linhas na seqüência e α_i é o número da linha na i -ésima posição da seqüência, onde $\alpha_i \in \{1, \dots, L\}$ e L é o número de linhas
- STk_{mn} : seqüência de possíveis tanques, onde o xarope de D_{mn} pode ser armazenado
 - $STk_{mn} = (\beta_1, \dots, \beta_k)$, k é a quantidade de tanques na seqüência e β_i é o número do tanque na i -ésima posição da seqüência, onde $\beta_i \in \{1, \dots, 2\bar{L}\}$ e \bar{L} é o número de tanques

Nessa representação, a demanda total de um produto, que deve ser produzida até o final da programação, é dividida em lotes menores (D_{mn}) com tamanhos aleatoriamente determinados. Esses lotes são distribuídos entre os genes nos períodos de T_l a T_m , cada um desses genes possuem uma seqüência (SL_{mn}) de linhas capazes de processá-lo. No caso do xarope, utilizado pelos produtos, o gene contém uma seqüência de tanques (STk_{mn}) com informações de como e onde ele será armazenado. Seja um tanque j , onde o xarope de P_{mn} será armazenado, β_i é obtido da seguinte forma em (2):

$$j = \begin{cases} \beta_i, & 1 \leq \beta_i \leq \bar{L} \\ \beta_i - \bar{L}, & \bar{L} < \beta_i \leq 2\bar{L} \end{cases} \quad (2)$$

Se $1 \leq \beta_i \leq \bar{L}$, o tanque $j = \beta_i$ será ocupado somente após o escoamento de todo o xarope previamente armazenado. Se $\bar{L} < \beta_i \leq 2\bar{L}$, o tanque $j = \beta_i - \bar{L}$ será ocupado imediatamente pelo xarope de P_{mn} . Todavia, algumas exceções se sobrepõem a esses critérios. Se j está vazio ou seu nível de ocupação não atingiu a capacidade mínima, então ele será ocupado imediatamente. Se o xarope armazenado em j for diferente do xarope de P_{mn} ou se o tanque estiver completamente ocupado, então o tanque não será imediatamente ocupado. O objetivo desses critérios é permitir ocupação parcial do tanque, sem ferir outras restrições do problema.

Definida a codificação dos indivíduos sua inicialização é feita de acordo com o pseudocódigo da Figura 3.4.

```

para cada macro-período  $t = 1, \dots, T$  faça
  repetir
    selecione aleatoriamente um produto  $i \in \{1, \dots, J\}$  com
    demanda  $d_{it} > 0$ 
    maxDem =  $d_{it}$ 
    enquanto maxDem > 0 faça
      selecione aleatoriamente um macro-período  $T_m \in \{1, \dots, t\}$ 
      determine aleatoriamente  $D_m \neq 0$  com  $D_m \in [0, \text{maxDem}]$ 
      insira  $D_m$  na primeira posição de  $T_m$ 
      maxDem = maxDem -  $D_m$ 
      para  $i = 1, \dots, k$  faça
        gerar aleatoriamente valores para  $SL_{mn}$  e  $STk_{mn}$  com
         $\alpha_i \in \{1, \dots, L\}$  e  $\beta_i \in \{1, \dots, 2L\}$ 
      fim para
    fim enquanto
  até que toda demanda em  $t$  tenha sido distribuída no indivíduo
fim para

```

Figura 3.4 - Inicialização dos Indivíduos

O exemplo a seguir mostra como funciona a inicialização de um cromossomo utilizando o algoritmo apresentado. Considere que existam duas linhas, dois tanques, dois produtos e seus respectivos xaropes. As demandas dos produtos são distribuídas em dois macro-períodos de acordo com a Tabela 3.1 abaixo. Dois indivíduos diferentes podem ser vistos na Figura 3.5, lembrando que ainda podem existir infinitas variações de cromossomos para esse exemplo.

	T ₁	T ₂
P ₁	150	100
P ₂	110	120

Tabela 3.1 - Demandas dos Produtos

		Indivíduo 1			Indivíduo 2			
T ₁		P ₁ 70	P ₂ 110	P ₁ 80	P ₁ 70	P ₁ 100	P ₂ 110	
		1 2 2 1 2 2 3 2	2 2 2 2 3 3 4 2	1 2 1 2 4 3 2 2	1 2 1 2 1 3 4 2	1 2 2 1 3 2 2 1	1 1 2 2 1 2 1 2	
T ₂		P ₁ 100	P ₂ 120		P ₂ 50	P ₁ 30	P ₁ 50	P ₂ 70
		1 1 2 2 4 2 1 3	1 1 1 2 1 3 3 2		2 2 1 2 1 2 3 2	1 1 2 2 1 2 3 3	1 2 1 1 4 2 4 2	2 1 2 2 1 2 3 1

Figura 3.5 - Exemplo de Dois Indivíduos Diferentes

Em ambos os indivíduos algumas demandas foram divididas entre mais de um gene, como é o caso da demanda de P₁ no primeiro macro-período do Indivíduo 1. Vale lembrar que em uma programação da produção é possível adiantar demandas, fazendo com que elas sejam produzidas e assim estocadas em macro-períodos antes do planejado. No indivíduo 2, por exemplo, foram produzidas 20 unidades a mais do que o esperado da demanda de P₁ em T₁, dessa maneira apenas 80 unidades de P₁ foram necessárias para atingir as 100 unidades necessárias em T₂.

Na seção seguinte será definido o processo de decodificação e avaliação do *fitness* dos indivíduos gerados.

3.3.2 Decodificação

A decodificação transforma a informação codificada em cada indivíduo em uma possível solução do problema. O algoritmo utilizado por Toledo (2005) aloca as demandas começando pelo primeiro alelo no último período de um indivíduo, o processo é feito dessa maneira, pois permite que a produção de uma demanda seja descolada para que ocorra em um macro-período anterior.

Cada alelo do indivíduo possui uma seqüência de pares (α_i, β_i) , $\alpha_i \in SL_m$ e $\beta_i \in STk_m$, que retorna uma linha e um tanque capazes de produzir o lote D_m . Se não há capacidade disponível ou se a capacidade permite apenas a produção

de parte do lote D_m , o próximo par $(\alpha_{i+1}, \beta_{i+1})$ é selecionado. O procedimento é repetido até que o último alelo no primeiro período tenha sido alcançado. A completa decodificação dos alelos retorna uma programação da produção sincronizada considerando os dois níveis do problema (linhas e tanques), porém podem ocorrer casos em que indivíduos infactíveis são gerados, isso ocorre quando uma demanda não é atendida completamente.

A seguir pode ser visto um pequeno exemplo de como é feita a decodificação do Indivíduo 1 da Figura 3.5, supondo que existam dois macro-períodos divididos em cinco micro-períodos cada. A decodificação começa pelo primeiro gene do último macro-período, o qual possui uma demanda de 100 unidades do produto 1. A alocação do tanque e da linha é feita utilizando-se o par $(\alpha_i, \beta_i) = (1, 4)$, ou seja, o produto deve ser alocado na linha 1 e no tanque 2, pois como $\bar{L} < \beta_i \leq 2\bar{L}$, então $j = \beta_i - \bar{L}$ (regra (2) vista na Seção 3.3.1). Como o tanque está vazio um novo lote do xarope do produto 1 (X_{p1}) é criado e inserido imediatamente no tanque 2, supondo que o tempo de preparo do tanque seja de um micro-período, esse processo ocorre antes do início da produção de P_1 . Como foi possível inserir todo o lote de P_1 os outros pares da seqüência não são necessários, na Figura 3.6 é possível observar como a decodificação ocorre graficamente.

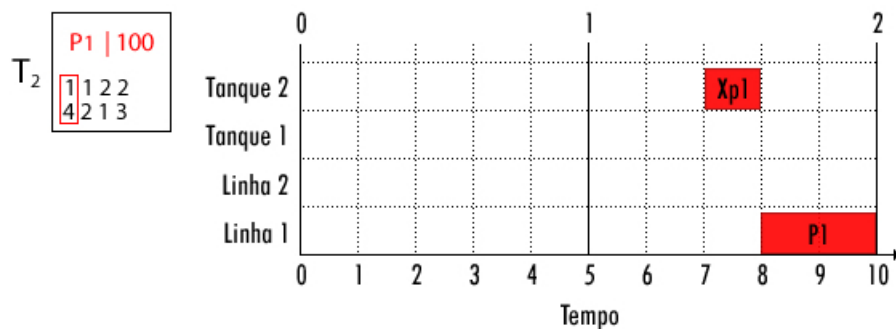


Figura 3.6 - Decodificação do Primeiro Gene em T_2

O próximo gene decodificado é do produto 2, seu xarope (X_{p2}) é inserido imediatamente no tanque 1 ($\beta_i = 1$) um micro-período antes do início de sua produção, a qual ocorre na linha 1 ($\alpha_i = 1$), porém como o produto processado em seguida é diferente é necessário um tempo de preparo entre eles (Figura 3.7).

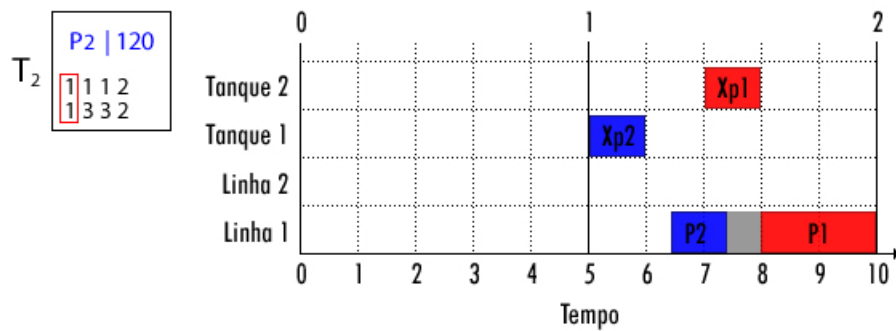


Figura 3.7 - Decodificação do Segundo Gene em T_2

A decodificação segue com o primeiro gene do macro-período T_1 , o lote de 70 unidade de P_1 é inserido na linha 1 e uma nova atribuição é feita ao tanque 2 (Figura 3.8).

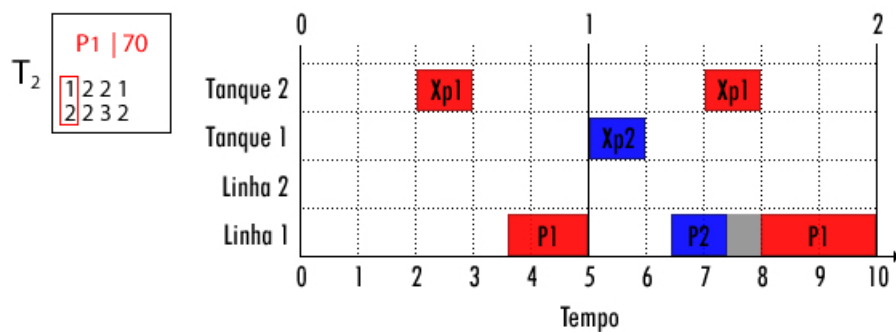


Figura 3.8 - Decodificação do Primeiro Gene em T_1

O segundo gene em T2 provoca a antecipação do preparo do xarope 2, pois como $\beta_i = 3$ o novo lote deverá fazer parte do lote já existente no tanque 1 (como $\bar{L} < \beta_i \leq 2\bar{L}$, então $\beta_i - \bar{L} = 3 - 2 = 1$) desde que nenhuma das restrições impostas na Seção 3.3.1 seja desobedecida. Nesse caso o tanque será reabastecido com um mesmo xarope e é suposto que exista capacidade o suficiente para ele (Figura 3.9).

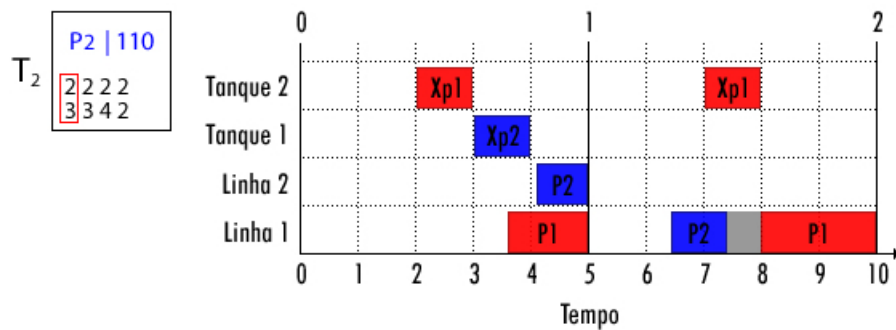


Figura 3.9 - Decodificação do Segundo Gene em T₁

No último gene, com $\beta_i = 4$, o lote existente do xarope 1 no tanque 2 deve ser antecipado. Supondo que o tanque possui capacidade para produzir apenas 50 unidades do produto 1, o segundo par (α_i, β_i) deve ser consultado para que o resto do lote possa ser programado. Dessa forma as 30 unidades restantes são produzidas na linha 2 e como o tanque 1 está ocupado com o Xp2 um novo lote de Xp1 deve ser criado. Na Figura 3.10 pode ser vista a decodificação completa do indivíduo.

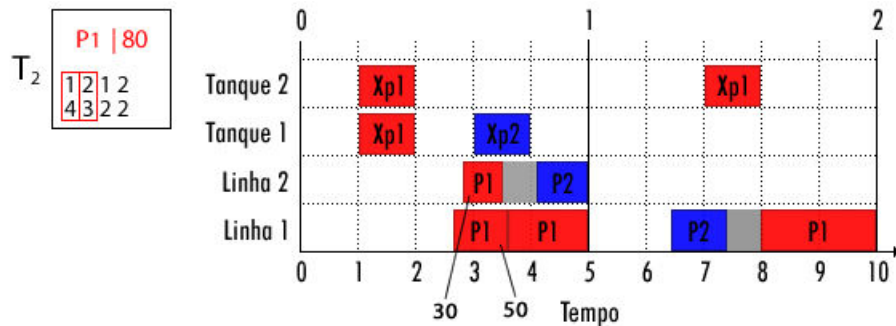


Figura 3.10 - Decodificação do Terceiro Gene em T_1

Através da decodificação é possível estabelecer a programação completa das linhas e dos tanques, ou seja, quando ocorre o início e o fim das atividades de criação dos lotes de xaropes e produtos, respeitando tempos de troca e de processamento. A avaliação do indivíduo é feita através da função de *fitness* vista na Seção 2.5.

3.4 Operadores Genéticos

Os operadores genéticos são responsáveis pela troca de informação entre os indivíduos da população e por sua diversidade. Nas próximas subseções serão apresentados os tipos de recombinação (*crossover*) avaliados e as mutações utilizadas.

3.4.1 Recombinação

A primeira recombinação (*crossover*) avaliada é do tipo uniforme, onde os pais são percorridos simultaneamente por macro-período sorteando com uma mesma probabilidade, se o filho receberá o gene do pai 1 ou do pai 2. Se um pai tiver mais genes que o outro, o processo continua normalmente, porém apenas

sorteando se o filho recebe o gene desse pai ou não. Na Figura 3.11 os genes sombreados foram os escolhidos durante o cruzamento.

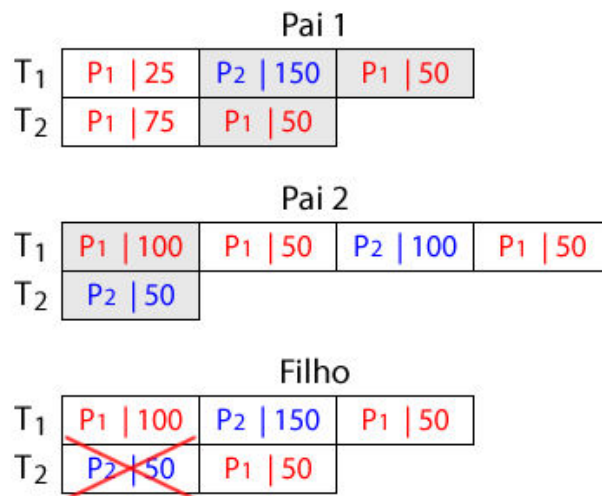


Figura 3.11 - Crossover Uniforme

A seqüência de linhas e tanques é ignorada na figura, pois o filho recebe a mesma informação vinda dos pais. Durante o processo ocorre também uma verificação para saber se as demandas estão sendo atendidas ou se estão em excesso. Na Figura 3.11 o gene com 50 unidades do produto 2 foi escolhido, porém sua cópia foi cancelada, pois a demanda ultrapassaria o total necessário. Ao fim do cruzamento todas as demandas de cada produto são verificadas e ocorre um reparo se alguma delas não tiver sido completamente atendida. O processo de reparo é feito adicionando-se a quantidade faltante a algum dos genes com o mesmo produto. Se não houver tal produto, um novo gene é inserido com a demanda faltante no respectivo macro-período.

O segundo *crossover* utilizado é o de um ponto, onde o cruzamento é feito a cada macro-período e o ponto de corte é escolhido aleatoriamente. O tamanho do pai que possui mais genes naquele macro-período é considerado. Se

o ponto de corte não estiver além do tamanho do pai com menos genes, o *crossover* ocorre normalmente, onde a primeira parte copiada pode ser do pai 1, a segunda do pai 2 e vice-versa. Do contrário, o pai menor é copiado completamente e os alelos do pai com mais genes são copiados a partir do ponto de corte sorteado. As duas situações podem ser vistas na Figura 3.12, os genes sombreados foram os escolhidos durante o cruzamento.

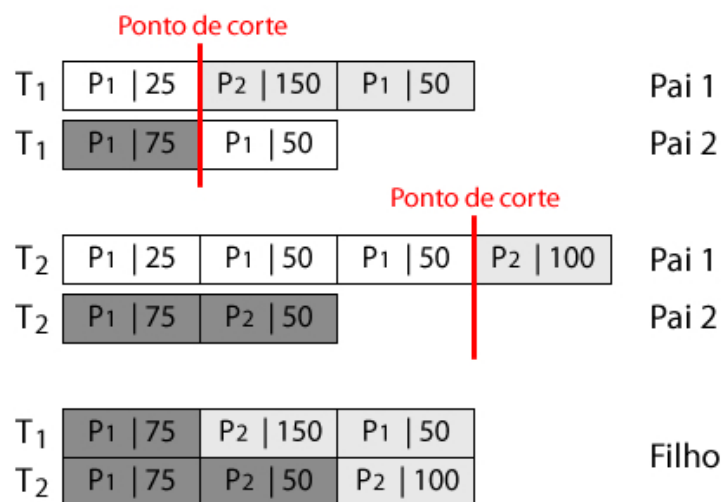


Figura 3.12 - Crossover de Um Ponto

Caso não exista nenhum gene em algum macro-período de um dos pais o outro pai é copiado integralmente. Da mesma maneira que no *crossover* uniforme, o reparo pode ocorrer no *crossover* de um ponto e no de dois pontos, que será visto a seguir.

O *crossover* de dois pontos é semelhante ao anterior, o que varia é que existem três situações durante o sorteio dos pontos de corte. A primeira situação é quando a posição dos dois pontos é menor que o tamanho do pai com menos genes, nesse caso o *crossover* ocorre normalmente, sendo que a primeira parte pode ser proveniente do pai 1, a segunda do pai 2, a terceira do pai 1 e vice-

versa. Na segunda situação apenas o segundo ponto de corte ultrapassa o tamanho do pai menor, dessa forma a primeira parte é copiada do pai menor até o primeiro ponto de corte e a segunda é copiada do pai maior a partir desse ponto até o segundo, sem que haja uma terceira parte. Na última situação ambos os pontos de cortes estão em posições fora do pai menor, o qual é copiado totalmente, e para o pai maior a cópia dos genes começa do primeiro ponto de corte até seu fim, nessas duas últimas situações é como se ocorresse um *crossover* de um ponto.

3.4.2 Mutação

A mutação é um operador genético que ajuda a garantir a diversidade da população, fazendo pequenas alterações nos indivíduos através da troca de posição entre seus genes.

Três tipos de mutação foram utilizados e todas ocorrem com a mesma probabilidade. A primeira apenas escolhe aleatoriamente dois genes que estão em um mesmo macro-período para trocá-los de posição, Figura 3.13.

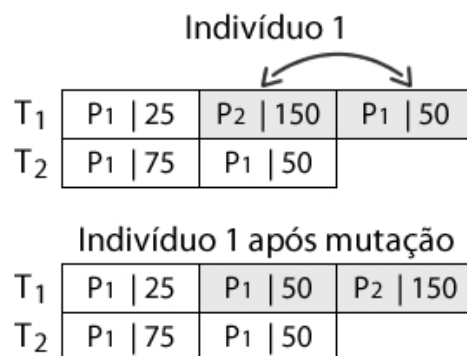


Figura 3.13 - Primeiro Tipo de Mutação

O segundo tipo de mutação escolhe aleatoriamente um gene e o insere em outra posição, esse processo pode ocorrer dentro de um mesmo macro-período ou em macro-períodos diferentes. No último caso, o macro-período de inserção do alelo deve ser anterior ao original para que a demanda não seja atendida com atraso e o indivíduo seja ineficaz (Figura 3.14).

Indivíduo 1				
T ₁	P ₁ 25	P ₂ 150	P ₁ 50	
T ₂	P ₁ 75	P ₁ 50		

Indivíduo 1 após mutação				
T ₁	P ₁ 25	P ₁ 50	P ₁ 50	P ₂ 150
T ₂	P ₁ 75			

Figura 3.14 - Segundo Tipo de Mutação

A última mutação é a troca de posição entre genes de macro-períodos diferentes, onde deve ser tomado o mesmo cuidado da mutação anterior. Por exemplo, se a tabela com a demanda dos produtos fosse a Tabela 3.2 a seguir, a troca dos genes indicados na Figura 3.15 seria ineficaz, pois 100 unidades de P₂ deveriam ter sido atendidas em T₁ e só foram produzidas de fato em T₂.

	T ₁	T ₂
P ₁	75	50
P ₂	100	50

Tabela 3.2 - Demandas dos Indivíduo 1

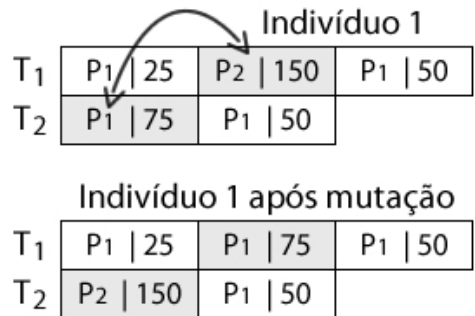


Figura 3.15 - Terceiro Tipo de Mutação

Dessa maneira, esse tipo de alteração não é executada em um indivíduo selecionado para mutação.

3.5 Estruturas Populacionais

O principal objetivo do presente trabalho foi implementar novas estruturas populacionais no ambiente Np-Opt, proposto por Mendes (2003), e avaliar o desempenho do algoritmo genético para o Problema Integrado de Dimensionamento de Lotes e Programação da Produção. Nas próximas subseções serão vistos cada tipo de estrutura avaliada.

3.5.1 População Não Estruturada

Populações não estruturadas são utilizadas em algoritmos genéticos clássicos, onde não existe nenhum tipo de estruturação conectando os indivíduos, apenas uma lista onde eles são ordenados do mais apto para o menos apto, como na Figura 3.16.

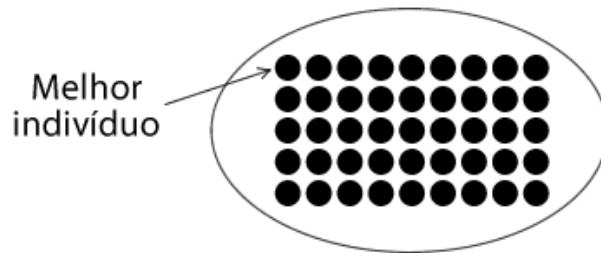


Figura 3.16 - População Não Estruturada

Nesse caso o operador genético de seleção utilizado é totalmente livre, ou seja, os pais são escolhidos aleatoriamente na população desde que eles sejam diferentes. Não é levado em conta nenhum tipo de quantificação que selecione apenas os indivíduos mais aptos e também não há elitismo (manutenção do melhor indivíduo a cada geração).

3.5.2 População Estruturada em Hierarquia

Nas populações estruturadas os indivíduos são dispostos considerando uma estrutura hierárquica. No presente trabalho esta estruturação se dá por meio de árvores. Dentro de cada estrutura há diversos nichos ou *clusters*, onde a hierarquia é definida por um nó líder e seus seguidores. O líder é o indivíduo com melhor valor de *fitness* quando comparado aos seus seguidores. Por exemplo, numa árvore binária um *cluster* tem tamanho 3 e numa árvore ternária cada *cluster* tem tamanho 4 (Figura 3.17).

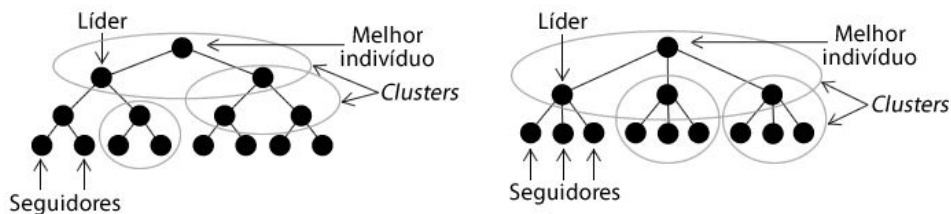


Figura 3.17 - Estruturas com Clusters de Tamanho 3 e 4

Neste trabalho, foram consideradas populações estruturadas em árvores com *clusters* de tamanho 3 (dois nós filhos por líder) a 11 (10 nós filhos por líder).

A seleção de dois indivíduos pais na execução do operador de recombinação, para o caso estruturado, é feita inicialmente selecionando aleatoriamente um *cluster* da estrutura. Em seguida, o melhor indivíduo desse *cluster* é sempre escolhido para ser um dos pais juntamente com um dos seus seguidores, que será o segundo pai, também aleatoriamente selecionado. O novo indivíduo gerado será inserido na população se apresentar valor de *fitness* melhor que um dos seus pais. Todavia, a população no caso estruturado deve manter a hierarquia entre os indivíduos. Por isso, uma reestruturação da população se faz necessária após a inserção de novos indivíduos.

O processo de reestruturação será exemplificado no contexto de uma população estruturada em árvore binária na Figura 3.18. O indivíduo filho será inserido no lugar de um dos pais, caso seu valor de *fitness* seja melhor.

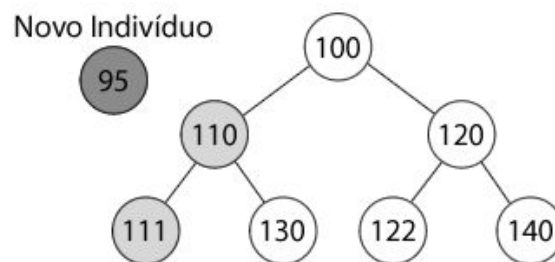


Figura 3.18 - População Antes da Inserção do Novo Indivíduo

Supondo que seja um problema de minimização, onde o pior *fitness* é aquele com maior valor numérico, um melhor indivíduo com valor de *fitness* 110 é aleatoriamente selecionado bem como seu seguidor com valor 111. O filho resultante dessa combinação apresenta um *fitness* de 95. O novo indivíduo será

inserido no lugar do pai com pior valor de *fitness*. A Figura 3.19 apresenta um *cluster* após a inserção do novo indivíduo.

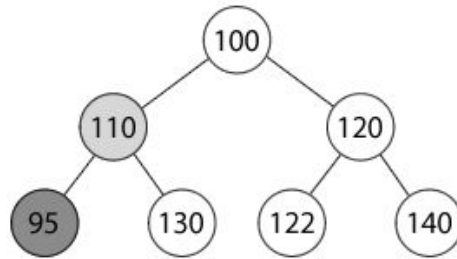


Figura 3.19 - População Após Inserção do Novo Indivíduo

A fase de reestruturação da população ocorre após as recombinações. Todos os *clusters* são percorridos e se algum dos seguidores apresentar melhor *fitness* que um líder, uma troca de posições irá ocorrer. A Figura 3.20 ilustra essa idéia dentro do *cluster* mais interno da população e o processo segue a mesma idéia na população como um todo (Figura 3.21).

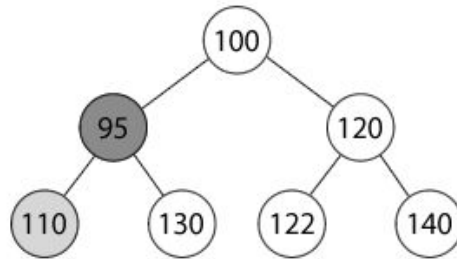


Figura 3.20 - Restauração da População – Passo 1

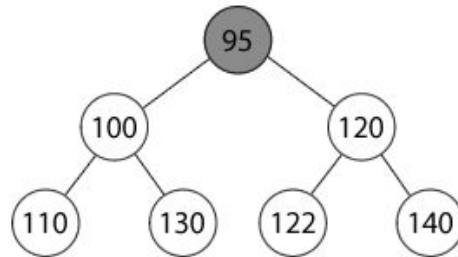


Figura 3.21 - Restauração da População – Passo 2

3.5.3 Abordagens Uni e Multipopulacional

Além das duas diferentes formas de estruturar a população, o presente trabalho avalia o desempenho do algoritmo genético para uma população (abordagem uni populacional) e também quando elas são separadas em duas ou mais populações (abordagem multipopulacional). No caso da abordagem multipopulacional foram testadas 2, 3 e 4 populações com um total de 60, 120 e 180 indivíduos distribuídos igualmente por elas.

Nas abordagens multipopulacionais, o operador de migração irá ocorrer após a convergência das populações. A convergência da população ocorre quando nenhum novo indivíduo é inserido, após as execuções dos operadores genéticos. Ao terminar de percorrer todas as populações, o operador de migração realiza a inserção de uma cópia do melhor indivíduo de determinada população na população seguinte. Essa cópia do melhor indivíduo é inserida no lugar de um indivíduo aleatoriamente selecionado, excetuando o melhor indivíduo. A Figura 3.22 exemplifica a migração considerando multipopulações, uma não estruturada e a outra com estrutura ternária.

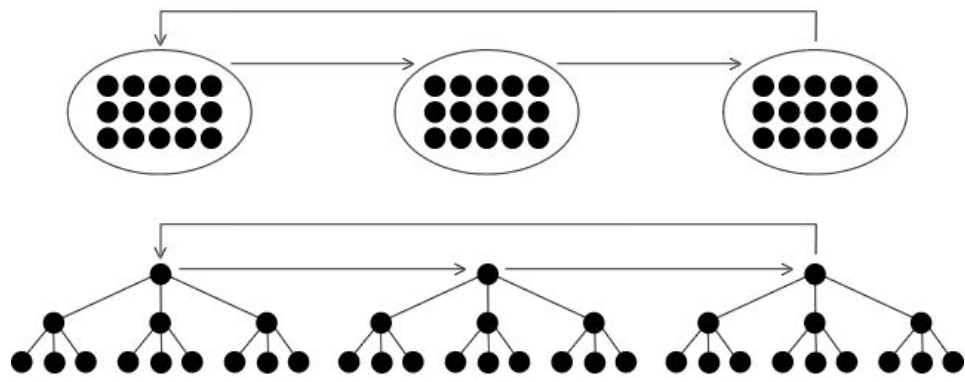


Figura 3.22 - Execução da Migração Entre Populações

4 RESULTADOS COMPUTACIONAIS

As diversas abordagens propostas foram avaliadas em um conjunto de 6 instâncias com dimensões industriais. Essas instâncias coletadas por Toledo *et al.* (2007) foram definidas a partir de programações reais da produção executadas por uma fábrica de refrigerantes. Uma combinação de parâmetros do problema define cada conjunto de instâncias. Os parâmetros são: número de linhas (L), tanques (\bar{L}), produtos (J), xaropes (\bar{J}) e o número de macro-períodos (T). Cada combinação apresentada na Tabela 4.1 se refere a uma única instância.

Comb.	$L/\bar{L}/J/\bar{J}/T$	Comb.	$L/\bar{L}/J/\bar{J}/T$
A1	5/9/33/11/1	B1	6/10/52/19/1
A2	6/9/49/14/2	B2	6/10/56/19/2
A3	6/9/58/15/3	B3	6/10/65/21/3

Tabela 4.1 - Parâmetros das instâncias industriais (Fonte: Toledo *et al.*, 2007)

Um período de produção corresponde a 7 dias de trabalho nas instâncias do tipo A e 10 dias nas instâncias do tipo B. As abordagens propostas para o Algoritmo Genético foram executadas 10 vezes em cada instância, durante meia hora, utilizando uma taxa de mutação de 0,7 e taxa de recombinação de 1,5. Os testes foram realizados em computadores Core 2 Duo, 2.0 GHz; com 2 GB de memória RAM.

A equação (3) na página seguinte foi usada para medir os desvios percentuais médios obtidos pelos AGs em relação ao custo estimado pela fábrica de refrigerantes.

$$Desvio(\%) = 100 \frac{(Z - Z^*)}{Z^*} \quad (3)$$

O parâmetro Z representa o valor médio obtido nas soluções finais de cada uma das 10 execuções do AGs, enquanto Z^* representa o custo estimado da programação da produção executada pela fábrica. O cálculo de Z é baseado na função objetivo apresentada no final da Seção 2.5, onde são somados todos os custos de troca, estoque e produção da solução encontrada pelo algoritmo genético.

O primeiro teste computacional avaliou diversos tipos de estruturas populacionais, a quantidade de indivíduos utilizados em cada população e o número de populações. As Tabelas 4.2, 4.3 e 4.4 a seguir apresentam os resultados obtidos considerando populações com 60, 120 e 180 indivíduos, distribuídos entre 1, 2, 3 e 4 populações. O tamanho do *cluster* em cada tabela representa o número de indivíduos (nós) que cada um desses *clusters* possui. Assim, uma população com tamanho de cluster igual a 3 contém no máximo 1 nó líder e 2 seguidores em cada cluster, ou seja, trata-se de uma estrutura em árvore binária. Da mesma forma, um cluster de tamanho 4 contém 1 líder e 3 seguidores, ou seja, os indivíduos estão estruturados em uma árvore ternária. Os números destacados nas entradas da tabela revelam os melhores desvios encontrados. Desvios percentuais negativos indicam que os AGs encontraram custos médios melhores do que aqueles estimados pela fábrica.

Em praticamente todos os casos o algoritmo genético foi melhor que as soluções obtidas pela fábrica de refrigerantes. A única exceção foi a abordagem com uma única população não estruturada, utilizando 60 indivíduos.

		Tamanho do <i>Cluster</i> (População Estruturada)								
	Não Estrut.	3	4	5	6	7	8	9	10	11
1 Pop.	0,1904	-3,0145	-3,2930	-3,1086	-3,2579	-3,3771	-3,3209	-3,3838	-3,4315	-3,3116
2 Pop.	-1,3258	-2,8472	-3,2769	-3,0332	-3,2396	-3,2131	-3,3860	-3,2358	-3,3875	-3,3526
3 Pop.	-0,6835	-2,6621	-3,0764	-2,9855	-3,2610	-3,2060	-3,2114	-3,3605	-3,2184	-3,2828
4 Pop.	-1,0797	-2,9841	-3,0152	-3,1355	-3,1071	-2,9121	-3,3634	-3,3540	-3,2875	-2,9656
Média	-0,7246	-2,8770	-3,1654	-3,0657	-3,2164	-3,1771	-3,3204	-3,3335	-3,3312	-3,2282

Tabela 4.2 - Resultados com 60 Indivíduos

		Tamanho do <i>Cluster</i> (População Estruturada)								
	Não Estrut.	3	4	5	6	7	8	9	10	11
1 Pop.	-0,4232	-2,9608	-3,1126	-3,2531	-3,3107	-3,2798	-3,3577	-3,3942	-3,3589	-3,3656
2 Pop.	-0,3280	-2,8996	-3,2275	-3,2219	-3,3204	-3,2480	-3,3466	-3,4819	-3,3546	-3,4263
3 Pop.	-0,2874	-2,9898	-3,1721	-3,1590	-3,1388	-3,3165	-3,2234	-3,4858	-3,2258	-3,2959
4 Pop.	-0,3789	-3,0441	-3,1345	-3,2182	-3,3356	-3,3098	-3,2935	-3,4463	-3,3076	-3,3478
Média	-0,3544	-2,9736	-3,1617	-3,2130	-3,2764	-3,2885	-3,3053	-3,4521	-3,3117	-3,3589

Tabela 4.3 - Resultados com 120 Indivíduos

		Tamanho do <i>Cluster</i> (População Estruturada)								
	Não Estrut.	3	4	5	6	7	8	9	10	11
1 Pop.	-0,3051	-2,8517	-3,0775	-3,1116	-3,2896	-3,3890	-3,2356	-3,3576	-3,3062	-3,3315
2 Pop.	-0,5742	-3,1152	-3,3058	-3,2933	-3,3437	-3,2413	-3,2556	-3,4456	-3,5008	-3,3802
3 Pop.	-0,5887	-3,2032	-3,2644	-3,4570	-3,4189	-3,3907	-3,4192	-3,3526	-3,3788	-3,5007
4 Pop.	-0,4663	-3,1479	-3,1980	-3,3885	-3,4080	-3,2959	-3,3599	-3,4373	-3,4712	-3,3864
Média	-0,4836	-3,0795	-3,2114	-3,3126	-3,3650	-3,3292	-3,3176	-3,3983	-3,4143	-3,3997

Tabela 4.4 - Resultados com 180 Indivíduos

Em relação ao tipo de população, estruturada ou não estruturada, observou-se que independentemente da quantidade de indivíduos, do número de populações e do tamanho do *cluster*, a abordagem com os indivíduos estruturados em árvores foi melhor em todos os experimentos. Os desvios

variaram de -3,5008% a -2,6621%. Além disso, há uma tendência de melhora conforme aumenta o número de indivíduos. Isso é destacado na Figura 4.1 que considera a média dos resultados para as 4 populações em cada cluster. Observe que a população com 180 indivíduos apresenta os melhores desvios em todos os tipos de *cluster*, exceto no *cluster* de tamanho 9. Todavia, o melhor desvio médio é obtido com tamanho de cluster 9 com 120 indivíduos.

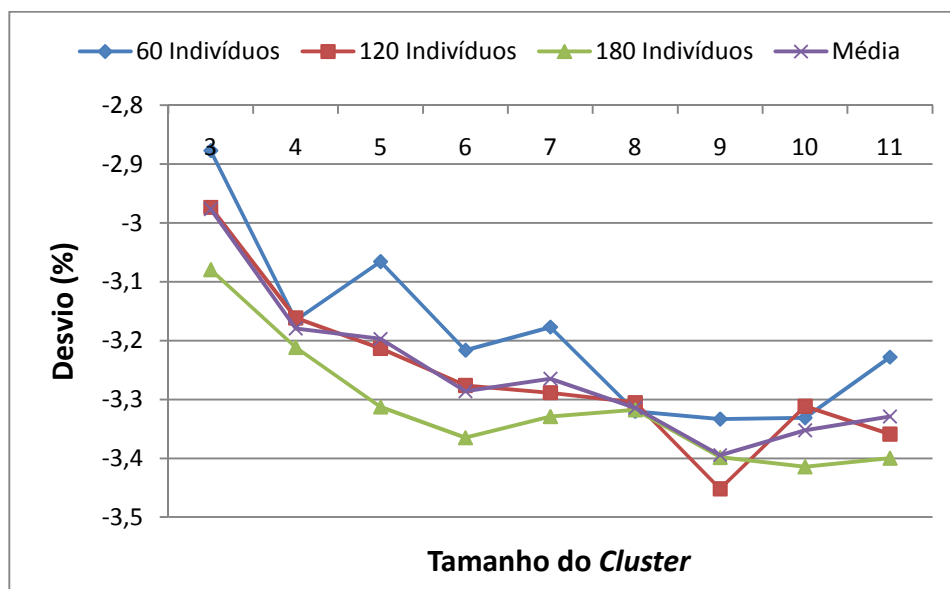


Figura 4.1 - Gráfico das Médias por Quantidade de Indivíduos

As Figuras 4.2 e 4.3 apresentam uma análise da variação da quantidade de populações e de indivíduos separados em população não estruturada (Figura 4.2) e estruturada (Figura 4.3). Os melhores resultados obtidos considerando o número de populações e o número de indivíduos são ilustrados. No caso estruturado, apenas o tamanho do cluster que retornou o melhor resultado é apresentado para cada população e número de indivíduos.

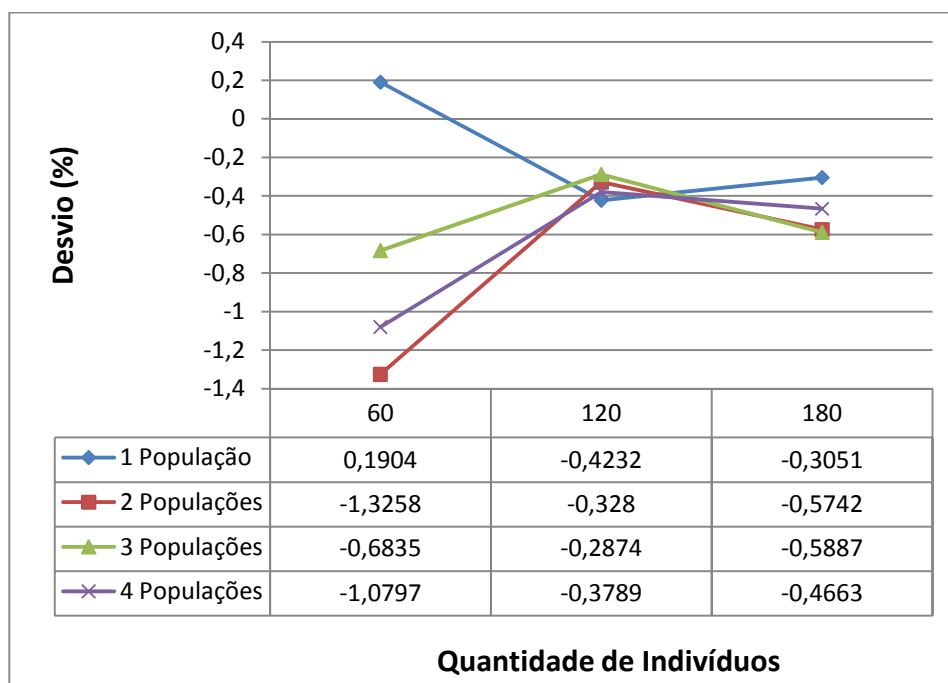


Figura 4.2 - Gráfico das Melhores Soluções com População Não Estruturada

No gráfico da Figura 4.2, observa-se que os valores obtidos pelos desvios pioram com o aumento do número de indivíduos em cada uma das variações de população consideradas. A alocação de 60 indivíduos em 2 populações retornou o melhor desvio para o caso não estruturado. Quando a quantidade de indivíduos aumenta para 120, a separação desses indivíduos em até 4 populações praticamente não altera os valores de desvio obtidos. Na população com 180 indivíduos, a separação em 2 ou 3 populações apresentam o mesmo desvio, sendo melhor do que aquele obtido com 120 indivíduos, mas pior que os obtidos com 60 indivíduos.

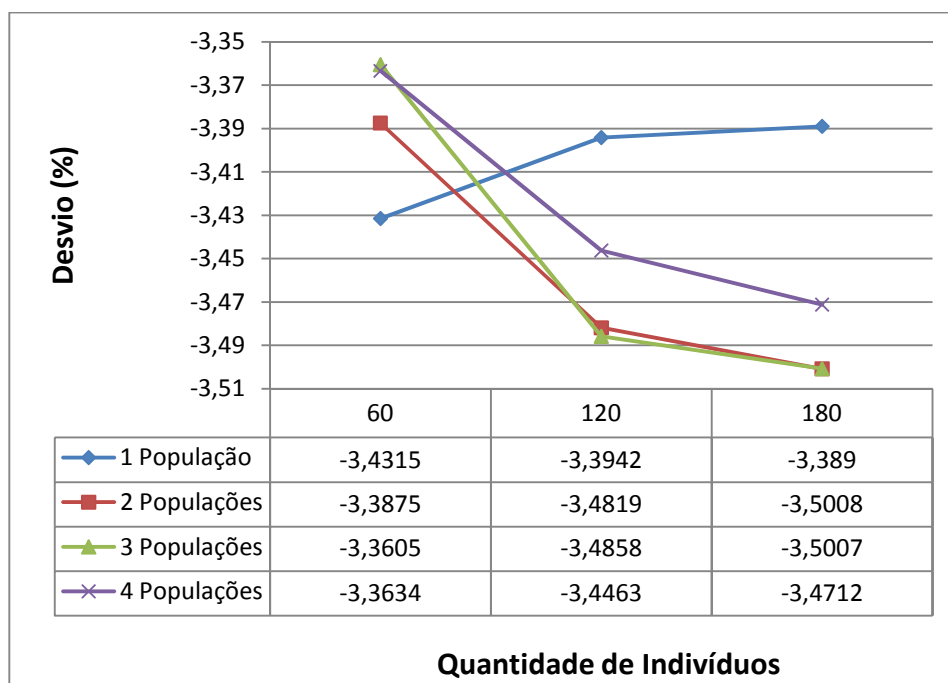


Figura 4.3 - Gráfico das Melhores Soluções com População Estruturada

Na Figura 4.3, a divisão dos indivíduos em 2 e 3 populações também apresenta melhores resultados. Por outro lado, essa melhora vem acompanhada pelo aumento no número de indivíduos. Inicialmente, ocorre o inverso dos resultados com população não estruturada, pois com poucos indivíduos a divisão em duas ou mais populações estruturadas piora os resultados. A população estruturada também apresenta comportamento diferente do caso não estruturado quando o número de indivíduos aumenta. Há piora nos resultados com 1 população e melhora com 2, 3 e 4 populações. Neste caso, os 180 indivíduos divididos em 2 e 3 populações retornam os melhores desvios. Pode-se concluir até esse ponto que a evolução dos indivíduos de forma independente em populações não estruturadas, nas instâncias deste problema estudado, parece perder desempenho com o aumento no número de indivíduos. Por outro lado, o

mesmo processo de evolução independente parece se beneficiar do aumento no tamanho da população, quando os indivíduos estão organizados em clusters.

Para o segundo experimento, as abordagens que retornaram as quatro melhores soluções também foram executadas utilizando dois outros tipos de *crossover*: um ponto e dois pontos A configuração utilizada nas abordagens que retornaram os melhores resultados são apresentadas a seguir:

- S1 - 120 Indivíduos, 2 Populações e *Cluster* de Tamanho 9
- S2 - 120 Indivíduos, 3 Populações e *Cluster* de Tamanho 9
- S3 - 180 Indivíduos, 3 Populações e *Cluster* de Tamanho 11
- S4 - 180 Indivíduos, 2 Populações e *Cluster* de Tamanho 10

Todas essas abordagens foram inicialmente executadas utilizando *crossover* uniforme. Assim, as Tabelas 4.5 e 4.6 comparam os resultados obtidos com os diferentes *crossovers* avaliados.

Instância	S1			S2		
	Uniforme	Um Ponto	Dois Pontos	Uniforme	Um Ponto	Dois Pontos
A1	-2,3026	-2,3913	-2,4799	-2,5154	-2,4267	-2,4090
A2	-4,8940	-5,0400	-5,0743	-5,1595	-5,0958	-5,0718
A3	-3,6302	-3,6850	-4,0112	-3,4719	-3,7387	-3,8666
B1	-2,8472	-2,8560	-2,9626	-2,6607	-2,9448	-2,8205
B2	-3,5216	-3,3655	-3,3347	-3,3341	-3,7353	-3,5609
B3	-3,6959	-4,1060	-4,0274	-3,7735	-3,7869	-3,9558
Média	-3,4819	-3,5740	-3,6483	-3,4858	-3,6214	-3,6141

Tabela 4.5 - Resultados de S1 e S2

Instância	S3			S4		
	Uniforme	Um Ponto	Dois Pontos	Uniforme	Um Ponto	Dois Pontos
A1	-2,4267	-2,4622	-2,3204	-2,4267	-2,3913	-2,4445
A2	-4,9141	-5,1067	-5,1522	-4,9908	-5,1563	-4,9173
A3	-3,8600	-3,7889	-4,0180	-3,7798	-3,9507	-3,9926
B1	-2,9004	-2,9182	-2,8649	-2,8649	-2,9537	-2,9004
B2	-3,3151	-3,2899	-3,6986	-3,3699	-3,5953	-3,5739
B3	-3,5878	-4,1330	-4,2796	-3,5729	-3,9910	-4,1208
Média	-3,5007	-3,6165	-3,7223	-3,5008	-3,6730	-3,6582

Tabela 4.6 - Resultados de S3 e S4

Considerando o número de vitórias como outra medida de desempenho, o *crossover* uniforme superou os demais nas instâncias industriais A1 e A2 em S2, já o *crossover* de um ponto foi melhor na instância B2 também com a abordagem S2. O *crossover* de dois pontos foi o melhor em 3 instâncias (1 com a abordagem S1 e 2 com a abordagem S3). A distribuição das vitórias pode ser vista no gráfico da Figura 4.4 a seguir:

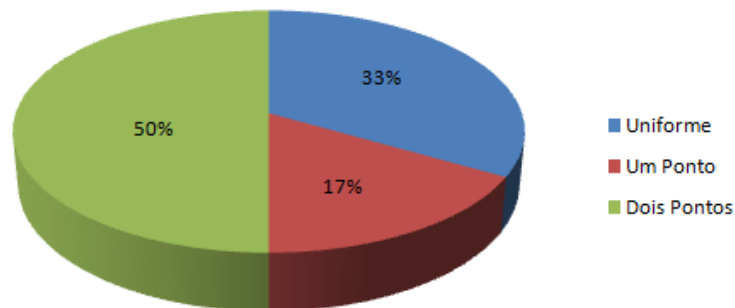


Figura 4.4 - Distribuição das Vitórias por Tipo de Crossover

Considerando a média dos desvios das soluções obtidas em todas as instâncias por cada abordagem como outra medida de desempenho, o *crossover* uniforme é superado pelo *crossover* de um e dois pontos (Figura 4.5).

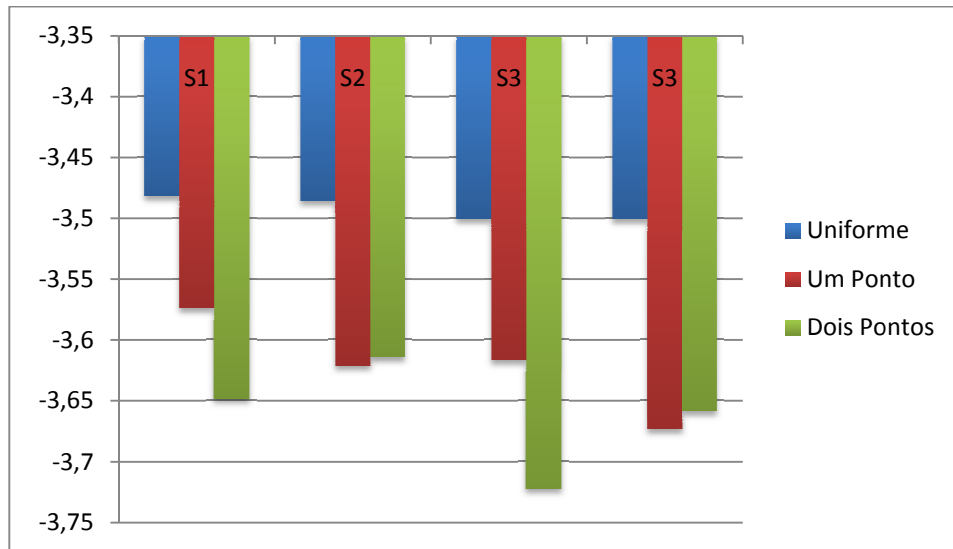


Figura 4.5 - Média das Soluções

Os resultados obtidos demonstram que a abordagem S3 (180 indivíduos, 3 populações e *cluster* de tamanho 11), associada ao *crossover* de dois pontos, apresenta o melhor desempenho entre a média das soluções nas instâncias avaliadas. Além disso, essa abordagem obteve o melhor resultado em duas instâncias (A3 e B3) considerando as médias dos desvios por instâncias.

5 CONSIDERAÇÕES FINAIS

5.1 Conclusões

O presente trabalho apresentou e avaliou o desempenho de diversas abordagens populacionais em um algoritmo genético aplicado ao Problema Integrado de Dimensionamento de Lotes e Programação da Produção (PIDLPP). Trata-se de um problema multinível com restrições de capacidade, custos e tempos de preparo. Além disso, novos tipos de operadores de recombinação também foram avaliados quando aplicados a codificação proposta para os indivíduos neste problema. Entre as abordagens avaliadas nos testes com as instâncias industriais coletadas em fábrica de bebidas a configuração vencedora foi S3 com 180 indivíduos, 3 populações, *cluster* de tamanho 11 e com recombinação (*crossover*) de dois pontos.

O primeiro resultado relevante indica que as estruturas hierárquicas, ou seja, populações dispostas em árvores superam as abordagens não estruturadas nas instâncias industriais aqui consideradas para o PIDLPP. Quando a população está hierarquicamente organizada em *clusters*, a seleção e a inserção de novos indivíduos ocorre de maneira mais criteriosa. Além disso, o processo de evolução utilizando várias populações permite a exploração de diferentes regiões do espaço de busca simultaneamente. Os procedimentos de reinicialização e migração ajudam esse tipo de abordagem a escapar de mínimos locais ou convergência prematura. Porém, nem sempre recorrer à divisão da população é o melhor método, pois o tamanho da população e o tipo de estrutura utilizada podem influenciar nos resultados. Como foi observado nos experimentos, ocorreu um comportamento contrário entre as populações estruturadas e não estruturadas quando se aumentou o número de indivíduos e de populações.

O segundo resultado foi obtido ao se avaliar diferentes tipos de recombinação aplicados à codificação proposta para um indivíduo no PIDLPP.

Operadores mais simples de recombinação como o cruzamento de um ponto ou dois pontos ajudaram o método a retornar melhores resultados. Isso se explica pelo fato desses operadores conseguirem manter uma maior integridade do indivíduo. Eles realizam menos trocas entre alelos, preservando uma boa estrutura. Dessa forma, também é reduzida a necessidade de se recorrer a reparos, causados pela perda de informação no processo de recombinação. A recombinação de dois pontos mostrou ser a mais eficaz na resolução de instâncias industriais, superando a recombinação uniforme previamente utilizada por Toledo (2005).

5.2 Contribuições

Os resultados obtidos reforçaram a utilização e a importância das heurísticas nas ferramentas de apoio a tomada de decisão em problemas de alta complexidade, como o dimensionamento de lotes e programação da produção. Esses tipos de problemas demandam muito tempo ou nem conseguem ser resolvidos em tempo hábil por métodos exatos e acabam tornando-se inviáveis para a indústria em geral. Além disso, em sua grande maioria, os resultados encontrados foram melhores que os custos estimados pela fábrica. Isso representa na prática uma diminuição de gastos desnecessários com a produção, como por exemplo, desperdício de matéria prima, tempos de preparação, custos de estocagem, etc. o que acaba refletindo em menos impacto ao meio ambiente, queda dos preços para o consumidor final, entre outros fatores.

O presente trabalho também contribuiu para a ampliação de uma área pouco explorada na literatura científica, que é a utilização de algoritmos genéticos multipopulacionais e estruturados. Os testes realizados poderão ajudar nas decisões de quais tipos de estruturas e populações devem ser utilizadas em

novos trabalhos dessa área ou em diferentes áreas que envolvam problemas combinatoriais de difícil resolução por métodos exatos.

5.3 Trabalhos Futuros

São propostas para trabalhos futuros os tópicos listados a seguir:

- Fazer análises comparativas envolvendo os tempos computacionais para as abordagens avaliadas
- Desenvolver uma nova codificação para os indivíduos que seja tão representativa quanto à apresentada, mas que não demande tanto esforço computacional durante sua decodificação;
- Utilizar heurísticas de busca local na resolução do mesmo problema ou associar essas heurísticas ao algoritmo genético, propondo assim algoritmos conhecidos como meméticos.
- Testar outros tipos de mutação e migração combinando-os com os resultados do presente trabalho;
- Testar o uso de técnicas de programação paralela e distribuída no processo evolutivo, por exemplo, executando a evolução da população em diferentes processadores.
- Associar o uso de algoritmos genéticos (AG) a técnicas de programação matemática, onde o AG determina variáveis inteiras e o método exato solucione o modelo contínuo relacionado.
- Realizar testes com as instâncias artificiais apresentadas por Toledo (2005) e também com outras instâncias industriais coletadas em fábricas de refrigerantes;
- Verificar a validade e a viabilidade dos resultados obtidos aplicando-os em empresas reais.

REFERÊNCIAS BIBLIOGRÁFICAS

ABIR, **Dados de Mercado-Refrigerantes**, Associação Brasileira da Indústria de Refrigerantes e Bebidas Não Alcoólicas. Disponível em: <http://www.abir.org.br>, acessado em: 20 de Março de 2009.

ABOUT.COM, **A Timeline of Soft Drink History**, About.com: Inventors. Disponível em: <http://inventors.about.com/library/weekly/aa091699.htm>, acessado em: 20 de Março de 2009.

ATMAR, W., **On the Rules and Nature of Simulated Evolutionary Programming**, Proc. of the First Ann. Conf. On Evolutionary Programming, pp. 17-26, 1992.

BÄCK T., FOGEL D. B., MICHALEWICZ Z., **Evolutionary Computation 2: Advanced Algorithms and Operators**, Institute of Physics Publishing, 2000.

BARANY, I., VAN ROY, T. J., WOLSEY, L. A., **Strong formulations for multi-item capacitated lot sizing**, Management Science, 30(10):1255–61, 1984.

BELVAUX, G., WOLSEY, L.A., **Modelling practical lot-sizing problems as mixed-integer programs**, Management Science 47 (7), 993–1007, 2001.

BESSAOU, M., SIARRY, P., **A genetic algorithm with real-value coding to optimize multimodal continuous functions**, Struct Multidisc Optim 23, 63–74, Springer-Verlag 2001.

BITRAN, G. R., YANASSE, H. H., **Computational complexity of the capacited lot size problem**, *Management Science*, 28:1174–1186, 1982.

BNDES, **BNDES Setorial**, Banco Nacional de Desenvolvimento Econômico Social, Rio de Janeiro, n. 23, p. 101-150, Março de 2006.

BRAHIMI, N., DAUZERE-PERES, S., NAJID, N. M., NORDLI, A., **Single item lot sizing problems**, *European Journal of Operational Research*, 168, 1-16, 2006.

BRUGGEMANN, W., JAHNKE, H., **The discrete lot-sizing and scheduling problem: complexity and modification for batch availability**, *European Journal of Operational Research* 2000, 124(3), 511–28, 2000.

CHENG, T. C. E., DING, Q., LIN, B. M. T., **A concise survey of scheduling with time-dependent processing times**, *European Journal of Operational Research*. v152, 1-13, 2004.

DARWIN, C., **The Origin of Species**, Londres, Murria, 1872.

DREXL, A., HAASE, K., **Proportional lot sizing and scheduling**, *International Journal of Production Economics*, 40:73–87, 1995.

DREXL, A., KIMMS, A., **Lot sizing and scheduling – survey and extensions**, *European Journal of Operational Research* 99, 221–235, 1997.

FERREIRA, D., MORABITO, R., RANGEL, S., **Solution approaches for the soft drink integrated production lot sizing and scheduling problem**, Elsevier Ltd. European Journal of Operational Research n. 196, p. 697–706, 2009.

FLEISCHMANN, B., **The discrete lot sizing and scheduling problem**, European Journal of Operational Research 44, 337–348, 1990.

FLORIAN, M., LENSTRA, J. K., RINNOOY KAN, A. H. G., **Deterministic production planning: Algorithms and complexity**, Management Science 26 (7), 669–679, 1980.

GALLEGO, G., SHAW, D. X., **Complexity of the ELSP with general cyclic schedules**, IEEE Transactions, 29:109–13, 1997.

GEN, M., CHENG, R., **Genetic algorithms & engineering design**, John Wiley & Sons New York, 1997.

GOLDBERG, D. E., **Genetic Algorithms in search, optimization, and machine learning**, Addison Wesley, 1989.

GUPTA, D., MAGNUSSON, T., **The capacitated lot-sizing and scheduling problem with sequencedependent setup costs and setup times**, Computer & Operations Research, 32:727–747, 2005.

HAASE, K., KIMMS A., **Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities**, International Journal of Production Economics, (66): 159–169, 2000.

HOLLAND, J. H., **Adaptation in natural and artificial systems**, The University of Michigan Press, 1975.

JANS, R., DEGRAEVE, Z., **Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches**, European Journal of Operational Research 177, p. 1855–1875, 2007.

KARIMI, B., FATEMI, GHOMI, S. M. T., WILSON, J. M., **The capacitated lot sizing problem: a review of models and algorithms**, Elsevier Ltd., Omega n. 31, p. 365-378, 2003.

KUIK, R., SALOMON, M., VAN WASSENHOVE, L. N., **Batching decisions: Structure and models**, European Journal of Operational Research 75, 243–263, 1994.

LAFIS, **Cervejas, Refrigerantes e Água Mineral**, Relatório setorial, 10 de Junho de 2005.

LAND A. H., DOIG, A. G., **An Automatic Method for Solving Discrete Programming Problems**, Econometrica, Vol.28, pp. 497-520, 1960.

LEUNG, J. M. Y., MAGNANTI, T. L., VACHANI, R., **Facets and algorithms for capacitated lot sizing. Mathematical Programming**, 45:331–59, 1989.

LINDEN, R., **Algoritmos Genéticos: uma importante ferramenta da Inteligência Computacional**, Rio de Janeiro: Braspor, 2006.

MAES, J. O., VAN WASSENHOVE, L. N., **Multilevel capacitated lotsizing complexity and lp-based heuristics**, European Journal of Operational Research, 53:131–148, 1991.

MENDES, A. S., **O Framework NP-Opt e suas Aplicações a Problemas de Otimização**, Tese de Doutorado, Universidade Estadual de Campinas, 2003.

MEYR, H., **Simultaneous lot sizing and scheduling by combining local search with dual reoptimization**, European Journal of Operational Research 120, 311–326, 2000.

MEYR, H., **Simultaneous lot sizing and scheduling on parallel production lines**, European Journal of Operational Research 39, 277–292, 2002.

MICHALEWICZ, Z., FOGEL, D. B., **How to Solve It: Modern Heuristics**, 2nd ed., XVIII, 554 p. 174 illus., Hardcover, 2004.

ÖZDAMAR , L., BIRBIL., S. I., **Hybrid heuristic for the capacitated lot sizing and loading problem with setup times and overtime decisions**. European Journal of Operational Research, pages 525–547, 1998.

PINEDO, M., **Scheduling – Theory, Algorithms and Systems**. Prentice Hall, 1995.

TOLEDO, C. F. M., **Problema Conjunto de Dimensionamento de Lotes e Programação da Produção**, Tese de Doutorado, Universidade Estadual de Campinas, 2005.

TOLEDO, C. F. M., FRANÇA, P. M., MORABITO R., KIMMS, A., **Um modelo de otimização para o problema integrado de dimensionamento de lotes e programação da produção em fábricas de refrigerantes**, Pesquisa Operacional, 27 (1), 155-186, 2007.

TOLEDO, C. F. M., FRANÇA P. M., MORABITO R., KIMMS A., **Multi-Population Genetic Algorithm to Solve the Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem**, International Journal of Production Research, Volume 47, p. 3097-3119, 2009.

VON ZUBEN, F. J., **Computação Evolutiva: Uma Abordagem Pragmática**, DCA/FEEC/UNICAMP. Disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/tutorial/tutorialEC.pdf>, acessado em: 10 de Junho de 2009.

VOSE, M., **Modeling Simple Genetic Algorithms, Foundations of Genetic Algorithms -2-**, D. Whitley, ed., Morgan Kaufmann, pp 63-73, 1993.

WEINER, J., **The beak of the finch**, Vintage Books, New York, 1995.