

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE PESQUISA E PÓS - GRADUAÇÃO
PROGRAMA DE PÓS - GRADUAÇÃO EM INFORMÁTICA

GERALDO PEREIRA JUNIOR

**ALGORITMOS EVOLUTIVOS HÍBRIDOS APLICADOS NO
SEQUENCIAMENTO DE PRODUÇÃO EM UMA INDÚSTRIA DE
ALIMENTOS**

DISSERTAÇÃO - MESTRADO

CORNÉLIO PROCÓPIO
2018

GERALDO PEREIRA JUNIOR

**ALGORITMOS EVOLUTIVOS HÍBRIDOS APLICADOS NO
SEQUENCIAMENTO DE PRODUÇÃO EM UMA INDÚSTRIA DE
ALIMENTOS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR como requisito parcial para a obtenção do título de “Mestre Profissional em Informática”.

Orientador: Prof. Dr. Danilo Sipoli Sanches

CORNÉLIO PROCÓPIO
2018



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Programa de Pós-Graduação em Informática



Título da Dissertação Nº XX:

ALGORITMOS EVOLUTIVOS HÍBRIDOS APLICADOS NO SEQUENCIAMENTO DE PRODUÇÃO EM UMA INDÚSTRIA DE ALIMENTOS.

por

GERALDO PEREIRA JUNIOR

Orientador: **Prof. Dr. Danilo Sipoli Sanches**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM INFORMÁTICA – Área de Concentração: Computação Aplicada, pelo Programa de Pós-Graduação em Informática – PPGI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 16h do dia 24 de maio de 2018. O trabalho foi _____ pela Banca Examinadora, composta pelos professores:

Prof. Dr. Danilo Sipoli Sanches
(Presidente – UTFPR-CP)

Prof. Dr. Orides Morandin Junior
(DC-UFSCAR)
Participação à distância via Skype

Prof. Dra. Glaucia Maria Bressa
(Presidente – UTFPR-CP)

Visto da coordenação:

Danilo Sipoli Sanches
Coordenador do Programa de Pós-Graduação em Informática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedico este trabalho a Deus, que pela sua misericórdia, enviou o seu Filho, Jesus Cristo para ter morte atroz, no duro madeiro da cruz, e assim salvar a minha alma e me oferecer a vida eterna, se na minha jornada for um filho firme e fiel.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Danilo Sipoli Sanches, pela sabedoria, paciência e preocupação com que me guiou nesta trajetória.

Agradeço ao Prof. Dr. Orides Morandin Junior pela excelente contribuição na estruturação e contribuições essenciais para a qualidade desta dissertação.

Agradeço à Prof^a. Dr^a. Glaucia Bressan pelas importantes recomendações durante a avaliação do projeto de dissertação de mestrado.

Agradeço ao Prof. Dr. Carlos Nascimento Silla Junior, por todo apoio oferecido durante suas disciplinas, a quem também devo boa parte do bom andamento desta dissertação.

Agradeço ao Prof. Dr. Elvis Fusco, pelo incentivo e colaboração neste regresso às salas de aula.

Aos meus colegas de sala, e aos professores das disciplinas que cursei.

Agradeço ao Prof. Dr. Luiz Artur Ledur Brito, o meu maior incentivador para voltar às salas de aula e fonte de inspiração para temas, desde os mais simples aos mais complexos.

A Secretaria do Curso, pela cooperação.

Gostaria de deixar registrado também, o meu reconhecimento e amor à minha mãe, que fez o possível e o impossível para criar 5 filhos com dignidade e dá-los a chance de serem seres humanos em toda a sua essência.

Agradeço aos meus irmãos e irmã, os quais sinto orgulho e amor infinito, e ao meu pai que não mais o vejo, mas sei que está ao meu lado sofrendo e sorrindo junto comigo durante este desafio.

Em especial, àqueles que acalmam o meu coração, meus filhos Nicollas, Lincoln, Alice e minha doce Silmara, aos quais não há palavras existentes para expressar o que sinto por eles.

Não é o mais forte que sobrevive, nem o mais inteligente, mas o que melhor se adapta as mudanças. Charles Robert Darwin, 1859.

RESUMO

PEREIRA JR, GERALDO. **Algoritmos Evolutivos Híbridos Aplicados no Sequenciamento de Produção em uma Indústria de Alimentos**. 2018. 69 f. Dissertação – Programa de Pós-Graduação em Informática. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Este trabalho descreve um estudo comparativo entre algumas abordagens de sequenciamento de produção para um caso real em uma indústria de alimentos da cidade de Marília, SP. Para este problema, caracterizado como *Job-Shop Scheduling Problem* (JSSP), foram utilizados dois tipos de algoritmos evolutivos: Algoritmos Genéticos e Evolução Diferencial. Com o objetivo de aumentar a eficiência no processo de busca, estes algoritmos foram adaptados com técnicas de busca local. Além disso, é proposta uma heurística de balanceamento de carga a fim de alcançar regiões promissoras no espaço de busca. Para validação das abordagens propostas foi utilizado um cenário real de uma indústria de alimentos.

Palavras-chave: SEQUENCIAMENTO DE PRODUÇÃO, SEQUENCIAMENTO FLEXÍVEL DE PRODUÇÃO, NP-COMPLEXO, ALGORÍTMO GENÉTICO, EVOLUÇÃO DIFERENCIAL, RECOZIMENTO SIMULADO, BALANCEAMENTO DE CARGA, INDÚSTRIA DE ALIMENTOS, VNS, GRASP.

ABSTRACT

PEREIRA JR, GERALDO. **Hybrid Evolutionary Algorithms Applied in Production Sequencing in a Food Industry.** 2018. 69 f. Dissertação – Programa de pós-graduação em Informática. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

This paper describes a comparative study of some production-scheduling scenarios for a real case in a food company located in Marília, SP. For this problem, well known Job-Shop Scheduling Problem (JSSP), it was used two types of evolutionary algorithms: Genetic Algorithm and Differential Evolution. In order to increase the efficiency during the search process, these two algorithms have been adapted with local search techniques. Furthermore, it was proposed a heuristic based on load balancing in order to achieve promising regions in the search space. Finally, the proposed approaches have been validated for a real scenario from a food company.

Keywords: JOB-SHOP, FLEXIBLE JOB-SHOP, NP-HARD, GENETIC ALGORITHM, DIFFERENTIAL EVOLUTION, SIMULATED ANNEALING, LOAD BALANCE, FOOD INDUSTRY, VNS, GRASP.

LISTA DE FIGURAS

Figura 1 – Fluxograma do Genetic Algorithm.....	19
Figura 2 – Fluxo <i>Genetic Algorithm</i>	20
Figura 3 - Processo de geração do vetor modificado da função bidimensional	23
Figura 4 - Funcionamento da Evolução Diferencial.....	24
Figura 5 - Exemplo DE com JOB-SHOP Fictício	24
Figura 6 - Algoritmo <i>Simulated Annealing</i>	26
Figura 7 - Exemplo movimento 2opt.....	27
Figura 8 - Execução do balanceamento de carga entre máquinas	27
Figura 9 - Algoritmo GRASP.....	28
Figura 10 - Algoritmo VNS	29
Figura 11 – <i>JOB-SHOP PROBLEM 3 X 5</i>	31
Figura 12 - Representação do FJSP	32
Figura 13 - <i>Flexible Job-Shop Problem</i>	32
Figura 14 - Fluxograma Evolução Diferencial com Mutação 2-opt.....	35
Figura 15 - Fluxograma Evolução Diferencial com Mutação SA + LB	36
Figura 16 - Fluxograma Algoritmo Genético com Mutação 2-opt.....	36
Figura 17 - Fluxograma Algoritmo Genético com Mutação SA + LB.....	36
Figura 18 - Fluxograma GA ajustado para novas heurísticas	37
Figura 19 - Aplicação R	47
Figura 20 - Aplicação JAVA	48
Figura 21 - Resultados por método	50
Figura 22 - <i>Boxplot</i> . Resultados Obtidos – Amendoins.....	51
Figura 23 - <i>Boxplot</i> - Empacotamento de Gomas.....	54
Figura 24 - <i>Boxplot</i> - Semielaborados de Gomas.....	56

LISTA DE TABELAS

Tabela 1 – Principais Trabalhos Relacionados	34
Tabela 2 - Possibilidades de produção: Máquina versus Produto - Amendoins	39
Tabela 3 - Possibilidades de empacotamento: Máquina versus Produto - Gomas.....	40
Tabela 4 - Produção de Gomas Semielaboradas.....	41
Tabela 5 - Composição do cromossomo.....	42
Tabela 6 - Parâmetros de Execução – Empacotamento de Amendoins	49
Tabela 7 - Resumo dos resultados obtidos – Empacotamento de Amendoins	50
Tabela 8 - Parâmetro Empacotamento de Gomas	52
Tabela 9 - Resultados Empacotamento de Gomas.....	53
Tabela 10 –Parâmetros de execução – Semielaborado Gomas	55
Tabela 11 - Resultados obtidos – Semielaborado Gomas	55

LISTA DE SIGLAS

DE	DIFFERENTIAL EVOLUTION
FJSP	FLEXIBLE JOB-SHOP PROBLEM
GA	GENETIC ALGORITHM
GRASP	GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES
JSP	JOB-SHOP PROBLEM
JSSP	JOB-SHOP SCHEDULING PROBLEM
LB	LOAD BALANCE
MS	MILESEGUNDOS
NP	NONDETERMINISTIC POLYNOMIAL TIME
OPT	OPTIMIZATION
OTIF	ON TIME IN FULL
PCP	PLANEJAMENTO E CONTROLE DA PRODUÇÃO
PDM	PROJETO DE DISSERTAÇÃO DE MESTRADO
SA	SIMULATED ANNEALING
SBPO	SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
SKU	STOCK KEY UNIT
VNS	VARIABLE NEIGHBORHOOD SEARCH

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	16
1.2 Organização do Trabalho	17
2 ALGORITMOS EVOLUTIVOS	18
2.1. Genetic Algorithm	18
2.2. Differential Evolution	21
2.3. Algoritmos de Busca Local	25
3 O PROBLEMA JOB-SHOP PROBLEM E FLEXIBLE JOB-SHOP PROBLEM	30
3.1. Trabalhos relacionados	33
4 METODOLOGIA	35
4.1 Abordagens propostas	35
4.2 Definição do Problema	38
4.3. Desafios desta indústria	44
5 RESULTADOS	49
5.1 Empacotamento de Amendoins	49
5.2 Empacotamento de Gomas	52
5.3 Produção de Semielaborado de Gomas	54
6 CONCLUSÕES E RECOMENDAÇÕES PARA TRABALHOS FUTUROS	58
6.1 Conclusões	58
6.2 Recomendações para trabalhos futuros	58
REFERÊNCIAS	60

1 INTRODUÇÃO

Nas indústrias, a atividade de programação é uma das mais complexas tarefas no gerenciamento de produção. Os programadores precisam lidar com diversos tipos de recursos simultaneamente. As máquinas terão diferentes capacidades e o pessoal terá diferentes habilidades. De maneira mais importante, o número de programações possíveis cresce rapidamente à medida que o número de atividades e processos aumenta. Ou seja, para n tarefas há $n!$ (n fatorial) maneiras diferentes de programação dos trabalhos em um processo simples. Considerando mais que uma máquina ($M > 1$), o número de programações possíveis passa para $(n!)^M$. (Pezzella et al. 2008).

Embora pesquisadores voltados à pesquisa operacional e à inteligência artificial estejam investigando o sequenciamento de produção por décadas, ainda há uma lacuna entre os problemas reais e aqueles de caráter acadêmicos. (Fayad & Petrovic, 2005). Em função da dificuldade de obtenção de resultados para problemas de um ambiente real complexo, aplicamos nesta dissertação a combinação de técnicas de inteligência artificial, visando tratar o problema real *job-shop* em uma indústria de alimentos.

Apesar das metaheurísticas DE - *Differential Evolution* (Evolução Diferencial) e GA - *Genetic Algorithm* (Algoritmo Genético) serem consideradas boas para o problema *Job-Shop* (Sequenciamento de Produção), torna-se relevante a utilização dos algoritmos híbridos, pois alguns estudos mostram a possibilidade de melhorar os resultados obtidos por estas metaheurísticas, através do uso de algumas técnicas de busca local. Čičková & Števo (2010), Yuan & Xu (2013).

O JSSP - *Job-Shop Scheduling Problem* (Problema de Sequenciamento de Produção) não é só mais um problema da classe *NP-hard*, mas um dos piores problemas desta classificação. Uma indicação disso é dada pelo fato de que o problema 10x10 formulado por Muth e Thompson (1963) permaneceu por mais de 40 anos sem solução. (Russel, et al., 1995)

Em função da complexidade e do elevado número de soluções possíveis para o problema de sequenciamento, é praticamente impossível modelar todas as possibilidades envolvidas no processo fazendo uso dos modelos matemáticos. Somado a isso, ao utilizar modelos matemáticos, o tempo de resposta cresce de forma

considerável. Sendo assim, encontramos grande dificuldade na obtenção de uma solução ótima em tempos satisfatórios. Algoritmos matemáticos para otimização são computacionalmente viáveis quando tratamos problemas simples, com baixa ou média complexidade.

Para problemas do tipo NP-*hard*, é comum abrir mão da busca de uma solução ótima em função do custo de processamento ou até mesmo da inviabilidade do tempo, por uma solução “quase” ótima, ou boa solução, com um tempo de processamento aceitável. (Morton & Pentico, 1993).

Diante da complexidade dos ambientes reais e conseqüente dificuldade na obtenção de ótimos resultados na aplicação de metaheurísticas, esta dissertação propõe a investigação de metaheurísticas modificadas, substituindo o processo de geração aleatória dos indivíduos e a mutação por algoritmos de busca local, explorando o espaço de busca sem a perda de diversidade e possibilitando uma melhor avaliação no direcionamento para regiões mais promissoras.

Durante a pesquisa, foi identificado que o ambiente estudado, uma indústria de alimentos da cidade de Marília, SP, apresentava boa parte das características de um problema da classe NP-*hard*, ou seja, haviam diversos cenários, incluindo diferentes famílias de produtos, unidades fabris, processos produtivos, máquinas de produção e uma grande diversidade de ordens de produção. Somado a esta complexidade, não foi observado a aplicação de técnicas de otimização no sequenciamento destas produções, apenas o uso limitado de planilhas eletrônicas e a experiência dos colaboradores da área de PCP (Planejamento e Controle da Produção).

Portanto, com o intuito de avaliarmos a aderência das heurísticas e das técnicas de busca local neste ambiente, primeiramente elegemos uma pequena amostra do grande problema encontrado, ou seja, delimitamos ao setor de empacotamento e também restringimos o caso de teste à família de produtos de amendoins. Nesta primeira etapa da pesquisa, o desafio inicial foi criar um conjunto de dados que representasse esta respectiva produção.

O processo de produção de amendoins, está configurado para trabalhar com três linhas de produção, denominadas: Salgado, Crocante e Doce. A operação pode atuar com até 3 turnos, no entanto, há uma variação de acordo com a demanda do mercado. O setor de empacotamento dispõe de 19 máquinas de produção que podem

ser utilizadas de forma simultânea, uma vez que há disponibilidade de mão de obra para tanto. Podem ser embalados nestas máquinas, nove gramaturas diferentes, desde 70g até 5kg.

Neste contexto, o primeiro desafio foi criar um *dataset* (conjunto de dados) e durante algumas semanas foi necessário efetuar diversas entrevistas com os colaboradores do PCP, efetuar visitas à fábrica onde os amendoins são empacotados e entrevistar coordenadores e o gerente de produção da unidade. Diante dos levantamentos efetuados, foi possível construir uma matriz com os tempos de produção para a relação máquina versus ordem de produção, que a partir daí é formado nosso primeiro *dataset*.

A partir deste *dataset*, foi possível o desenvolvimento para a aplicação do método de pesquisa baseado em algoritmos híbridos, que foram construídos até a apresentação do Projeto de Dissertação de Mestrado (PDM) e também apresentados no XLIX Simpósio Brasileiro de Pesquisa Operacional. Para este primeiro caso de teste, utilizamos o *GA* tradicional e o *DE*. Estas metaheurísticas foram adaptadas durante o processo de mutação, utilizando as heurísticas *SA - Simulated Annealing* (Recozimento Simulado), *LB - Load Balance* (Balanceamento de Carga) e a inversão entre genes, conhecida como *K-opt*, como técnicas de busca local.

Os resultados deste primeiro caso de teste, além de confirmar a boa aderência para o ambiente pesquisado, também demonstrou excelentes resultados, tanto para a heurística *GA*, quanto para *DE*. Este último apresenta resultados positivos. No entanto, o maior destaque ficou por conta das técnicas de busca local, apresentando bons resultados, o que nos levou à decisão de aprofundá-las na sequência do trabalho, e nos demais casos de testes que foram criados.

Durante os levantamentos efetuados na empresa, consideramos também estudar com mais profundidade os desafios apresentados para a sua principal linha de produtos, as balas de goma, que são produzidas em uma unidade fabril, diferente da estudada no caso de teste anterior, planejando assim a execução de mais dois casos de testes, a produção de semielaborados de gomas e o seu respectivo empacotamento.

É importante ressaltar que nas linhas são produzidos quatro tipos de produtos: goma tubo, gomas formatos (anel, urso, sino, minhoca, coração, pudim, etc.), goma

confeitada, popularmente conhecida como *Jelly Beans* e *Jubes*. Já no setor de empacotamento, estão disponíveis 11 máquinas de empacotamento para absorver 18 diferentes gramaturas.

Após entrevistas com os gestores da área de planejamento e também da área produtiva, foram identificados grandes desafios, sendo um destes denominado “segundo semestre”. Este problema encontra-se em função da produção ser um processo linear durante todo o ano, no entanto, na demanda não há esta linearidade, ou seja, para o primeiro semestre as vendas são menores se comparadas ao segundo semestre que por sua vez é influenciado positivamente pelas exportações.

A partir deste cenário e diante da disponibilidade das informações, seguimos com a pesquisa no intuito de aprofundar e encontrar uma função *fitness* dentro de uma técnica de otimização no sequenciamento da programação de produção de semielaborados e do empacotamento. Este desafio consiste em disponibilizar mais produtos para atender a demanda sem gerar rupturas e ainda não elevar os níveis de estoque, melhorando assim os resultados da empresa.

Diante deste desafio, e baseado nos resultados obtidos no primeiro caso de teste, foi construído um novo algoritmo, modificando o GA Tradicional, durante a geração de novos indivíduos e também no processo de mutação, as heurísticas VNS – *Variable Neighborhood Search* (Pesquisa variável na vizinhança) e GRASP - *Greedy Randomized Adaptive Search Procedure* (Procedimento Ágil de Pesquisa Adaptativa Aleatória) foram adaptadas ao problema.

Para estes dois novos casos de testes também foram necessárias novas entrevistas e levantamentos de dados, bem como visitas *in loco* nos setores de produção e empacotamento de gomas. Este levantamento possibilitou a criação de dois novos *datasets* (produção de semielaborados de gomas e empacotamento de gomas), que foram utilizados para o segundo e terceiro casos de teste.

Os resultados obtidos para estes dois novos casos de testes também foram bastante promissores e totalmente aderentes ao ambiente, possibilitando para a área de planejamento de produção uma visão conjunta entre as fases de produção e empacotamento, otimizando assim os recursos disponíveis.

1.1 Objetivos

O objetivo geral desta dissertação é propor a automatização do processo de sequenciamento de produção em uma indústria de guloseimas doces e salgadas, utilizando um método de pesquisa baseado em algoritmos híbridos.

Os objetivos específicos são:

- Melhorar o processo de sequenciamento de produção por meio de revisão dos métodos utilizados atualmente;
- Comparar os resultados obtidos dos diferentes algoritmos com os sequenciamentos realizados pelos programadores de produção;
- Formalizar o processo de sequenciamento de produção através de políticas e procedimentos;
- Reduzir as ociosidades de máquinas e mão de obra, proporcionando maior produção;
- Melhorar os custos de produção;
- Oferecer informações relevantes para projetos de melhoria contínua em fábrica voltados a redução de ociosidade nas máquinas de empacotamento e no processo de produção dos seus principais produtos.
- Otimizar o processo de produção através de um sequenciamento inteligente, proporcionando os produtos adequados para as linhas de balas de goma tanto no primeiro, quanto no segundo semestre, evitando assim aumentar os custos de produção e reduzir rupturas na entrega aos clientes.

1.2 Organização do Trabalho

No Capítulo 2 desta dissertação, descrevemos os algoritmos utilizados no trabalho e suas respectivas adaptações. O JSP e o FJSP são descritos no Capítulo 3, bem como os trabalhos relacionados. No Capítulo 4 são descritas as abordagens propostas, bem como os fluxogramas do funcionamento de cada algoritmo, a importância e relevância do problema abordado, o *Job-Shop* para um caso real composto por um espaço de busca contendo uma grande diversidade de máquinas e ordens de produção (*Jobs*), e a metodologia aplicada nesta pesquisa e os respectivos desafios desta indústria. Detalhamos os resultados e as conclusões obtidas com as metaheurísticas e suas variações no Capítulo 5, bem como os resultados preliminares e finais da pesquisa. O respectivo cronograma de trabalho é apresentado no Capítulo 6.

2 ALGORITMOS EVOLUTIVOS

Neste capítulo descrevemos as principais características dos algoritmos evolutivos utilizados neste trabalho: *Genetic Algorithm* e *Differential Evolution*. Também é feita uma introdução das respectivas heurísticas de busca local, ou seja, *Simulated Annealing*, *K-opt*, *Load Balance*, *Greedy Randomized Adaptive Search Procedures* e *Variable Neighborhood Search*.

Introduzidos por JOHN HOLLAND na década de 70 e popularizados por DAVID GOLDBERG no final da década de 80, GA é uma técnica de busca e otimização inspirada na teoria da evolução de Charles Darwin e fazem parte de uma área de pesquisa denominada Computação Evolucionária. Desde então diversos estudos foram feitos buscando a aplicação de metaheurísticas que aplicadas revelaram oportunidades em melhorar resultados para problemas dos mais simples aos mais complexos. Pesquisadores denominam estes resultados como *makespan*.

No intuito de melhorar o *makespan*, são testadas no GA as heurísticas que obtiveram bons resultados para o problema JSP, sejam em casos reais ou em base de dados utilizadas amplamente pelos pesquisadores. Utilizaremos artigos e publicações clássicas da literatura, sendo fora de escopo deste trabalho a criação e proposta de novas metaheurísticas.

Os resultados de *makespan* são registrados e tabulados no sentido de compará-los aos resultados históricos das programações efetuadas manualmente ou com o auxílio de planilhas eletrônicas pelos programadores de produção. Estes resultados tabulados serão apresentados ao comitê executivo da empresa para avaliação, validação e conseqüente incorporação do novo processo para a empresa.

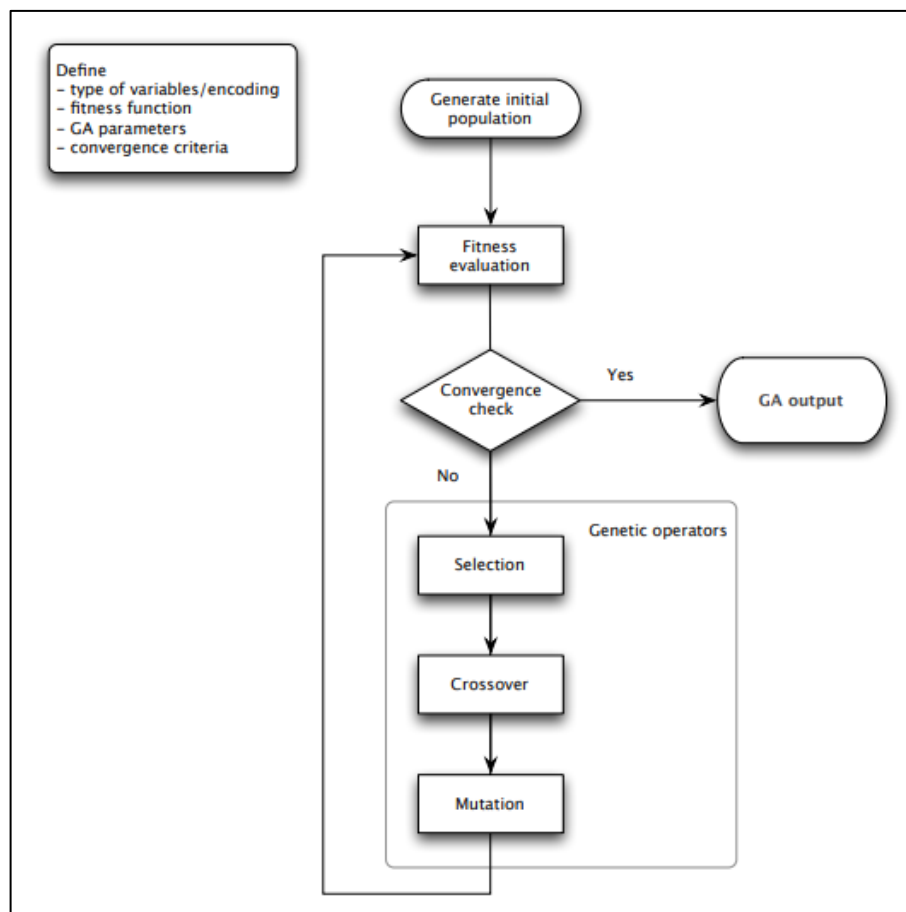
2.1. *Genetic Algorithm*

Métodos genéticos são técnicas que otimizam soluções. Estes métodos são baseados na evolução natural, onde os indivíduos que conseguirem melhor adaptação ao ambiente irão sobreviver e gerar descendentes, e os demais irão desaparecer. Estes métodos utilizam uma estratégia de criação e testes de gerações, permitindo uma pesquisa paralela no espaço de soluções possíveis. Tais métodos são denominados estocásticos e não dão a garantia de solução ótima, porém na prática, eles são

amplamente utilizados, gerando bons resultados em diversas aplicações. (Forrest, 1993; Linden, 2012; Mitchell, 1997).

De acordo com Lawrynowicz (2011), o GA deve considerar seis etapas, sendo: 1- A definição da estrutura dos indivíduos que irá representar nosso problema; 2 - Um mecanismo para criação da população inicial, que pode ser obtida através de um método probabilístico, sendo este mais comum, mas que para alguns casos métodos determinísticos também podem ser utilizados; 3 - Uma função para avaliação do *fitness*, ou seja, determinar o valor de cada indivíduo; 4 - O processo de seleção dos pais, que irão gerar novos indivíduos para as populações seguintes; 5 - Heurísticas capazes de efetuar a mutação e o cruzamento para pesquisar melhores indivíduos; e 6 - A calibração dos parâmetros, tais como taxas de mutação, número de interações, tamanho da população, etc.

Figura 1 – Fluxograma do Genetic Algorithm



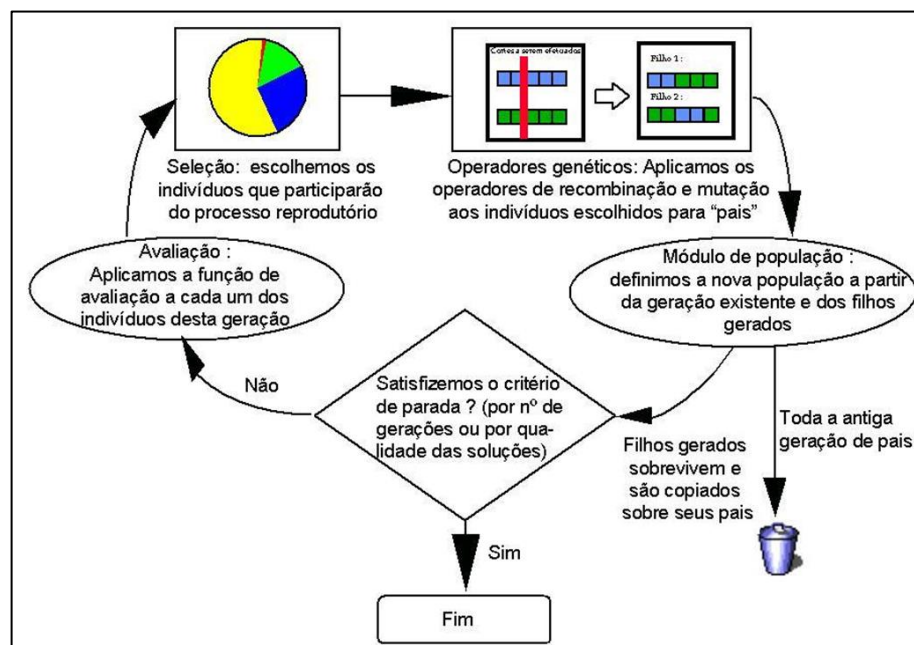
A Figura 1 traz o funcionamento geral do GA que funciona de acordo com os seguintes passos: Determinar uma população inicial de indivíduos; calcular a aptidão de cada indivíduo, valorizando-os; verificar se houve ou não convergência, e em caso

afirmativo, termina o processo e retorna o indivíduo com a melhor aptidão; caso contrário seguir para as próximas etapas. Selecionar os cromossomos que produzirão descendentes para a próxima geração, em função dos seus desempenhos calculados durante a seleção; gerar descendentes através da aplicação das operações de cruzamento e mutação sobre os indivíduos selecionados durante a reprodução.

De acordo com Scrucca, (2013), a criação de diversidade populacional explorando o espaço de busca se dá através dos operadores de mutação e cruzamento. O cruzamento forma novos descendentes a partir de dois cromossomos pais, combinando parte da informação genética de cada um. Já a mutação, altera de forma aleatória os valores dos genes dos pais.

A seguir, na Figura 2, o fluxo básico para melhor ilustração do funcionamento do GA

Figura 2 – Fluxo Genetic Algorithm



De acordo com Pezzella, et.al, (2008), no processo de seleção, a cada interação os melhores cromossomos podem ser escolhidos para reprodução através de um entre três diferentes métodos, sendo eles: torneio binários, torneio *n-size* e *ranking linear*.

- Torneio binário: Dois indivíduos são escolhidos aleatoriamente na população e o melhor deles é selecionado para reprodução.

- Torneio *n-Size*: O indivíduo para reprodução é escolhido entre um número aleatório de indivíduos.
- Ranking Linear: Os indivíduos são classificados de acordo com a sua aptidão (fitness), e uma nota é atribuída a cada indivíduo, onde o melhor indivíduo recebe a nota mais alta e o pior a nota mais baixa. Isto faz com que a probabilidade daquele indivíduo melhor classificado tenha maior probabilidade de ser escolhido.

Mitchell & Forrest (1994) e Yamada & Nakano (1997), destacam a importância de se aplicar os processos de mutação e cruzamento, onde operadores genéticos são utilizados para produzir descendentes. Operadores de cruzamento são aplicados em pares de cromossomos, enquanto operadores de mutação são aplicados nos indivíduos. Os principais operadores de cruzamentos são: Um ponto, onde um ponto de cruzamento é escolhido e a partir deste ponto as informações genéticas dos pais serão trocadas. As informações anteriores a este ponto em um dos pais são ligadas às informações posteriores a este ponto no outro pai. Multipontos, onde é uma generalização desta ideia de troca de material genético através de pontos, onde muitos pontos de cruzamento podem ser utilizados. Entretanto, eles são uniformes, não utilizam pontos de cruzamento e gera através de um parâmetro global a probabilidade de cada variável. Já para o processo de mutação, utilizam os processos uniformes, não uniformes, aleatórios, entre outros, como exemplo a inversão de bits onde um gene é alterado de 0 para 1 ou vice-versa, a mudança de ordem, dois números são escolhidos e trocados entre si e até mesmo a mudança de operador que modifica nós escolhidos.

2.2. Differential Evolution

Bem como o GA, o DE pertence à classe dos algoritmos evolucionários. Todavia existem diferenças fundamentais entre os algoritmos genéticos e a evolução diferencial. Na evolução diferencial um indivíduo é criado a partir de quatro pais e sofre dupla mutação, enquanto no algoritmo genético tradicional um indivíduo é criado a partir de dois pais e sofre mutação simples. (Čičková & Števo, 2010).

O DE é uma heurística para minimização de funções não-lineares e não-diferenciáveis no espaço contínuo. Os testes com o DE em diversas instâncias comprovaram que esta heurística obteve resultados superiores comparados à outras heurísticas.

Segundo Storn & Price (1997), o funcionamento do DE também exige a geração de uma população inicial de forma aleatória, que é composta de N_p indivíduos, denominados vetores, abrangendo todo o espaço de busca. Em geral, esta população é gerada por uma distribuição de probabilidade uniforme, caso não haja nenhum conhecimento sobre o problema. A população seguirá a evolução natural, mantendo o mesmo número de indivíduos para todas as gerações.

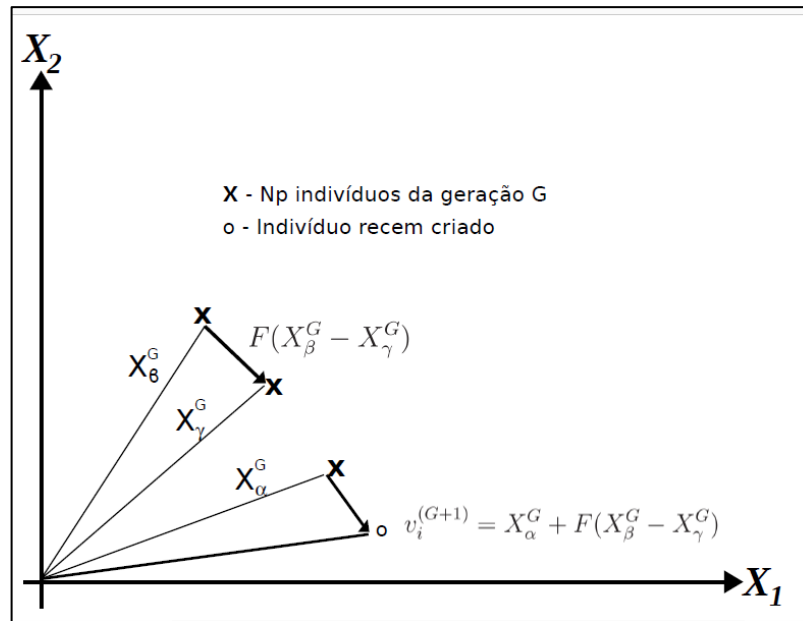
Após a geração da população inicial, são necessários mais dois parâmetros iniciais, sendo: o fator escalar F , da equação de mutação diferencial, um número real positivo pertencente ao intervalo $[0,2]$, que irá controlar a amplitude da diferença vetorial. De forma indireta essa amplitude irá controlar a taxa em que a população corrente evolui e a sua distribuição no espaço de busca; o segundo parâmetro é o escalar C_r , que é a taxa de cruzamento de um indivíduo da população durante o processo de cruzamento entre indivíduos, também conhecido como *crossover*.

A operação de mutação para o DE acontece através da geração de novos indivíduos, denominados vetores modificados, que são obtidos através da soma da diferença vetorial ponderada pelo fator escalar F , utilizando dois indivíduos aleatórios a um terceiro indivíduo. Nesta etapa, para cada indivíduo corrente X_α^G da população, ou seja, o vetor alvo, é criado um vetor modificado v_i^{G+1} , conforme equação (1):

$$v_i^{G+1} = X_\alpha^G + F(X_\beta^G - X_\gamma^G) \quad (1)$$

Para obter o vetor modificado v_i^{G+1} , devemos considerar os vetores X_α^G , X_β^G , X_γ^G , distintos entre si, selecionados em uma população com N_p indivíduos de forma aleatória. Os índices aleatórios α , β e $\gamma \in \{1, \dots, N_p\}$ e são inteiros e distintos entre si. Para o par de vetores X_β^G , X_γ^G da G -ésima geração, obtemos o vetor diferença ($X_\beta^G - X_\gamma^G$). Após isso, multiplicamos esta diferença pelo escalar F , sendo este resultado a diferença ponderada, que por sua vez será usada para perturbar o vetor X_α^G . Na Figura 3 é possível observar um exemplo bidimensional composto pelos diferentes vetores que geram o vetor modificado v_i^{G+1} .

Figura 3 - Processo de geração do vetor modificado da função bidimensional



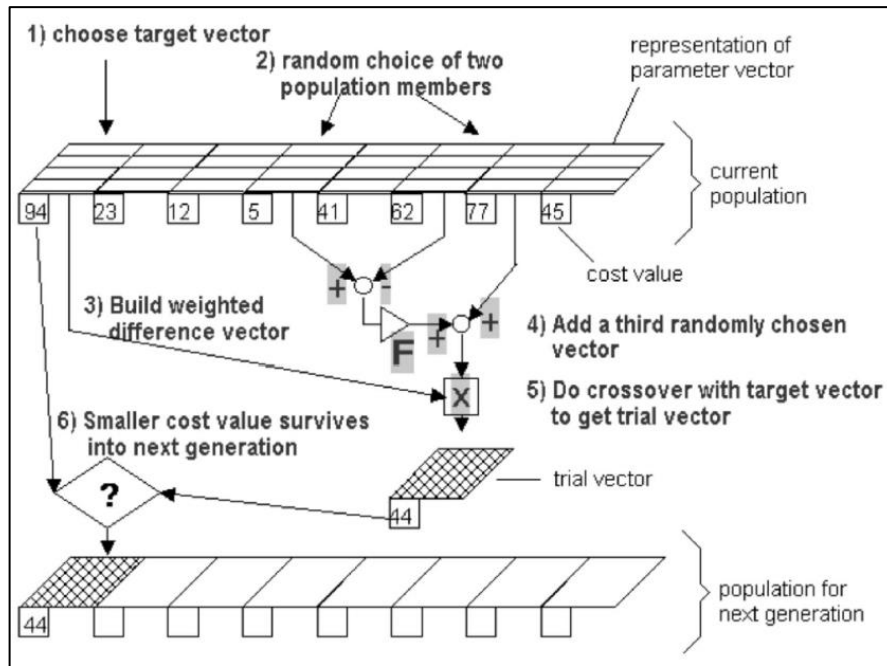
Após o processo de mutação, é executado o cruzamento, onde um indivíduo é escolhido aleatoriamente para que os seus componentes sejam misturados com os componentes do vetor modificado, assim gerando o vetor experimental U^{G+1} . Portanto, utilizando o vetor alvo e o vetor modificado, os componentes do vetor experimental U^{G+1} são selecionados pela comparação de acordo com a equação (2):

$$U^{G+1} = \begin{cases} v(i)^{G+1}, & \text{se } rand_i \leq C_r \\ X_\alpha(i)^G, & \text{se } rand_i \geq C_r, i = 1, \dots, n \end{cases} \quad (2)$$

Onde, $v(i)^{G+1}$ é o i -ésimo componente do vetor modificado $v(i)^{G+1}$, e $X_\alpha(i)^G$ o i -ésimo componente do vetor alvo X_α^G , e $rand_i$ um número gerado aleatoriamente no intervalo $[0,1]$, e C_r que representa a taxa de probabilidade do vetor experimental herdar os valores das variáveis do vetor modificado.

No processo de seleção, caso o custo do vetor experimental for menor que o custo do vetor alvo, o vetor experimental passa a ser o vetor alvo para ser utilizado nas próximas gerações. O procedimento poderá ser encerrado através da convergência, ou mesmo por um número máximo de avaliações, o que pode ser observado na Figura 4:

Figura 4 - Funcionamento da Evolução Diferencial



O DE, comparado com outros algoritmos evolucionários, tem obtido melhores resultados, apesar da sua simplicidade e da facilidade de implementação. Este bom desempenho está relacionado com a sua habilidade na diversificação dos indivíduos e na execução de buscas locais com qualidade. Em função disto, o DE tem despertado na comunidade acadêmica e industrial um grande interesse nas últimas décadas.

Para efeito de aprofundamento no entendimento da heurística DE, foi desenvolvido um exemplo prático, porém fictício, ligado ao problema *Job-Shop*, que pode ser observado na Figura 5:

Figura 5 - Exemplo DE com JOB-SHOP Fictício

Cenário JOB-SHOP fictício – 5 x 3 : 5 Jobs e 3 máquinas

PARÂMETROS

F (Taxa de mutação) [0-2] = 0,8

Probabilidade de Crossover = 90% (0,9)

Limite inferior = 1; Limite Superior = 3

Velocidades: MAQ1=5 MAQ2=10 MAQ3=12

Indivíduo alvo (a ser substituído) = 5; selecionado aleatoriamente

Vetor principal: 1; Segundo: 2; Terceiro: 3

Fitness*: 1=20 2=24 3=24 4=24 5=36 6=15

*Considera caminho crítico

1 - POPULAÇÃO INICIAL ALEATÓRIA / CÁLCULO FITNESS

INDIVÍDUOS

1	3	2	1	3	1	.
2	1	3	2	3	2	.
3	2	1	1	2	3	.
1	3	2	3	2	3	.
2	1	3	1	1	3	.
1	3	2	1	3	1	.

2 - MUTAÇÃO

Vetor Novo* = Vetor Principal + Taxa Mutação x (Segundo Vetor – Terceiro Vetor)

2	(1,8)	1	3	2			
1	(0,4)	2	1	3			
3	(3,8)	3	+ 0,8 x (2	-	1)
2	(1,8)	1	3	2			
1	(0,4)	2	1	3			

*Nas posições 1,2 e 4 do Vetor novo, foram utilizados os limites superior e inferior. Fitness = 20

3 - CRUZAMENTO

2	RAND < 0,9	2	Fitness = 24
1	RAND > 0,9	3	Fitness = 36
3	RAND < 0,9	3	
2	RAND < 0,9	2	
1	RAND < 0,9	1	

O processo inicia com a definição dos parâmetros. O primeiro parâmetro é o escalar F , utilizado como fator no processo de mutação. Também se faz necessário inicializar o parâmetro de cruzamento, e os limites inferiores e superiores, a fim de evitar que máquinas inexistentes se tornem genes dentro do cromossomo e invalide o indivíduo. Ainda nesta etapa são definidas as respectivas velocidades de cada máquina, que serão utilizadas na sequência para composição do *fitness*. Em seguida é feita a geração da população inicial aleatória, e o cálculo do respectivo *fitness* para cada indivíduo. Selecionamos então de forma aleatória o indivíduo alvo e os genitores (principal, segundo e terceiro). De posse dos parâmetros, indivíduos e respectivos *fitness*, iniciamos os processos de mutação e crossover para encontrar um novo indivíduo. Caso este novo indivíduo gerado tenha um melhor *fitness* que o indivíduo alvo, este novo indivíduo passar a ser o indivíduo alvo. E é isto que acontece no exemplo da figura 5, pois o indivíduo alvo possui um *fitness* de 36, contra um *fitness* do novo indivíduo de 24. A estratégia aqui utilizada foi: DE/rand/1/bin, ou seja, DE (*differential evolution* / genitor principal gerado de forma aleatória / 1 vetor diferencial / cruzamento binomial).

2.3. Algoritmos de Busca Local

Descrevemos neste tópico, os algoritmos de busca local, utilizados neste trabalho para substituir o processo de mutação aleatória e também durante a geração da população inicial, originalmente utilizado pelas metaheurísticas.

O primeiro algoritmo é o SA. De acordo com Jeong, et.al, (2016), *Simulated Annealing* é um método inspirado no processo de tratamento de metais da indústria metalúrgica, no qual um sólido é aquecido lentamente e esfriado sob uma condição natural. Também é uma técnica probabilística que tem sido aplicado em grandes áreas, como a engenharia de controle, o aprendizado de máquina e o processamento de imagem. Este tipo de abordagem é reconhecido como uma abordagem prática para a solução de muitos problemas complexos, incluindo casos reais de diferentes naturezas. O SA também fornece um conjunto de soluções que se aproximam do ótimo, ao invés de uma solução ótima que pode ser alcançada pelos métodos clássicos de otimização determinística existentes, o que para casos mais complexos, pode se tornar impeditivo pelo seu custo de processamento. (Tavakoli, 2016; Chakraborty & Bhowmik, 2015).

A estrutura de funcionamento do SA pode ser observada na Figura 6. O algoritmo inicia com uma temperatura alta, aceitando soluções que aparentemente são menos promissoras, o que na verdade privilegia a diversidade. A medida que a temperatura vai diminuindo a aceitação das soluções tornam-se mais restritas, de acordo com a equação (3). Quando há o congelamento, ou seja, a energia (ΔE) é inferior a zero, significa que o processo está encerrado e é retornada à solução mais próxima do ideal possível.

Figura 6 - Algoritmo *Simulated Annealing*

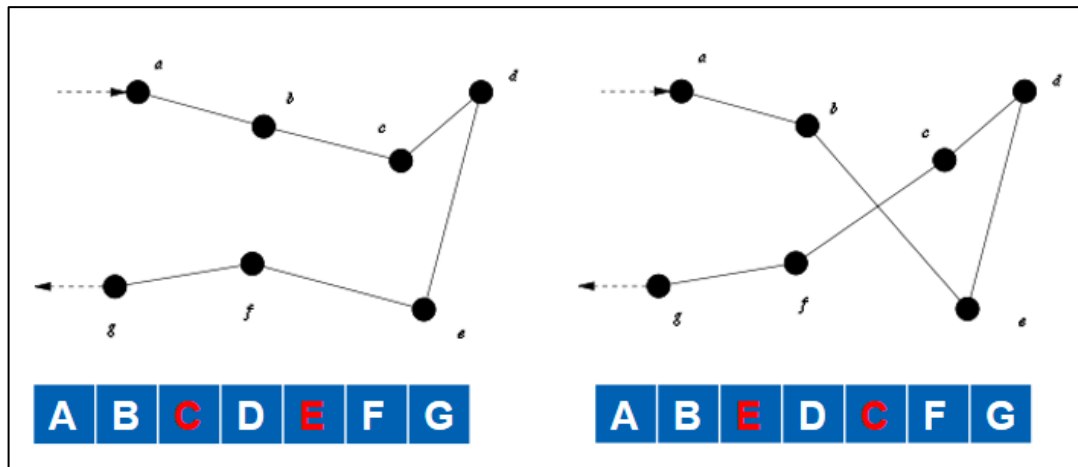
```

T = temperatura = quente (alta);
Congelamento = falso;
Enquanto (não há congelamento) {
    Repete {
        Perturba ligeiramente o sistema movendo suas partículas
        Calcula E // Altera a energia em função da perturbação
        Se ( $\Delta E < 0$ ) então
            Aceita esta perturbação, esta é a nova configuração do sistema
        Caso contrário
            Talvez aceite, probabilidade =  $\exp(-\Delta E/kT)$ 
    } até (o sistema está em equilíbrio térmico nesta temperatura- T)
    Se ( $\Delta E$  ainda diminuindo ao longo das últimas temperaturas) então
         $T = 0.9 \times T$  // Reduz a temperatura e promove mais perturbações
    Caso contrário
        congelamento = verdadeiro
}

```

O segundo algoritmo é o *K-opt*, também considerado um algoritmo de busca local. Seu processo inicia uma pesquisa no espaço amostral e faz movimentos de um determinado local do indivíduo para locais em sua vizinhança. Seu mecanismo é baseado em movimentos ou trocas entre genes dentro de um indivíduo, que poderão converter uma solução candidata em outra. Diante de um deslocamento viável, o algoritmo executa repetidamente intercâmbios visando melhorar o *fitness* do indivíduo a cada movimento. Estes movimentos podem percorrer todo o indivíduo, buscando a evolução, ou executar movimentos pré-determinados, todavia, a solução candidata somente é substituída caso o *fitness* seja melhorado. O *k* representa a quantidade de movimentos que podem ser feitos de uma única vez, como por exemplo, 2-kopt, 3-kopt, etc. (Helsgaun, 2009). A Figura 7 exhibe um exemplo simples do movimento 2-opt.

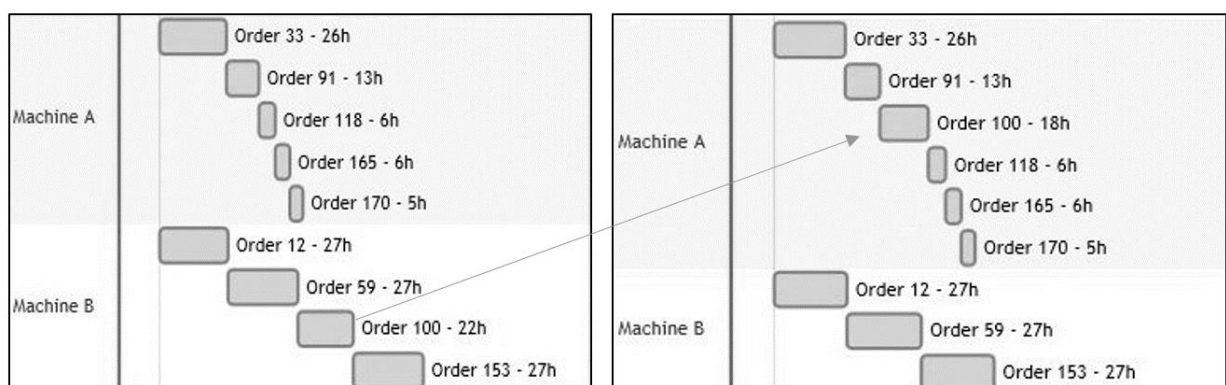
Figura 7 - Exemplo movimento 2opt



Já o *Load Balance* é um algoritmo utilizado para melhorar o sequenciamento de produção, que consiste na adequada distribuição de trabalhos para cada máquina, visando cumprir os objetivos de produção. Em função das mais diversas aplicações no mundo real, os pesquisadores têm estudado o problema do sequenciamento de produção paralelo. Encontrar o equilíbrio na distribuição dos trabalhos é fundamental para maximizar o uso dos recursos. Diante deste cenário o LB está sendo utilizado com sucesso nas aplicações para reduzir o tempo total de trabalho. A máquina com o tempo máximo de trabalho é denominada gargalo, que deverá ser tratada, ou seja, redistribuída, para evitar o estrangulamento do sistema. Assim o balanceamento de carga é utilizado para remover o gargalo existente, buscando atingir o desempenho global ótimo. (Rajakumar et al., 2007).

Na Figura 8, podemos observar a redução do *makespan* de 103 horas para 81 horas, movimentando o *job* 100 para o cenário 2 (quadro à direita). Este simples movimento representou um ganho de aproximadamente 21% neste cenário hipotético.

Figura 8 - Execução do balanceamento de carga entre máquinas



A redução obtida considera o caminho crítico, ou seja, a máquina que apresenta a maior carga de trabalho durante o processamento de todas as ordens. No cenário à esquerda, somando todos os trabalhos da máquina *B* chegamos a um total de 103 horas (27 + 27 + 22 + 27). Quando o trabalho da ordem 100 é movimentado para o cenário à direita, é possível observar que a máquina *A*, apesar de receber um novo trabalho, ainda permanece com um tempo inferior, ou seja, irá demorar 74 horas (26 + 13 + 18 + 6 + 6 + 5), contra 81 horas da máquina *B* (27 + 27 + 27). Importante ressaltar que a Máquina *A* possui melhor configuração para execução da ordem 100, concluindo o trabalho em 18 horas, contra 22 horas da máquina *B*.

Segundo Bierwirth & Kuhpfahl, (2017), outro algoritmo de busca local muito utilizado pela pelos pesquisadores é o GRASP (*Greedy Randomized Adaptive Search Procedure*). Proposto pela primeira vez por Feo e Resende (1995), trata-se de um algoritmo de pesquisa iterativo. A cada iteração é construída uma nova solução através de um procedimento aleatório. A partir daí tenta-se melhorá-lo alterando os indivíduos através de procedimentos de busca local. A cada iteração um novo elemento é selecionado a partir de uma lista de candidatos (LC) utilizada na fase construtiva do GRASP. Em seguida é criada a lista restrita de candidatos (LRC), a partir de um critério pré-definido, como por exemplo utilizando os $x\%$ dos melhores indivíduos da lista de candidatos criados na fase construtiva. (Diaz et.al, 2017). A Figura 9 demonstra o funcionamento básico do GRASP, contemplando a fase construtiva, a fase de melhoramento e o critério de parada:

Figura 9 - Algoritmo GRASP

```

Procedure GRASP
input:  $i_{\text{máx}}$ 
 $f^* \leftarrow \infty$ ;
for  $i := 1$  to  $i = i_{\text{máx}}$  do
     $\chi \leftarrow \text{GreedyRandomized}()$ ;
     $\chi \leftarrow \text{LocalSearch}(\chi)$ ;
    if  $f(\chi) < f^*$  then
         $f^* \leftarrow f(\chi)$ ;
         $\chi \leftarrow \chi^*$ ;
    endif
endfor
return  $\chi$ ;

```

A última metaheurística utilizada nesta pesquisa é o VNS (*Variable Neighborhood Search*), que tem o propósito de resolver problemas de otimização global e combinatórios promovendo a troca sistemática da vizinhança dentro de uma busca local. (Zhao et al, 2017). Segundo Adib et al, (2010), a ideia central deste algoritmo é a fuga do ótimo local alterando a estrutura da vizinhança. Em sua forma básica o algoritmo explora um conjunto de vizinhanças em relação ao indivíduo corrente. Ele faz uma busca local em uma solução vizinha para um ótimo local, movendo-se para ele apenas caso haja melhoria detectada na solução. Os autores demonstram o funcionamento do VNS com o algoritmo representado na Figura 10:

Figura 10 - Algoritmo VNS

<p>Procedure VNS()</p> <p>Begin</p> <p>Selecionar o conjunto de estruturas de vizinhança ψ_k, $k = 1, 2, \dots, k_{\text{máx}}$, que será usado na busca; encontrar um solução inicial x; escolher uma condição de parada;</p>
<p>Repeat</p> <p> Faça $k \leftarrow 1$</p> <p> Repeat</p>
<p>1) Gerar um ponto aleatório x' da $k^{\text{ésima}}$ vizinhança de x ($x' \in \psi_k(x)$);</p>
<p>2) Busca local. Aplicar algum método de busca local sendo x' a solução inicial; faça x'' ser o ótimo local encontrado;</p>
<p>3) Mover ou não. Se o ótimo local é melhor que a solução atual, mover ($x \leftarrow x''$), e continue a busca com ψ_k ($k \leftarrow 1$);</p> <p>4) $k \leftarrow k + 1$;</p>
<p>Until $k = k_{\text{máx}}$</p> <p>Until Condição de parada atingida</p> <p>End; // VNS</p>

3 O PROBLEMA *JOB-SHOP PROBLEM* E *FLEXIBLE JOB-SHOP PROBLEM*

JSP E FJSP são problemas de otimização combinatória que surgiram em sistemas de produção em que os produtos são feitos sob encomenda. Nesta produção é necessário que se encaixe uma quantidade de trabalhos para um número determinado de máquinas com tecnologias diferentes. Cada máquina é capaz de processar um trabalho por vez. Os trabalhos seguem rotas de processamento diferente entre as diferentes máquinas, e um trabalho não pode ser processado em mais de uma máquina simultaneamente. O JSP tem com o objetivo minimizar o *makespan* e é considerado dentro da classe de problemas *NP-hard*. (Abdelmaguid, 2010).

O JSP trabalha com um conjunto de n *Jobs*, que é uma ordem de trabalho, ou ordem de produção de um produto específico $J = \{J_1, J_2, \dots, J_n\}$, que são processados em um conjunto de m Máquinas $M = \{M_1, M_2, \dots, M_m\}$ disponíveis no ambiente. Cada *job* será processado em uma ordem determinada em cada máquina. Um *job* é composto de uma lista ordenada de tarefas, cada uma definida pela máquina requerida e pelo tempo que esta máquina demorará para executar cada trabalho e seu respectivo tempo de processamento. (Jones & Rabelo, 1999).

O JSP por definição leva em consideração importantes restrições para ser considerado válido:

- Não há relação de precedência entre as tarefas executadas por uma determinada máquina.
- Não há relação de precedência entre tarefas de diferentes *jobs*.
- Tarefas não podem ser interrompidas.
- Cada máquina pode processar uma tarefa por vez.
- Cada *job* é produzido em uma máquina por vez.
- Cada *job* é produzido por uma sequência conhecida de operações.

Partindo do princípio que as sequências de máquinas de cada *job* são fixas, o problema a ser resolvido consiste no sequenciamento dos *jobs* em cada máquina, de forma que o tempo de execução (*makespan*), do primeiro ao último *job*, seja o menor possível, respeitando o tempo de execução na relação *job* x máquina. O *makespan* é o critério mais simples e o mais utilizado. O objetivo é encontrar a melhor solução para o problema, ou seja, com menor *makespan*. Em geral o número de restrições é muito

grande, o que torna o *Job-Shop* um dos problemas mais difíceis. Testes para problemas maiores são mais difíceis de obter o agendamento ótimo conhecido, ou seja, é complexo organizar todas as tarefas nas máquinas que satisfaçam as restrições de precedência para que o processamento se faça em um menor tempo. Daí a dificuldade avaliar a performance de algoritmos de agendamento. (Yuan & Xu, 2013; Kuhpfahl & Bierwirth, 2016).

Na Figura 11 é possível observar um exemplo bastante simples de representar, no entanto que já traz um grau de dificuldade de resolução alto. Este problema representa um cenário com 3 máquinas e 5 *jobs*, o que leva a 1.728.000 ($5!^3$) possibilidades de resolução diferentes.

Figura 11 – JOB-SHOP PROBLEM 3 X 5

JSP 3 X 5 (3 máquinas e 5 Jobs)				
J1	M1			
J2			M1	
J3	M2			
J4				M1
J5	M3			

Outra classe de problemas que tem sido alvo de estudo pelos pesquisadores é o *Flexible Job-Shop Problem (FJSP)* que é uma extensão do problema clássico de *Job-shop Problem*. Este permite uma operação ser processada por qualquer máquina a partir de um determinado conjunto. O problema consiste em atribuir cada operação em uma máquina e suas respectivas ordens, de tal modo que o tempo de conclusão máxima de todas as operações seja minimizado, melhorando assim o indicador de *makespan*. (Pezzella, et.al, 2008).

Os autores organizaram um exemplo deste problema em uma tabela, onde as linhas representam as operações e as colunas representam as máquinas. Os valores de entrada representam os tempos de processamento de cada combinação, conforme demonstrado na Figura 12, a seguir:

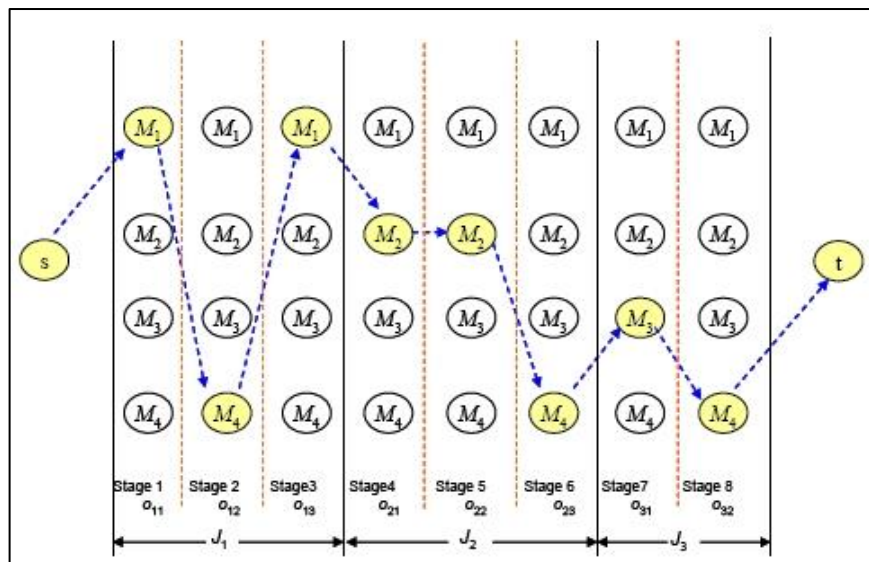
Figura 12 - Representação do FJSP

Processing time table				
	M_1	M_2	M_3	M_4
O_{11}	7	6	4	5
O_{12}	4	8	5	6
O_{13}	9	5	4	7
O_{21}	2	5	1	3
O_{22}	4	6	8	4
O_{23}	9	7	2	2
O_{31}	8	6	3	5
O_{32}	3	5	8	3

Algoritmos determinísticos não são eficazes para a resolução de FJSP mesmo em pequenas instâncias, apesar de estudos com representações gráficas disjuntivas estarem sendo desenvolvidas, não conseguem resolver problemas para instancias com 20 *jobs* e 10 máquinas. Vários procedimentos heurísticos têm sido desenvolvidos para FJSP, por exemplo, busca gulosa, busca local e metaheurísticas, tais como pesquisa Tabu, SA e GA. Eles podem ser classificados em duas categorias principais: abordagem hierárquica e abordagem integrada.

A Figura 13 exibe uma configuração tradicional desta classe de problema:

Figura 13 - Flexible Job-Shop Problem



A abordagem hierárquica tenta resolver o problema por decomposição em uma sequência de subproblemas, com dificuldade reduzida. A decomposição típica é a atribuição por sequência, proveniente da observação trivial uma vez que a atribuição é feita, o resultado de um problema de sequenciamento é um JSP, entre outros. Todos eles resolvem o problema de atribuição utilizando algumas buscas gulosas, e em

seguida resolvem o JSP resultante utilizando diferentes heurísticas de busca tabu, uma abordagem que é seguida por Zhang et al., (2011),

Uma abordagem integrada é muito mais difícil de resolver, todavia, em geral, consegue-se melhores resultados. Pezzella et.al, (2008), adotou uma abordagem integrada, propondo diferentes buscas tabu para resolver o problema, mostrando resultados computacionais provando que suas buscas tabu possuem melhor desempenho do que qualquer outra heurística desenvolvida até agora, tanto em termos de tempo de processamento quanto na qualidade da solução

No ambiente selecionado neste trabalho para aplicação das técnicas será utilizado o JSP. Isto se dá pelo fato desta indústria utilizar o modelo de produção por processo, onde a produção é dividida em várias etapas, gerando estoques intermediários. Portanto o JSP é a melhor técnica a se aplicar neste caso. Segundo relata (Yuan & Xu, 2013; Kuhpfahl & Bierwirth, 2016), o FSJP terá melhor aderência em indústrias de manufatura discreta, onde o processo é contínuo e exige uma visão antecipada de todas as etapas / operações de produção.

3.1. Trabalhos relacionados

Os estudos que estão relacionados com o foco de pesquisa deste trabalho, foram classificados em sete categorias: (i) *genetic algorithm* e; (ii) *differential evolution*, que representam as técnicas que serão aplicadas para obter o melhor *makespan* possível no tempo de processamento aceitável. Nos trabalhos pesquisados dentro destas categorias, foram avaliadas as abordagens adotadas por cada autor e sua respectiva aplicação em diferentes problemas, bem como a adoção de heurísticas para melhorar os resultados obtidos por eles; (iii) *simulated annealing*, que é aplicado durante o processo de mutação como heurística de busca local; (iv) *load balance*, que é utilizado conjuntamente com o *simulated annealing* visando equilibrar o fitness dos cromossomos para uma solução global melhor; (v) *greedy randomized adaptive search procedures*, que é uma metaheurística constituída por heurísticas construtivas e busca local que são utilizadas tanto na geração da população inicial, quanto nas mutações primárias e secundárias; (vi) *variable neighborhood search*, um método de busca em vizinhança variável, também utilizado na população inicial, quanto nas mutações primárias e secundárias; (vii) *job-shop problem*, que representa a classe de problemas que será objeto de estudo desta proposta de dissertação de mestrado. A

intenção é avaliar como a comunidade científica tem proposto a otimização do sequenciamento de produção e quais os resultados obtidos recentemente.

A fim de encontrar trabalhos que estão sendo publicados na área de inteligência artificial, foi realizado um mapeamento dos trabalhos correlatos. Neste mapeamento os principais trabalhos divulgados nos últimos anos sobre os temas de *JSP*, *AG*, *DE*, *LB*, *SA*, *GRASP* e *VNS* foram organizados e tabulados.

A Tabela 1 apresenta a abrangência das obras considerando as heurísticas, metaheurísticas e o *JSP*, que poderá conter apenas um tema específico ou qualquer combinação dos sete temas.

Tabela 1 – Principais Trabalhos Relacionados

AUTOR e ANO	AG	DE	SA	LB	VNS	GRASP	JSP
ABDELMAGUID, T. F., 2010	X						X
ADIB ET AL, 2010					X		X
BIERWIRTH, C.KUHPFAHL, J.(2017)						X	X
ČIČKOVÁ, Z., ŠTEVO, S., 2010		X					X
DIAZ ET AL, 2017						X	
FAYAD, C., PETROVIC, S., 2005	X						X
FORREST S., 1993	X						
JEONG, I-S., KIM, H., KIM, T., LEE, D. H., KIM, K. J., KANG, S., 2016			X				
JONES, A.; RABELO, L., 1999	X		X				X
LAWRYNOWICZ, A. 2011	X						X
LINDEN, R., 2012	X						
MITCHELL, 1997	X						
PEZZELLA ET. AL., 2008	X						X
RAJAKUMAR, S., ARUNACHALAM, V. P., SELLADURAI, V., 2007				X			
STORN R. PRICE K., 1997		X					
TAVAKOLI, R., SADJEDI H., POURMIR FIROOZABADI, S. M., 2016			X				
YAMADA, T.; NAKANO, R., 1997	X						X
YUAN, Y., XU, H., 2013		X					X
ZHANG ET AL., 2011	X						X
ZHAO ET AL., 2017					X		

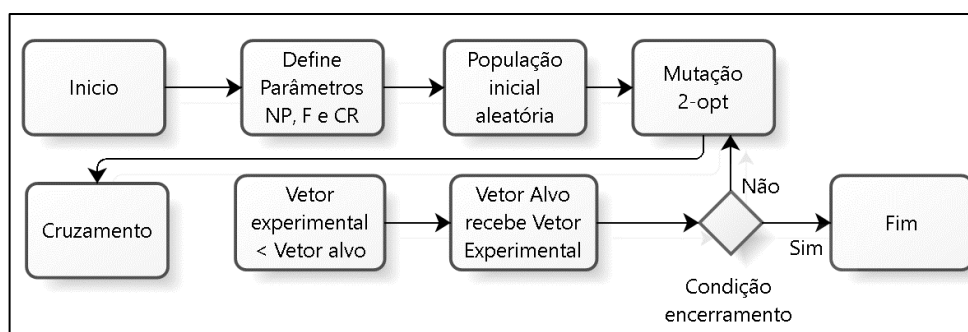
4 METODOLOGIA

Neste capítulo, contextualizamos as abordagens propostas e as características do ambiente onde as técnicas foram aplicadas, em uma indústria de alimentos, produtora de guloseimas. Também serão apresentadas as variáveis consideradas para a construção do *dataset* utilizado, a configuração do cromossomo e seus respectivos genes, bem como a função de avaliação *fitness*.

4.1 Abordagens propostas

Em função da dificuldade de obtenção de resultados para problemas de um ambiente real complexo, esta abordagem tem como objetivo a minimização do *makespan* do sequenciamento de produção. Foram desenvolvidos dois algoritmos, utilizando as linguagens de programação R, RStudio e o *Shiny*. O primeiro algoritmo foi escrito para executar as etapas do *Genetic Algorithm* tradicional e o segundo algoritmo representa o funcionamento da *Differential Evolution*. Ambos são descritos no Capítulo 2. Para os dois algoritmos foram criadas duas derivações: modificação na mutação, utilizando a heurística 2-opt como busca local; e modificação na mutação, utilizando a heurística *Simulated Annealing* conjuntamente com o *Load Balance*, também descritas no capítulo 2 deste trabalho. As Figuras 14 à 17, ilustram estas abordagens por meio de fluxogramas.

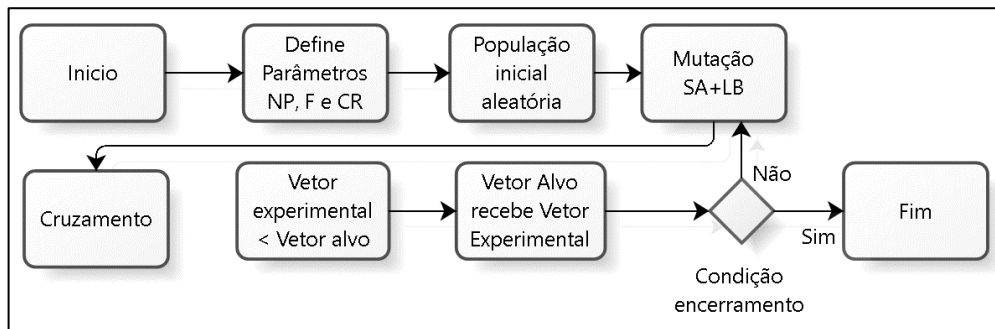
Figura 14 - Fluxograma Evolução Diferencial com Mutação 2-opt



O fluxo da heurística DE da Figura 14 demonstra as fases de definição dos parâmetros iniciais; a geração da população inicial que é feita de maneira aleatória; o processo de mutação, que usa a metaheurística de busca local *K-opt* e o cruzamento de um ponto. A partir daí o vetor experimental é comparado com o vetor alvo e caso o *fitness* do vetor experimental seja menor que o vetor alvo, o vetor alvo passa a ser

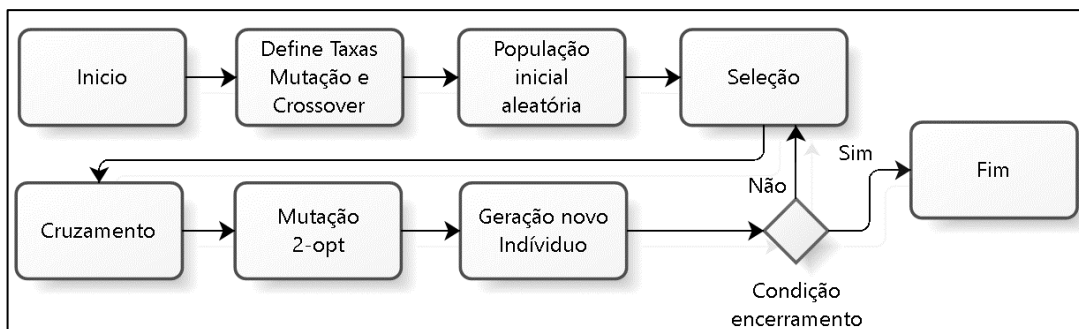
o vetor experimental. Por fim uma condição de estagnação é testada e caso esta seja atendida, o fluxo é encerrado, sendo o vetor alvo o resultado com o melhor *fitness*.

Figura 15 - Fluxograma Evolução Diferencial com Mutação SA + LB



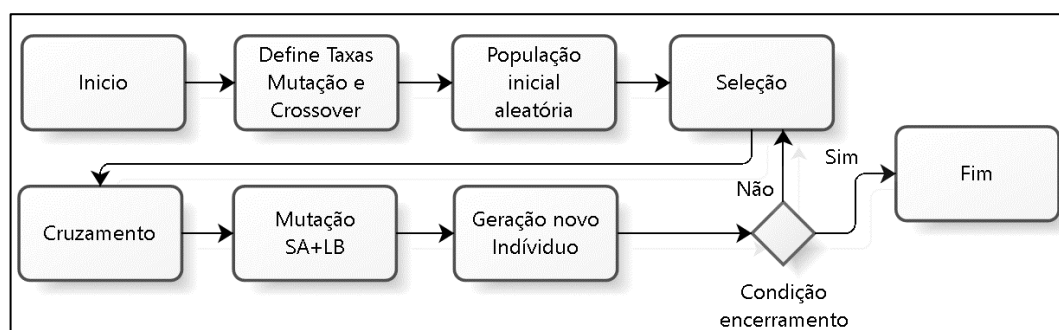
No fluxo da Figura 16 o processo é o mesmo do anterior diferenciando-se apenas pelo processo de mutação, que neste caso é utilizado dentro da mutação o *Simulated Annealing* e também o *Load Balance*.

Figura 16 - Fluxograma Algoritmo Genético com Mutação 2-opt



Neste fluxo o processo segue as etapas do algoritmo genético convencional, ou seja, a definição das taxas de mutação e cruzamento; a geração da população inicial, através do método aleatório; a seleção e cruzamento dos indivíduos. A exceção se dá por conta da mutação, onde é realizada através da metaheurística *K-OPT*.

Figura 17 - Fluxograma Algoritmo Genético com Mutação SA + LB

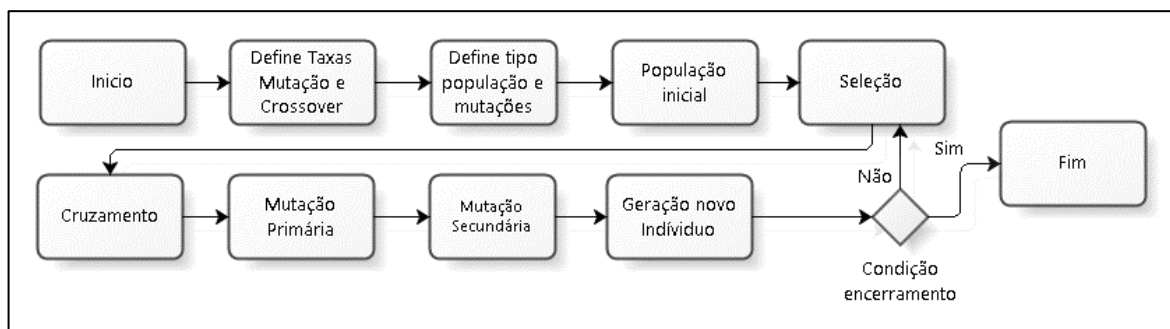


Finalmente o fluxo da Figura 17 demonstra também os mesmos passos do algoritmo genético exibido na Figura 16, no entanto utilizando a metaheurística SA com LB.

Após a aplicação dos resultados destes algoritmos no ambiente real de produção, e em função da pequena diferença prática entre os resultados do GA e do DE, avançamos na exploração do *Genetic Algorithm* com duas novas metaheurísticas: GRASP E VNS, mantendo a heurística SA e também o LB. No entanto, a partir deste cenário, incluímos uma segunda mutação no processo, denominada “mutação secundária”, a qual combina diferentes metaheurísticas, com o intuito de obter melhores *makespans*.

Durante a fase de geração da população inicial, já fazemos uso de uma das metaheurísticas: VNS, GRASP ou SA, além da geração da população do processo aleatório. Isto também ocorre tanto no processo de mutação primária, quanto no processo de mutação secundária, considerando também o LB. Na Figura 18 demonstramos os ajustes necessários no fluxograma do *Genetic Algorithm*:

Figura 18 - Fluxograma GA ajustado para novas heurísticas



O escopo de aplicação também foi alterado, anteriormente aplicamos as técnicas na linha de empacotamento de amendoins, agora passamos a atuar na fábrica de gomas, mais precisamente na etapa de semielaborados.

Vale observar que no ambiente estudado, não foi possível a aplicação das técnicas para a produção de semielaborado dos amendoins. Isto é justificado pela configuração das linhas de produção, que são bastante inflexíveis em relação aos seus respectivos produtos sendo que, a linha de amendoins crocantes, via de regra, faz apenas os amendoins crocantes, o que também ocorre com as linhas de amendoins doces e do amendoim japonês.

4.2 Definição do Problema

Para o estudo e aplicação das metaheurísticas e heurísticas de busca local abordadas nesta dissertação, utilizamos como estudo de caso, uma fabricante de balas e confeitos considerada a maior neste ramo de atividade, contando com mais de 2.000 funcionários, três unidades fabris e 4 centros de distribuição, exportando para mais de 60 países.

- **Ambiente**

A primeira etapa do trabalho limitou-se a um dos setores de empacotamento de uma das três unidades fabris. Nesta unidade, são produzidos amendoins de diferentes formatos e sabores, entre doces e salgados, compondo aproximadamente 50 SKUs (*Stock Key Units*). Nas linhas de amendoins salgados, além do amendoim japonês tradicional, também são produzidos amendoins nos sabores pimenta e cebola & salsa. Já nas linhas de amendoins doces são produzidos os amendoins chocolate, chocolate branco e amendoim confeitado colorido. Para estes sabores, existem pacotes para os segmentos de atacado e varejo. Para o segmento atacado são produzidos os pacotes de 500g, 700g, 1,010g e 5kg. Já para o segmento do varejo são consideradas as cartelas de 70g, pacotes de 90g, 100g, 150g, 200g.

Este setor está configurado para trabalhar com até 19 máquinas de empacotamento, e também é responsável pela execução de aproximadamente 200 ordens de produção por mês. A operação atua com dois turnos para a maioria das máquinas, mas havendo gargalos, o terceiro turno é ativado para cumprir o planejamento de produção. São consideradas 14,5 horas úteis para dois turnos e 21 horas no caso de 3 turnos. Recentemente foi realizada a cronoanálise em todas as máquinas visando melhorar a acuracidade do planejamento e a consequente entrega de produtos. Do número total obtido na cronoanálise, a organização considera apenas 80% desta capacidade, pois os 20% restantes são considerados para limpeza, manutenção, troca de produtos e sabores, de acordo com estudos internos realizados pela empresa. Diante deste cenário, considerando um mês de produção a partir do volume proposto pela área de planejamento da empresa, este volume foi dividido igualmente durante as quatro semanas do mês. Em função do processo de atendimento ao cliente ser pelo método *make-to-stock* não houve a necessidade de impor restrições para o sequenciamento das ordens durante o mês. Considerando este cenário, um *dataset* de 47 linhas (*Jobs*) e 19 colunas (máquinas) foram criadas

as combinações de cada *job* para cada máquina. Como alguns produtos não podem ser empacotados por determinadas máquinas, no *dataset* foi utilizada a técnica de penalização, atribuindo para esta combinação um custo elevado para que os algoritmos possam identificá-los e evitá-los, durante o processo de geração dos indivíduos, seja na geração da população inicial, como nas etapas de cruzamento e mutação.

Cada um dos aproximadamente 200 *Jobs*, estão classificados em apenas uma gramatura, no entanto, podendo ser produzido em mais de uma máquina. A exceção são os produtos de 5 Kg, que poderão neste ambiente ser produzidos em apenas uma máquina. Portanto, esta não é uma tabela dos *Jobs*, mas sim uma tabela de suporte para o cálculo do *fitness*, pois cada combinação gramatura/máquina oferece um tempo de processamento diferente. Na Tabela 2 é possível observar esta configuração.

Tabela 2 - Possibilidades de produção: Máquina versus Produto - Amendoins

Gramatura	Máquina																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
70g		☺	☺	☺								☺	☺		☺		☺	☺	☺	
90g							☺	☺								☺				
100g				☺					☺	☺			☺		☺	☺	☺			
150g	☺						☺	☺					☺	☺				☺		
200g				☺			☺	☺						☺		☺				
500g				☺	☺	☺	☺	☺						☺		☺				
700g					☺	☺										☺				
1,010g							☺	☺												
5kg																				☺

Para a segunda etapa deste trabalho, considerando as novas heurísticas, avançamos para a fábrica de gomas, nesta unidade trabalhamos tanto com a fase de produção de semielaborados, quanto na fase de empacotamento, bem como havíamos feito na unidade de fabricação de amendoins.

Nesta unidade são produzidas gomas de diferentes formatos e sabores, entre gomas de amido de milho e gelatinas, compondo aproximadamente 99 diferentes produtos. Dentro da mesma unidade fabril, existem 3 setores, denominados: Goma 1, Goma 2 e Goma 3, cada qual com a sua respectiva capacidade e necessidade de recursos. Cada produto é fabricado para um propósito de mercado diferente, incluindo o mercado nacional e mercado internacional. Existem apresentações para os

segmentos de atacado e varejo. Para o segmento de atacado são produzidas sete apresentações diferentes, visando atingir um grande número de estabelecimentos. São elas: 300g, 450g, 500g, 600g, 700g, 1KG e 2,5KG. Já para o segmento de varejo, o número é um pouco maior, atualmente contando 11 apresentações: Cartelas de 8g, 20g, 30g, 32g e 48g, além das apresentações de 70g, 100g, 142g, 180g, 190g e 200g.

O setor de empacotamento está configurado para trabalhar com até 11 máquinas, e também é responsável pela execução de aproximadamente 350 ordens de produção por mês. A operação atua com três turnos para a maioria das máquinas, mas havendo ociosidade, algumas máquinas são desligadas durante o terceiro turno. São consideradas 14,5 horas úteis para as operações que requerem dois turnos e 21 horas no caso de 3 turnos. Considerando um mês de produção a partir do volume proposto pela área de planejamento da empresa, este volume foi dividido igualmente durante as quatro semanas do mês, apresentando um novo *dataset* de 77 linhas (*Jobs*) e 11 colunas (máquinas) foram criadas as combinações de cada *job* para cada máquina. Demonstramos na Tabela 3 a respectiva configuração.

Tabela 3 - Possibilidades de empacotamento: Máquina versus Produto - Gomas

Gramatura	Máquinas										
	A	B	C	D	E	F	G	H	I	J	K
8g	😊										
20g			😊		😊			😊			
30g					😊			😊	😊		
32g	😊									😊	
48g		😊									
70g				😊	😊						
100g			😊	😊	😊	😊					
142g					😊	😊	😊				
180g					😊	😊	😊				
190g						😊	😊				
200g					😊	😊	😊				
300g					😊	😊	😊				😊
450g					😊		😊				
500g					😊	😊	😊				😊
600g					😊	😊	😊				😊
700g					😊	😊	😊				😊
1kg					😊	😊	😊				😊
2,5kg							😊				😊

Assim como na unidade de amendoim, a unidade de gomas possui uma cronoanálise recente. O processo de atendimento ao cliente também é efetuado pelo

método *make-to-stock*, bem como o percentual sobre número total obtido na cronoanálise, portanto foram considerados 80% do valor total da cronoanálise. Preservamos para esta fábrica a mesma técnica de penalização utilizada anteriormente.

Os setores de gomas (1, 2 e 3) são responsáveis pela disponibilização do produto semielaborado, justamente o processo antes do empacotamento. De maneira análoga demonstramos na Tabela 4 suas configurações, que é composta dos três setores e uma variedade de 30 formatos diferentes.

Tabela 4 - Produção de Gomas Semielaboradas

SKU	Setores		
	Goma 1	Goma 2	Goma 3
1	😊	😊	😊
2	😊	😊	😊
3		😊	
4		😊	
5		😊	
6		😊	
7		😊	
8		😊	
9	😊	😊	😊
10	😊	😊	
11		😊	
12		😊	
13	😊	😊	😊
14	😊	😊	😊
15		😊	
16		😊	
17	😊		😊
18	😊	😊	😊
19		😊	
20	😊	😊	😊
21	😊	😊	😊
22		😊	
23	😊	😊	😊
24	😊	😊	😊
25	😊		
26	😊		😊
27		😊	
28	😊	😊	
29		😊	
30	😊		😊

Diferentemente das Tabelas 2 e 3, a Tabela 4, representa a possibilidade de cada *job* em cada máquina. Neste caso, não se trata de um grupo de produtos, mas sim do próprio produto (SKU).

- **Representação cromossomial**

Tanto para o algoritmo genético, quanto para a evolução diferencial, a representação cromossomial é fundamental para o bom funcionamento dos algoritmos. (Abdelmaguid, 2010). Por se tratar de um problema combinatório, utilizamos a representação por números inteiros. Para todos os experimentos, nossa população, definida de maneira empírica, é composta por 100 indivíduos. Cada indivíduo ou cromossomo é composto por 47 ou 77 genes, que representam as ordens de produção ou *Jobs* do mês. Cada gene poderá conter um número inteiro entre 1 a 11 ou de 1 a 19, que representam as máquinas possíveis, que poderão receber ordens de produção. Existem diversas máquinas que não poderão ser atribuídas a determinados *Jobs* pela limitação técnica daquela máquina em produzir determinado produto. Para isso, utilizamos uma penalização, atribuindo um custo elevado para estas situações para que os algoritmos saiam rapidamente deste local do espaço amostral. Portanto, durante os processos de geração da população inicial e demais populações, bem como durante o processo de cruzamento e mutações primárias e secundárias, tomou-se o cuidado de não efetuar a atribuição de uma máquina a um *JOB*, com elevado custo. Na Tabela 5 é demonstrado um exemplo aleatório de uma possível solução de cromossomo contendo 47 genes, que representam o problema estudado na fábrica de amendoins.

Tabela 5 - Composição do cromossomo

6	6	4	4	16	11	5	5	16	4	14	2	5	14	19	4	17	8
4	3	18	8	7	7	7	19	9	8	7	13	7	7	1	8	18	7
7	8	19	7	19	8	8	7	19	19	8							

- **Função *Fitness***

Em geral, a aptidão do indivíduo é calculada pela da função objetivo, que está diretamente relacionada a cada problema. Neste trabalho cada indivíduo ou cromossomo com os seus respectivos genes são a entrada para uma função de análise de desempenho, que irá efetuar o *fitness* de cada indivíduo, possibilitando

assim a comparação e evolução dos diferentes indivíduos para as diferentes gerações.

O *makespan* é calculado a partir da equação (4). Primeiramente, criamos um vetor contendo as máquinas e suas respectivas alocações:

$$Cost_m = \sum_{i=1}^j C(j, m) \quad (4)$$

Em que m representa o número da máquina, que para o nosso *dataset* terá um intervalo entre 1 e 11 ou 1 e 19, ou seja, o número de máquinas possíveis para cada *job* e j representa a quantidade de *Jobs* contidos em cada cromossomo. C é a matriz que possui o custo em horas da execução de um determinado *job* (j), em uma determinada máquina (m).

Após calcular o custo para cada máquina, o *fitness* é obtido a partir do caminho crítico (*max*), ou seja, a máquina que contiver o maior número de horas alocadas, será o caminho crítico, e conseqüentemente representará o *makespan*, conforme a equação (5):

$$f(x) = \max(Cost_m) \quad (5)$$

Em que x é o cromossomo a ser calculado e $Cost_m$ é o vetor que contém a duração de tempo para execução de um conjunto de *jobs* em cada máquina.

- **Seleção, Mutação e Cruzamento**

Os algoritmos desenvolvidos utilizaram a seleção por *ranking* linear. Já no processo de mutação, além do balanceamento de carga, a operação 2-opt também foi utilizada, enquanto no cruzamento dos indivíduos foi utilizada a técnica de um ponto.

Na segunda etapa deste trabalho, incrementamos uma nova mutação, denominada mutação secundária. Nesta, além de utilizarmos as metaheurísticas anteriores, também acrescentamos as metaheurísticas GRASP e VNS.

4.3. Desafios desta indústria

Durante as entrevistas, também foram levantados os desafios de planejamento e produção na categoria de gomas, que se tornam diferenciais competitivos para um posicionamento estratégico e perene da empresa, destacando-se:

Segundo Semestre: A demanda por este tipo de produto cresce muito no segundo semestre e em caso de não conseguir formar estoques no primeiro semestre, existem rupturas no abastecimento ao mercado. Por outro lado, produzir além da demanda do primeiro semestre incorre no aumento de custos de diversas áreas, tais como estoque, aquisição de matéria prima, elevação do número de funcionários, entre outros.

Exportação / Variação do dólar: A variável exportação está ligada à variação do dólar. Caso o dólar esteja acima do dólar orçado para atingimento do EBITDA, há uma inclinação para o atendimento a este mercado, que exige um processo mais rigoroso de produção, considerando inclusive outras etapas no processo.

Pedidos do canal de vendas *Key Account*: Este é um canal de vendas que requer atenção especial pelos seus volumes e influência no mercado como um todo. Não é incomum que surjam pedidos emergenciais deste canal de distribuição, em especial no final do mês, o que pode impactar nos planos de produção e consequentemente exigir um sequenciamento de produção mais rápido e inteligente.

Parque Fabril: Existem máquinas de produção que possuem desempenho aquém do especificado pelo fabricante em função dos vários anos de vida do equipamento, o que diminui sua velocidade de produção. Por outro lado, existem equipamentos com alta performance, e que pode produzir o mesmo que as máquinas de baixa performance.

Processo de cura (descanso): O processo de descanso do produto é longo, podendo chegar até 34 horas. A empresa tem trabalhado de maneira contínua para reduzir este tempo sem perda de qualidade do produto.

Cumprimento dos objetivos: Apesar da empresa ter efetuado a cronoanálise recentemente, ainda há uma inferência sobre os números planejados, o que dificulta o cumprimento dos objetivos estipulados pela área de planejamento.

Métodos – Instruções de trabalho: Atualmente não há para todos os processos produtivos métodos de execução padrão de fabricação (instrução de trabalho), o que também gera oscilações na performance de produção. Entretanto, atualmente, a empresa atua através da sua diretoria industrial com programa que busca a excelência operacional.

Mix de produtos: Existem 99 produtos diferentes a serem confeccionados em três linhas de produção. Como cada produto não pode rodar em qualquer uma das três linhas, temos a combinação de 241 possibilidades, o que torna a elaboração do plano de produção extremamente complexo, seja diário, semanal ou mensal.

Diante destes complexos desafios, houve o consenso entre o mestrando e seu orientador, em aprofundar o desafio denominado Segundo Semestre, buscando a aplicação de técnicas que tragam sinergia tanto no âmbito da produção de semielaborados, quanto no empacotamento. Em especial pela maior disponibilidade de informações deste tema e visando otimizar os recursos de máquina e mão de obra disponíveis, aumentando assim o volume de produção, proporcionando melhor utilização dos recursos e equilíbrio nos volumes de produção, tanto no primeiro, quanto no segundo semestre.

Sobretudo, será utilizada a metodologia de estudo de caso por meio de pesquisa quantitativa de investigação por meio de dados secundários, visando otimizar o processo de sequenciamento de produção e empacotamento de gomas da mesma indústria objeto de estudo, onde já foram apresentados resultados na otimização no sequenciamento de produção do centro de trabalho empacotamento na unidade produtora de amendoins.

Este projeto visa otimizar o processo de sequenciamento de produção desta empresa que atua no ramo alimentício. Atualmente, o sequenciamento de produção utilizado está limitado ao comprometimento e experiência do time de planejamento e sujeito a vulnerabilidades. Visando atingir os objetivos propostos, a metodologia está dividida em sete macro etapas: entrevista com os colaboradores da área de planejamento, seleção dos dados a serem coletados, pré-processamento dos dados, transformação dos dados, desenvolvimento da aplicação, introdução das metaheurísticas e a apuração e divulgação dos resultados, as quais serão detalhadas a seguir.

- **Entrevista com os membros da área de planejamento**

Levando em consideração que o planejamento de produção e o consequente sequenciamento das ordens tem sido realizado ao longo de algumas décadas, utilizando planilhas eletrônicas e especialmente a experiência adquirida pelos programadores de produção ao longo do tempo, julga-se de grande importância, antes da elaboração e aplicação dos algoritmos e suas respectivas heurísticas, a coleta de informações sobre o ambiente de produção e as respectivas restrições de produção, gargalos e nuances do negócio de guloseimas doces e salgadas.

- **Seleção dos dados a serem coletados**

No processo de sequenciamento das ordens de produção, ou também conhecido como *Jobs*, vários dados devem ser levados em consideração para que esta etapa de seleção seja bem-sucedida. Portanto, após a entrevista (etapa anterior), temos exatamente quais os dados serão imprescindíveis para a execução do trabalho.

- **Pré-Processamento dos dados**

A fase de pré-processamento de dados é utilizada para avaliar a qualidade dos dados obtidos na fase de seleção. Após a coleta dos dados, faz-se necessário a seleção dos respectivos atributos de cada dado citado no tópico anterior, a limpeza de dados que estão registrados erroneamente, dados ausentes ou inconsistentes.

- **Transformação dos dados**

Nesta etapa os dados são consolidados e transformados para uma configuração apropriada para ser entendido pelos algoritmos, formando assim a representação cromossomial. Esta transformação é fundamental para traduzir os dados do nosso problema em uma maneira viável para ser processada pelo computador. A menor parte desta informação é denominada gene, que deverá ser indivisível, e um determinado conjunto de genes é denominado cromossomo.

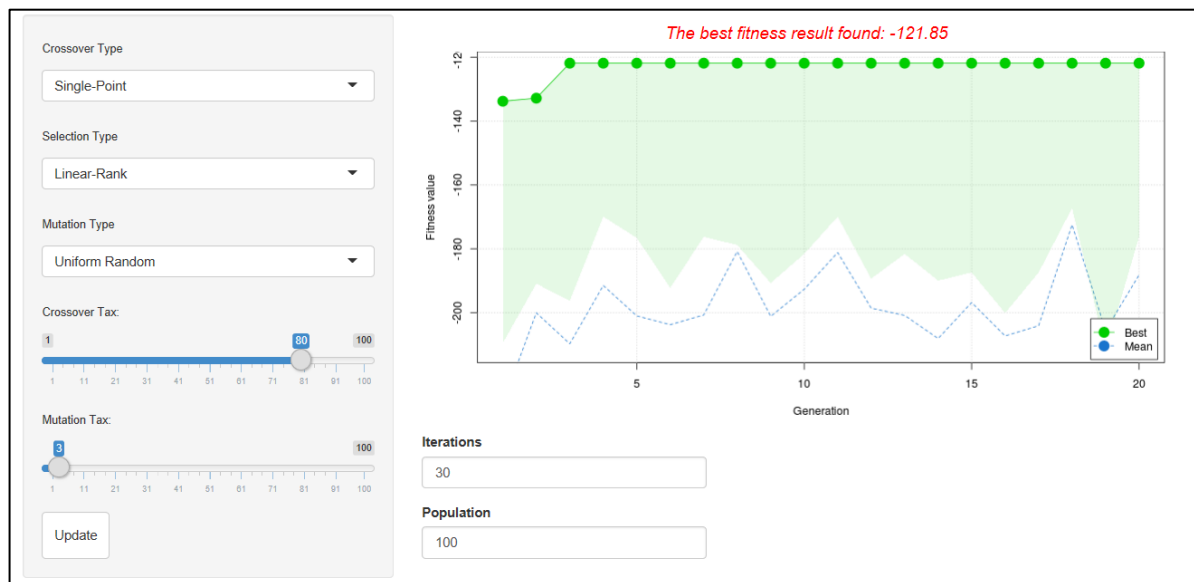
- **Desenvolvimento da aplicação**

Na primeira parte do trabalho, para atingir o objetivo de automatizar o sequenciamento de produção, foram utilizadas as linguagens de programação R, R Studio, e *Shiny*, para desenvolvimento da aplicação usando as técnicas de busca e otimização e suas respectivas metaheurísticas de busca local. Na Figura 19 é possível

observar a aplicação, que reutiliza bibliotecas do R. Esta versão foi entregue à empresa, e também pode ser acessada no endereço eletrônico: <https://jssp.shinyapps.io/Scrucca/>.

No componente de aplicação à direita da Figura 19 são definidas as formas de cruzamento, seleção e mutação, bem como as taxas de cruzamento e mutação também. Ainda no mesmo componente, o botão “Update” poderá ser acionado para executar a aplicação após alteração nos parâmetros. No componente à esquerda da Figura um gráfico de convergência é exibido, que contém o valor do fitness e a média de cada geração obtidos a cada iteração. Como título deste gráfico é exibido o resultado final do *fitness*. Logo abaixo deste gráfico, é possível alterar o número de iterações e o tamanho da população.

Figura 19 - Aplicação R



Para a segunda parte do trabalho, foi desenvolvida uma nova aplicação em Java, com o incremento das metaheurísticas GRASP e VNS. Esta aplicação pode ser observada na Figura 20:

Figura 20 - Aplicação JAVA

The screenshot shows a Java application window with the following configuration options:

- Tipo de Mutação Primária: GRASP
- Tipo de Mutação Secundária: Simulated Annealing
- Geração dos indivíduos: Simples
- Tamanho da População: 300 Indivíduos
- Taxa de Crossover: 80 %
- Taxa de Mutação: 3 %
- Taxa de Elitismo: 10 %
- Valor da Penalidade: 999

Buttons: Iniciar Algoritmo

Label: Saída

A aplicação apresentada na Figura 20 é composta pelos parâmetros de geração dos indivíduos, mutação primária e secundária, e que podem ser escolhidos através dos seguintes parâmetros: GRASP, VNS, LB, SA e tradicional. Também é possível determinar o tamanho da população, as taxas de cruzamento, mutação e elitismo, bem como o valor da penalidade, para que o algoritmo fuja de soluções infactíveis.

5 RESULTADOS

Neste capítulo apresentamos os resultados obtidos a partir das abordagens propostas e suas respectivas metaheurísticas e técnicas de busca local, no ambiente real de produção. Durante a pesquisa, foram executados três casos de testes buscando dar alta abrangência ao problema encontrado. Estes resultados estão divididos em: I. Empacotamento para a família de produtos de amendoins; II. Empacotamento para a família de produtos de gomas e III. Produção de semielaborados de gomas. Os detalhes dos resultados destes casos de testes serão apresentados aqui, bem como os respectivos parâmetros de execução utilizados para cada caso de teste.

5.1 Empacotamento de Amendoins

Neste primeiro caso de teste foram utilizadas as metaheurísticas GA e DE, hibridizando-as com as técnicas de busca local SA, LB e K-OPT. Neste cenário, os parâmetros apresentados na Tabela 6 foram utilizados para a aplicação desenvolvida em R/R Studio/*Shiny* executar 188 *Jobs* (ordens de produção), para 19 máquinas disponíveis, considerando as respectivas restrições.

Tabela 6 - Parâmetros de Execução – Empacotamento de Amendoins

Parâmetro	Valor
Tamanho da população	100 indivíduos
Taxa Crossover / Mutação	80 % / 3%
Fator de Ponderação	0.8
Critério Parada	Estagnação em 100 gerações
Seleção	Torneio
Jobs/Máquinas	188 / 19
Espaço de busca	Fatorial (188)¹⁹
Execuções	30
Taxa SA / 2-opt	20 %

Após executar cada um dos 6 cenários por 30 vezes, obtivemos os seguintes resultados, conforme demonstra a Tabela 7:

Tabela 7 - Resumo dos resultados obtidos – Empacotamento de Amendoins

Metaheurística	Heurística	Média	Desvio Padrão	Tempo Médio (minutos)
GA	-	303,27	9,76	10
DE	-	320,95	8,83	5
GA	2-opt	298,85	9,40	12
DE	2-opt	318,88	7,48	60
GA	SA+LB	283,27	4,55	28
DE	SA+LB	278,49	1,24	128

Nesta tabela, foram reunidas as configurações e resultados obtidos para as duas metaheurísticas e suas respectivas derivações. Na coluna metaheurística temos o GA (*Genetic Algorithm*) e DE (*Differential Evolution*). A seguir, identificamos a heurística utilizada para a busca local, podendo ser “-“, ou seja, neste caso foi utilizado o algoritmo tradicional, sem modificação; 2-opt, que faz a troca entre dois genes do cromossomo até que se obtenha otimização do *fitness*. No entanto, se não há esta otimização ao final do cromossomo, sua configuração original é mantida; SA é aplicado durante o processo de mutação juntamente com o balanceamento de carga, também como heurística de busca local. As colunas Média e Desvio Padrão são calculadas a partir dos resultados dos melhores fitness identificados durante as 30 execuções de cada versão dos algoritmos. Por fim, a coluna Tempo Médio representa o custo computacional em minutos para um computador com 4GB de memória e processador Intel Core i5 2,20GHz.

Aplicou-se inicialmente o teste de normalidade indicando que os resultados pertencem a uma distribuição normal. Diante disto, foi realizado um teste t para múltiplas comparações, considerando um valor p de 0.005. As diferenças encontradas entre as abordagens são descritas a seguir.

Os resultados listando os métodos por ordem de desempenho (do melhor para o pior, após 30 interações, estamos resumidos, conforme demonstrado na Figura 21.

Figura 21 - Resultados por método

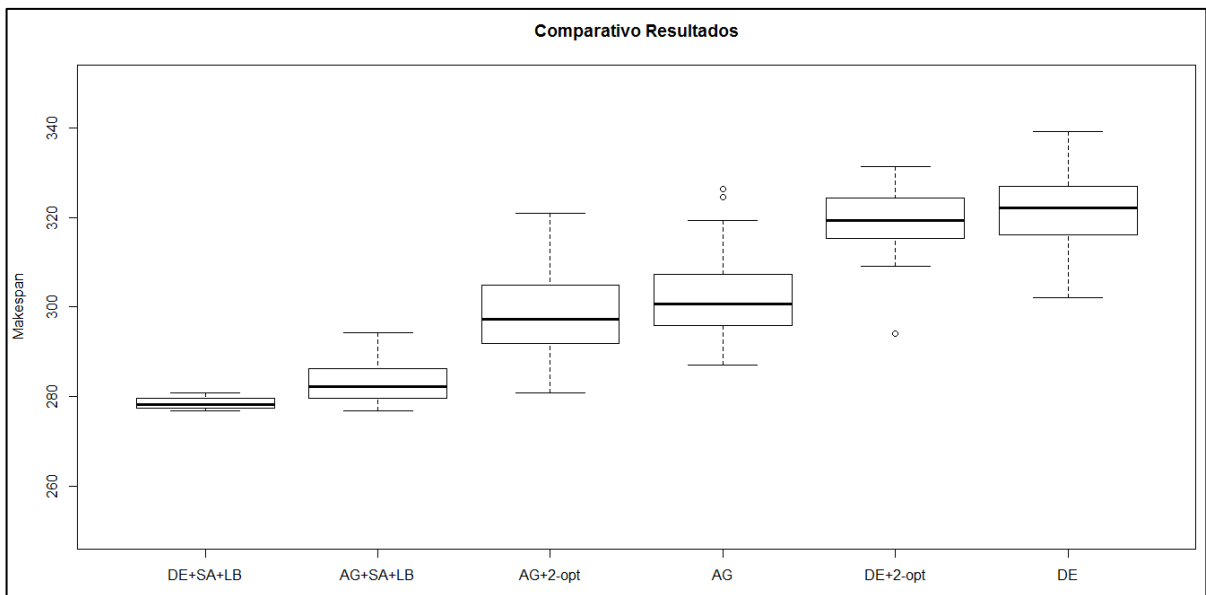
<u>DE+SA+LB</u>	<u>GA+SA+LB</u>	<u>GA+2-opt</u>	<u>GA</u>	<u>DE+2-opt</u>	<u>DE</u>
-----------------	-----------------	-----------------	-----------	-----------------	-----------

Nesta lista, a linha superior e/ou inferior é utilizada para indicar quais métodos não apresentaram diferenças estatisticamente significativas quando comparado com

as outras abordagens. Neste caso, 4 abordagens diferem significativamente em relação ao valor do *makespan* obtido, uma vez que estes métodos possuem uma linha superior/inferior de forma isolada. Entretanto, se dois ou mais métodos foram sublinhados juntos, isso significaria que a diferença observada entre estes métodos não foi estatisticamente significativa. Diante disto, podemos observar que não há diferenças significativas entre as abordagens DE+2opt e DE. Finalmente, de acordo com o teste *t* utilizado, a abordagem DE+SA+LB apresentou resultados significativos, com 95% de confiança em relação as outras abordagens utilizadas.

Para este cenário, a abordagem evolução diferencial adaptada com uma técnica de busca local apresentou resultados mais promissores quando comparado com as outras abordagens para o cenário real e complexo considerado neste caso, conforme demonstrado no gráfico *boxplot* exibido na Figura 22:

Figura 22 - *Boxplot*: Resultados Obtidos – Amendoins



Comparando os resultados obtidos com o sequenciamento de produção efetuado com a ajuda de planilhas eletrônicas, para os mesmos meses utilizados para elaboração dos *datasets* nesta dissertação, os programadores de produção conseguiram otimizar as ordens de produção no primeiro caso de testes, ou seja, no setor de empacotamento de amendoins, nas respectivas máquinas em 423 horas, considerando as restrições de máquinas versus produtos. Este trabalho durou cerca de 16 horas, (dois dias úteis). Quando comparado com os resultados obtidos através das técnicas utilizadas nesta dissertação, alcançamos uma redução de 34%, ou seja,

os resultados obtidos através dos algoritmos conseguiram sequenciar os mesmos trabalhos em aproximadamente 279 horas.

5.2 Empacotamento de Gomas

No segundo caso de teste, foi utilizada a metaheurística GA Tradicional, hibridizando-a com as técnicas de busca local SA, LB, GRASP, VNS e K-OPT. Neste cenário os parâmetros apresentados na Tabela 8 foram utilizados para a aplicação desenvolvida em JAVA executar 140 *Jobs* (ordens de produção), para 11 máquinas disponíveis, considerando as respectivas restrições.

Tabela 8 - Parâmetro Empacotamento de Gomas

Parâmetro	Valor
Tamanho da população	100 indivíduos
Taxa Crossover / Mutação	80 % / 3%
Critério Parada	Estagnação em 100 gerações
Seleção	Torneio
Jobs/Máquinas	140 / 11
Espaço de busca	Fatorial (140)¹¹
Execuções	30
Taxa SA / 2-opt	20 %
Conjunto soluções GRASP	50
Taxa de aceite GRASP	80%
Tamanho da Vizinhança VNS	50
Critério de parada GRASP/VNS	Estagnação em 5 iterações

Para este caso, foram gerados 100 cenários do *Genetic Algorithm*, considerando 4 formas diferentes para geração da população (1-Aleatória, 2-SA, 3 – VNS, 4-GRASP), 5 formas diferentes de executar a primeira mutação (1-Tradicional, 2- SA, 3-VNS, 4-GRASP, 5-LB) e 5 formas diferentes de efetuar a mutação secundária (1-Puro, 2-+AS, 3-+VNS, 4-+GRASP, 5-+LB).

Após executar cada um dos 100 cenários por 100 vezes, obtivemos os resultados e apresentamos os 4 melhores de acordo com a metaheurística utilizada para a geração da população, conforme pode ser observado na Tabela 9. No entanto, a lista completa dos 100 resultados de todas as combinações possíveis entre a

geração da população, mutação primeira e mutação secundária, podem ser consultados no apêndice A.

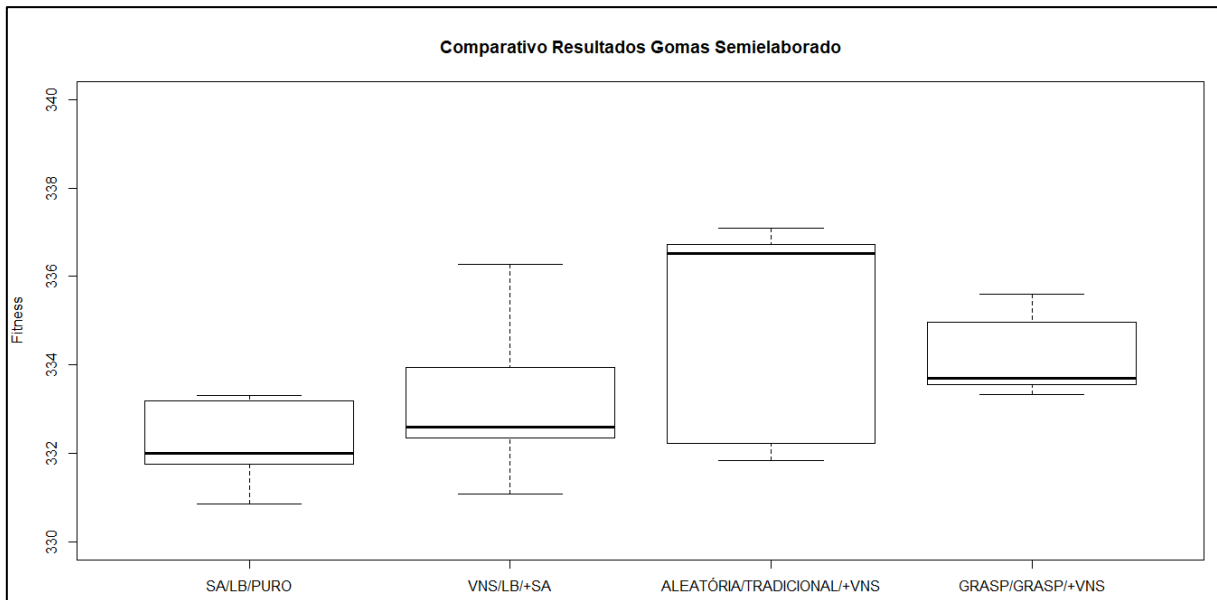
Tabela 9 - Resultados Empacotamento de Gomas

Geração da População	Mutação Primária	Mutação Secundária	Melhor Fitness	Fitness Médio	Desvio Padrão	Tempo Médio (minutos)
SA	LB	Puro	330,86	332,22	0,92	2
VNS	LB	+AS	331,08	333,25	1,76	10
Aleatória	Tradicional	+VNS	331,18	332,89	1,33	1
GRASP	GRASP	+VNS	331,27	333,80	3,65	9

Na Tabela 9 são reunidas as configurações e resultados obtidos para as quatro melhores técnicas de busca local utilizadas durante a geração dos indivíduos e suas respectivas mutações, tanto a primária, quanto a secundária, selecionadas através de um ranking dos melhores *fitness* obtidos. A seguir, temos a coluna Melhor Fitness, onde é exibido o resultado da melhor combinação possível entre as ordens de produção e o setor produtivo disponível. As colunas Média e Desvio Padrão são calculadas a partir dos resultados dos melhores *fitness* identificados durante as 100 execuções de cada versão dos algoritmos. Por fim, a coluna Tempo Médio representa o custo computacional em minutos para um computador com 4GB de memória e processador Intel Core i5 2,20GHz.

Para este cenário, a abordagem *Genetic Algorithm* adaptada para geração dos indivíduos e durante o processo de mutação primária, utilizando o LB, somado a implementação de mutação secundária utilizando a técnica de um único ponto (Puro), foram obtidos melhores resultados quando comparado com as outras abordagens, conforme demonstrado no gráfico *boxplot* exibido na Figura 23:

Figura 23 - Boxplot - Empacotamento de Gomas



Já para o caso de teste no empacotamento no setor de gomas, alcançamos um melhor *fitness* de 331 horas por meio das metaheurísticas e buscas locais utilizadas versus uma programação manual de 399 horas, obtendo assim uma redução de 22% na quantidade de horas necessárias para conclusão de todos os Jobs contidos no *dataset*. Importante destacar que o ganho alcançado neste setor, apesar de ser inferior ao empacotamento de amendoins (34% x 22%), na prática se torna bem superior em função dos volumes de produção e também do número inferior de máquinas disponíveis para empacotamento. Cada hora otimizada neste setor representa um volume expressivo de produto acabado disponível ao mercado, ampliando assim a competitividade nesta que é a principal família de produtos comercializadas por esta empresa.

5.3 Produção de Semielaborado de Gomas

Por fim, no terceiro caso de teste, também foi utilizada a metaheurística GA Tradicional, hibridizando-a com as técnicas de busca local SA, LB, GRASP, VNS e K-OPT. Os parâmetros apresentados na Tabela 10 foram utilizados para a aplicação desenvolvida em JAVA executar 47 *Jobs* (ordens de produção), para 3 setores disponíveis (Goma 1, Goma 2, Goma 3), considerando as respectivas restrições para a relação Ordem de Produção versus Setor.

Tabela 10 –Parâmetros de execução – Semielaborado Gomas

Parâmetro	Valor
Tamanho da população	100 indivíduos
Taxa Crossover / Mutação	80 % / 3%
Critério Parada	Estagnação em 100 gerações
Seleção	Torneio
Jobs/Máquinas	47 / 3
Espaço de busca	Fatorial (47) ³
Execuções	30
Taxa SA / 2-opt	20 %
Conjunto soluções GRASP	50
Taxa de aceite GRASP	80%
Tamanho da Vizinhança VNS	50
Critério de parada GRASP/VNS	Estagnação em 5 iterações

Neste último caso de teste, também foram gerados 100 cenários do *Genetic Algorithm*, considerando as mesmas diferentes formas para geração da população e execução das mutações primária e secundária, utilizadas na seção 5.2.

Após executar cada um dos 100 cenários por 100 vezes, também efetuamos um *ranking* com os melhores *fitness* e os apresentamos na Tabela 11 de acordo com a metaheurística utilizada para a geração da população. A lista completa dos 100 resultados através de todas as combinações possíveis entre a geração da população, mutação primeira e mutação secundária podem ser consultados no Apêndice B.

Tabela 11 - Resultados obtidos – Semielaborado Gomas

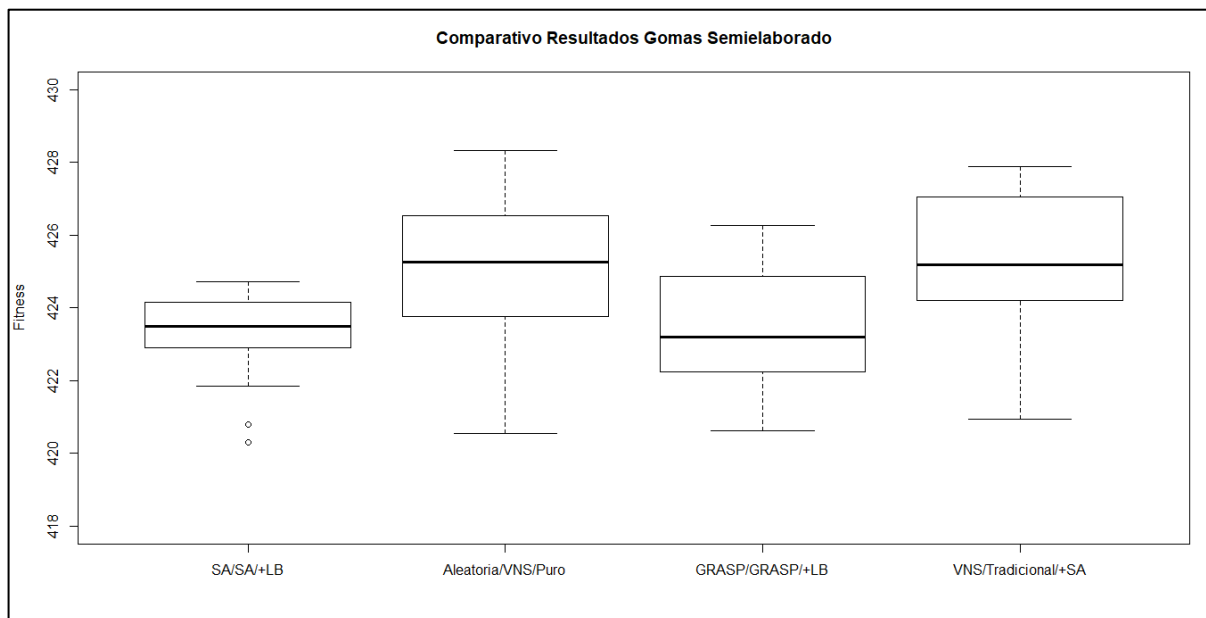
Geração da População	Mutação Primária	Mutação Secundária	Melhor Fitness	Fitness Médio	Desvio Padrão	Tempo Médio (minutos)
SA	SA	+LB	420,31	423,16	1,02	3
GRASP	GRASP	+LB	420,41	423,62	1,52	14
Aleatória	VNS	+Puro	420,56	424,98	1,95	1
VNS	Tradicional	+AS	420,97	424,80	1,67	7

Na Tabela 11 são reunidas as configurações e resultados obtidos para as quatro metaheurísticas utilizadas durante a geração dos indivíduos e suas respectivas mutações, tanto a primária, quanto a secundária. A seguir, temos a coluna Melhor

Fitness, é exibido o resultado da melhor combinação possível entre as ordens de produção e o setor produtivo disponível. As colunas Média e Desvio Padrão são calculadas a partir dos resultados dos melhores *fitness* identificados durante as 100 execuções de cada versão dos algoritmos. Por fim, a coluna Tempo Médio representa o custo computacional em minutos para um computador com 4GB de memória e processador Intel Core i5 2,20GHz.

Para este cenário, a abordagem *Genetic Algorithm* adaptada para geração dos indivíduos e durante o processo de mutação primária, utilizando o SA, somado a implementação de mutação secundária utilizando o LB, obteve melhores resultados quando comparada com as outras abordagens, conforme demonstra através o gráfico *boxplot* exibido na Figura 24:

Figura 24 - Boxplot - Semielaborados de Gomas



Por fim, neste cenário, considerando que a área de PCP conseguiu sequenciar as ordens de produção em 515 horas versus o sequenciamento otimizado, utilizando técnicas que alcançaram 420 horas. Nota-se que o sequenciamento de produção dos algoritmos propostos, obteve redução significativa no tempo necessário para conclusão dos trabalhos, gerando ganhos de produtividade e otimização de mão de obra na ordem de 18%.

Estes resultados geram um impacto muito positivo na organização, visto que o sequenciamento de produção otimizado reduz a necessidade de recursos humanos, e também aumenta a disponibilidade de máquina, possibilitando assim

aumentar os níveis de produção, proporcionalmente ao nível de redução alcançado através das técnicas utilizadas nesta pesquisa.

Como produto final foram entregues dois sistemas computacionais. O primeiro desenvolvido em linguagem R com interface WEB, fazendo uso do R Studio e *Shiny* e o segundo desenvolvido em JAVA, ambos para o sequenciamento de produção, e com grande aderência à indústria alimentícia.

Por meio da utilização destes produtos espera-se proporcionar uma redução direta nos níveis de estoques, elevar o indicador de entrega perfeita (*OTIF - On Time In Full*), que visa a entrega dos produtos no prazo e quantidade acordados, e também reduzir a ociosidade de mão de obra e máquinas de empacotamento e produção, além dos seguintes benefícios:

- Redução de investimentos em mão de obra dedicada para o sequenciamento de produção, seja em um software especialista ou em uma planilha eletrônica;
- Redução nos investimentos de insumos e embalagens, que também deverão ter reflexos em função do menor nível de itens acabados;
- Redução de itens em obsolescência, diminuindo assim também o impacto ambiental em função do descarte de embalagens plásticas e caixas de papelão.
- Auxílio às empresas, no sentido de evitar liquidações de seus produtos em função de altos níveis de estoques, podendo inclusive comprometer a sua marca.

6 CONCLUSÕES E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões do presente trabalho e as recomendações para a continuidade dos trabalhos nesta área de estudo.

6.1 Conclusões

Com base nos resultados obtidos, podemos concluir que a utilização de algoritmos híbridos para o problema *Job-Shop* que foram propostos nesta dissertação para um caso real da indústria de alimentos se mostraram muito efetivos, tanto no aspecto teórico, mas especialmente no aspecto prático. A utilização destas metaheurísticas, tanto o genético, quanto a evolução diferencial e suas derivações com a utilização de algoritmos para busca local, em todos os casos de testes, trouxeram ganhos expressivos em relação à programação convencional.

Em todos os casos de testes houveram otimizações no sequenciamento de produção, com ganhos entre 18% a 34% quando comparamos a forma tradicional como a indústria até então elaborava seu planejamento versus a utilização dos algoritmos híbridos.

Adicionalmente devemos considerar que ao evitar planilhas eletrônicas, diminuimos o risco do erro, pelo volume de dados que é manuseado, bem como o tempo dispendido, principalmente se comparado ao custo computacional dos algoritmos.

Por fim, após as configurações sugeridas pelos algoritmos estarem validadas pela equipe de planejamento, conclui-se sua factibilidade para realizar o sequenciamento de produção otimizado no dia-a-dia.

6.2 Recomendações para trabalhos futuros

Conforme apresentado no capítulo 3 deste trabalho de pesquisa, o *Flexible Job-Shop Problem (FJSP)* é uma extensão do problema clássico *Job-Shop Problem*, e tem sido alvo de estudo pelos pesquisadores em função de seus benefícios para as atividades de otimização de sequenciamento através de operações. Recomenda-se, portanto, para trabalhos futuros a incorporação deste problema aos

cenários, heurísticas e metaheurísticas hibridizadas nesta dissertação, a fim automatizar e trazer sinergia entre as diferentes fases de um processo produtivo.

Outra oportunidade bem interessante, associada a este trabalho está a ampliação dos estudos voltados a cadeia de suprimentos (*Supply Chain*). Utilizar o sequenciamento de produção em um projeto com escopo mais abrangente, buscando integrá-lo ao planejamento de vendas e operações (*S&OP – Sales and Operation Planning*), buscando como resultado um plano de vendas, operações e estoques mais assertivos através do uso de técnicas, algoritmos e da estatística.

Entendemos também haver espaço para aprofundar este estudo através da utilização dos *datasets* criados em outras metaheurísticas e técnicas de busca local, tais como *Ant Colony* (Colônia de Formigas), Redes Neurais Artificiais, *Random Walks* (Caminhada Aleatória), Busca Tabu, entre outras.

REFERÊNCIAS

- Abdelmaguid, T. F. (2010). Representations in Genetic Algorithm for the Job Shop Scheduling Problem: A Computational Study. *J. Software Engineering & Applications*. Vol. 3, p. 1155-1162.
- Adib, M.A., Zandieh, M., Amiri M. (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*. Vol. 37, p. 282-287.
- Bierwirth, C., Kuhpfahl, J. (2017). Extended GRASP for the job shop scheduling problem with total weighted tardiness objective. *European Journal of Operational Research*, Vol. 261, p. 835-848.
- Chakraborty, S., Bhowmik, S. (2015), An Efficient Approach to Job Shop Scheduling Problem using Simulated Annealing. *International Journal of Hybrid Information Technology*. Vol. 8, p. 273-284.
- Čižková, Z., Števo, S. (2010). Flow Shop Scheduling using Differential Evolution. *Management Information Systems*, Vol. 5, p. 008-013.
- Diaz, J.A., Luna, D.E.,Vallejo, J.F.C.,Ramírez, M.S.C. (2017) GRASP and hybrid GRASP-Tabu heuristics to solve a maximal covering location problem with customer preference ordering. *Expert Systems With Applications*, Vol. 82, p. 67-76.
- Fayad, C., Petrovic, S. (2005). A Fuzzy Genetic Algorithm for Real-World Job Shop Scheduling. *Innovations in Applied Artificial Intelligence*, Vol. 3533 of the series *Lecture Notes in Computer Science*. Vol. 3533, p. 524-533.
- Forrest S. (1993). *Genetic Algorithms: Principles of Natural Selection Applied to Computation*. Science, New Series, Vol. 261, p. 872-878.
- Helsgaun, K. (2009). *General k-opt submoves for the Lin–Kernighan TSP heuristic*. Springer and Mathematical Programming Society. Vol. 1, p. 119–163.
- Jeong, I-S., Kim, H., Kim, T., Lee, D. H., Kim, K. J., Kang, S. (2016). A Feature Selection Approach Based on Simulated Annealing for Detecting Various Denial of Service Attacks. *Convergence Security*, Vol. 1, p. 1 -18.
- Jones, A., Rabelo, L. (1999). *Survey of job shop scheduling techniques*. Wiley Encyclopedia of Electrical and Electronics Engineering.
- Kuhpfahl, J., Bierwirth, C. (2016). A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective. *European Journal of Operational Research*, Vol. 252, p. 432-442.
- Lawrynowicz, A. (2011). *Genetic Algorithms for Solving Scheduling Problems in Manufacturing Systems*. *Foundations of Management*, Vol. 3, p. 493-518.
- Linden, R. (2012). *Algoritmos genéticos*. São Paulo: Ciência Moderna.
- Mcclave, J. T., Benson, P. G., Sincich, T. (2009). *Estatística para administração e economia*. 10. ed. São Paulo: Pearson Prentice Hall.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

- Mitchell, M., Forrest, S. (1994). Genetic algorithms and artificial life. *Artificial Life*, Vol. 1, p. 267–289.
- Morton, T.E., Pentico, D.W. (1993). *Heuristic Scheduling Systems*. John Wiley & Sons. N.Y.
- Pezzella, F., Morgantia, G., Ciaschetti, G. (2008). A Genetic Algorithm for the Flexible Job-Shop Scheduling Problem. *Computers & Operations Research* 35, Vol. 35, p. 3202-3212.
- Rajakumar, S., Arunachalam, V. P., Selladurai, V. (2007). Workflow balancing in parallel machines through genetic algorithm. *Int J Adv Manuf Technol*. Vol. 1, p. 1212 -1221.
- Russell, S. J. et al. (1995). *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice-Hall.
- Scrucca, L. (2013). GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software*. Vol. 53. p. 1-37.
- Storn R. Price K. (1997). Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, Vol. 11, p. 341-359.
- Tavakoli, R., Sadjedi H., Pourmir Firoozabadi, S. M. (2016). An application of simulated annealing to optimal transcranial direct current stimulation of the human brain. *Turkish Journal of Electrical Engineering & Computer Sciences*. Vol. 24, p. 1135 – 1149.
- Yamada, T., Nakano, R. (1997). Genetic Algorithms for Job-Shop Scheduling Problems. *Proceedings of Modern Heuristic for Decision Support*, Vol. 1 p. 67–81.
- Yuan, Y., Xu, H. (2013). Flexible job shop scheduling using hybrid differential evolution algorithms. *Computers & Industrial Engineering*. Vol. 65, p. 246–260.
- Zhang, G., Gao, L., Shi, Y. (2011). An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications*, Vol. 38, p. 3563–3573.
- Zhao, F., Liu, Y., Zhang, Y., Ma, W., Zhang C. 2017. A hybrid harmony search algorithm with efficient job sequence scheme and variable neighborhood search for the permutation flow shop scheduling problems. *Engineering Applications of Artificial Intelligence*. Vol. 65, p. 178-199.

APÊNDICE A – Resultados do *Genetic Algorithm* para o cenário de Empacotamento de Gomas

População / Mutação Primária / Mutação Secundária	Melhor Fitness	Média Fitness	Desvio Padrão	Tempo Execução (ms)
SA	330,86	3477,376	1,087274	332,5669
VNS	331,08	6967,416	2,779042	335,0113
GRASP	332,3	7040,6	1,804068	335,1296
GRASP	332,3	6484,6	2,761895	335,386
LB	332,4	6786,8	1,51595	334,496
Puro	333,65	6046,8	1,10632	335,116
SA	335,05	7581,2	1,172184	336,422
VNS	333,32	8303,6	0,893653	334,228
LB	331,08	9677,12	3,522472	335,5892
GRASP	332,34	11725,8	1,621562	334,924
LB	332,68	10175,4	3,056962	335,298
Puro	332,94	7940,8	1,106898	334,766
SA	331,08	9584,2	1,763571	333,246
VNS	331,77	8959,4	4,694266	339,712
SA	331,76	6187,6	3,064081	334,7584
GRASP	332,44	9709,4	2,519671	334,996
LB	332,35	5793,8	4,64828	336,134
Puro	331,79	4428,2	2,174019	334,69
SA	331,76	4987,6	1,295861	333,152
VNS	332,18	6019	2,836258	334,82
Tradicional	331,52	3742,76	2,573359	334,2256
GRASP	331,92	4098,2	2,122008	334,23
LB	331,54	3528,2	2,428163	333,372
Puro	332,05	3456,2	1,769061	334,152
SA	331,52	3759,2	3,591971	334,498
VNS	331,83	3872	2,33428	334,876
VNS	331,54	8189	2,407428	335,3536
GRASP	332,91	7528,2	1,293006	334,536
LB	334,42	7176,8	2,428692	337,906
Puro	332,26	7986,8	2,877114	335,044
SA	332,64	10178,2	1,481413	334,674
VNS	331,54	8075	1,62448	334,608
Aleatoria	331,18	1410,608	3,235573	335,2517
GRASP	331,27	10050,35	2,174512	333,5829
GRASP	331,27	8352,28	2,673249	333,6572
GRASP	332,03	7322	0,81939	333,17
LB	331,46	8131,4	2,110052	333,44
Puro	331,76	8156	2,895877	334,706
SA	331,49	8225,6	2,698626	333,176
VNS	331,27	9926,4	3,647479	333,794
LB	331,35	6508,8	1,572426	333,3112

GRASP	331,77	8121,8	1,826785	333,526
LB	331,64	5743,6	1,555679	333,358
Puro	332,34	4778	0,224321	332,59
SA	331,35	7108,8	1,83406	332,988
VNS	332,63	6791,8	1,380414	334,094
SA	331,37	8051,36	1,966684	333,7264
GRASP	332,43	6274	0,902898	333,616
LB	331,45	6264,4	2,251538	334,546
Puro	331,74	12232,8	1,680388	333,184
SA	331,58	8885	1,666721	333,44
VNS	331,37	6600,6	2,603548	333,846
Tradicional	331,64	19300,24	2,034196	333,566
GRASP	331,76	20294,2	1,219377	333,23
LB	331,64	15115	0,469084	332,26
Puro	332,08	21417	1,585492	333,256
SA	332,23	19362,6	2,058627	334,746
VNS	331,77	20312,4	2,885844	334,338
VNS	331,33	8039,08	2,431608	333,6536
GRASP	331,99	8928,6	4,119479	334,744
LB	331,33	7977,6	2,20688	333,08
Puro	332,42	7481	1,346055	334,116
SA	331,81	8502,4	1,392193	333,27
VNS	331,75	7305,8	1,318869	333,058
SA	330,86	3477,376	1,087274	332,5669
VNS	331,08	6967,416	2,779042	335,0113
GRASP	332,3	7040,6	1,804068	335,1296
GRASP	332,3	6484,6	2,761895	335,386
LB	332,4	6786,8	1,51595	334,496
Puro	333,65	6046,8	1,10632	335,116
SA	335,05	7581,2	1,172184	336,422
VNS	333,32	8303,6	0,893653	334,228
LB	331,08	9677,12	3,522472	335,5892
GRASP	332,34	11725,8	1,621562	334,924
LB	332,68	10175,4	3,056962	335,298
Puro	332,94	7940,8	1,106898	334,766
SA	331,08	9584,2	1,763571	333,246
VNS	331,77	8959,4	4,694266	339,712
SA	331,76	6187,6	3,064081	334,7584
GRASP	332,44	9709,4	2,519671	334,996
LB	332,35	5793,8	4,64828	336,134
Puro	331,79	4428,2	2,174019	334,69
SA	331,76	4987,6	1,295861	333,152
VNS	332,18	6019	2,836258	334,82
Tradicional	331,52	3742,76	2,573359	334,2256
GRASP	331,92	4098,2	2,122008	334,23

LB	331,54	3528,2	2,428163	333,372
Puro	332,05	3456,2	1,769061	334,152
SA	331,52	3759,2	3,591971	334,498
VNS	331,83	3872	2,33428	334,876
VNS	331,54	8189	2,407428	335,3536
GRASP	332,91	7528,2	1,293006	334,536
LB	334,42	7176,8	2,428692	337,906
Puro	332,26	7986,8	2,877114	335,044
SA	332,64	10178,2	1,481413	334,674
VNS	331,54	8075	1,62448	334,608
Aleatória	331,18	1410,608	3,235573	335,2517
GRASP	331,27	10050,35	2,174512	333,5829
GRASP	331,27	8352,28	2,673249	333,6572
GRASP	332,03	7322	0,81939	333,17
LB	331,46	8131,4	2,110052	333,44
Puro	331,76	8156	2,895877	334,706
SA	331,49	8225,6	2,698626	333,176
VNS	331,27	9926,4	3,647479	333,794
LB	331,35	6508,8	1,572426	333,3112
GRASP	331,77	8121,8	1,826785	333,526
LB	331,64	5743,6	1,555679	333,358
Puro	332,34	4778	0,224321	332,59
SA	331,35	7108,8	1,83406	332,988
VNS	332,63	6791,8	1,380414	334,094
SA	331,37	8051,36	1,966684	333,7264
GRASP	332,43	6274	0,902898	333,616
LB	331,45	6264,4	2,251538	334,546
Puro	331,74	12232,8	1,680388	333,184
SA	331,58	8885	1,666721	333,44
VNS	331,37	6600,6	2,603548	333,846
Tradicional	331,64	19300,24	2,034196	333,566
GRASP	331,76	20294,2	1,219377	333,23
LB	331,64	15115	0,469084	332,26
Puro	332,08	21417	1,585492	333,256
SA	332,23	19362,6	2,058627	334,746
VNS	331,77	20312,4	2,885844	334,338
VNS	331,33	8039,08	2,431608	333,6536
GRASP	331,99	8928,6	4,119479	334,744

APÊNDICE B – Resultados do *Genetic Algorithm* para o cenário de Semielaborados de Gomas

População / Mutação Primária / Mutação Secundária	Melhor Fitness	Média Fitness	Desvio Padrão	Tempo Execução (ms)
Aleatória	420,56	424,64	1,73	1.821.932
GRASP	421,13	424,70	1,75	607.607
Puro	421,75	425,03	2,02	91.158
+SA	421,13	423,87	1,22	123.726
+VNS	422,55	424,65	1,34	132.367
+GRASP	421,79	424,76	2,03	166.315
+LB	422,21	425,18	1,79	94.041
LB	420,95	424,50	1,74	222.413
Puro	420,95	424,81	1,98	10.109
+SA	421,99	424,28	1,42	26.672
+VNS	421,99	424,33	1,93	55.625
+GRASP	421,57	424,35	1,67	117.961
+LB	421,63	424,71	1,69	12.046
SA	420,95	424,34	1,59	306.666
Puro	420,95	424,15	1,44	32.203
+SA	421,75	424,61	2,08	50.735
+VNS	421,75	424,32	1,45	65.016
+GRASP	421,63	424,23	1,36	128.400
+LB	421,60	424,39	1,59	30.312
Tradicional	421,64	424,77	1,69	265.297
Puro	422,21	425,24	2,03	20.327
+SA	422,06	424,52	1,54	30.875
+VNS	422,91	425,09	1,52	58.946
+GRASP	421,99	424,46	1,82	139.242
+LB	421,64	424,54	1,42	15.907
VNS	420,56	424,91	1,84	419.949
Puro	420,56	424,98	1,95	55.492
+SA	421,80	424,63	1,58	65.580
+VNS	422,55	425,15	1,78	92.203
+GRASP	421,72	424,32	1,58	165.947
+LB	421,52	425,44	2,14	40.727
GRASP	420,41	423,50	1,45	29.382.178
GRASP	420,59	423,50	1,42	4.952.315
Puro	421,02	423,46	1,35	889.747
+SA	420,59	423,52	1,36	982.026
+VNS	421,42	423,71	1,71	938.386
+GRASP	421,75	423,61	1,33	1.103.094
+LB	420,63	423,22	1,33	1.039.062
LB	420,41	423,45	1,60	4.601.179
Puro	421,04	423,63	1,71	940.235
+SA	420,92	423,23	1,20	926.709

+VNS	420,41	423,62	1,51	848.175
+GRASP	420,41	423,42	1,80	995.300
+LB	421,07	423,35	1,75	890.760
SA	420,63	423,37	1,41	4.661.114
Puro	420,66	423,74	1,74	919.699
+SA	420,70	423,22	1,48	985.765
+VNS	421,12	423,19	1,35	923.272
+GRASP	420,63	423,01	1,16	958.879
+LB	421,39	423,70	1,17	873.499
Tradicional	420,41	423,51	1,44	10.572.583
Puro	421,08	423,71	1,48	889.986
+SA	421,78	423,63	1,19	5.053.066
+VNS	420,41	423,27	1,57	954.742
+GRASP	420,66	423,57	1,38	2.573.267
+LB	420,52	423,36	1,58	1.101.522
VNS	420,95	423,67	1,40	4.594.987
Puro	421,79	424,01	1,36	843.201
+SA	421,11	423,94	1,46	904.320
+VNS	421,59	423,61	1,36	980.327
+GRASP	420,95	423,49	1,50	948.643
+LB	421,07	423,27	1,26	918.496
SA	420,31	423,17	0,96	5.530.018
GRASP	420,41	423,02	1,00	1.347.862
Puro	421,11	423,10	0,93	250.791
+SA	421,56	423,20	1,06	274.690
+VNS	420,41	422,82	1,07	267.978
+GRASP	421,61	422,77	0,94	316.189
+LB	421,23	423,23	1,00	238.214
LB	421,18	423,22	0,94	1.007.204
Puro	421,78	423,12	0,92	174.462
+SA	421,35	423,29	0,86	192.431
+VNS	421,68	423,33	1,03	203.476
+GRASP	421,18	423,17	1,02	265.030
+LB	421,84	423,21	0,93	171.805
SA	420,31	423,27	1,07	1.091.304
Puro	421,63	423,32	0,87	218.415
+SA	421,05	423,40	1,29	175.429
+VNS	421,02	423,16	1,02	250.667
+GRASP	421,63	423,36	1,13	259.861
+LB	420,31	423,13	1,02	186.932
Tradicional	421,09	423,13	0,92	982.267
Puro	421,21	423,33	0,96	166.617
+SA	421,53	423,13	0,91	192.977
+VNS	421,09	423,01	0,99	210.524
+GRASP	421,32	423,09	0,90	233.625

+LB	421,60	423,09	0,86	178.524
VNS	421,08	423,19	0,87	1.101.381
Puro	421,52	423,33	0,85	189.915
+SA	421,28	423,13	0,86	215.227
+VNS	421,08	423,00	1,01	225.808
+GRASP	421,75	423,20	0,71	278.203
+LB	421,27	423,30	0,90	192.228
VNS	420,94	424,67	1,81	10.969.650
GRASP	421,11	424,61	1,75	2.394.557
Puro	421,68	425,17	2,17	464.377
+SA	421,11	424,07	1,46	405.483
+VNS	422,06	424,51	1,55	489.617
+GRASP	421,58	424,52	1,42	560.415
+LB	421,89	424,79	1,98	474.665
LB	421,11	424,76	1,86	2.056.835
Puro	421,17	424,63	1,74	358.232
+SA	421,75	424,35	1,44	387.381
+VNS	422,34	425,00	2,03	404.948
+GRASP	421,99	424,59	1,89	499.215
+LB	421,11	425,25	2,09	407.059
SA	421,75	424,39	1,53	2.171.510
Puro	421,82	424,17	1,67	434.311
+SA	421,75	424,36	1,48	425.309
+VNS	421,84	424,39	1,69	464.466
+GRASP	422,56	424,37	1,37	457.589
+LB	422,33	424,65	1,47	389.835
Tradicional	420,94	424,71	1,89	2.233.881
Puro	421,75	424,46	1,97	381.439
+SA	420,94	424,80	1,67	427.687
+VNS	421,84	424,36	1,76	428.009
+GRASP	422,43	424,78	1,78	580.544
+LB	421,11	425,15	2,26	416.202
VNS	421,24	424,86	1,99	2.112.867
Puro	421,91	425,01	1,91	414.512
+SA	421,62	424,32	1,63	463.214
+VNS	421,24	425,57	2,21	401.261
+GRASP	421,42	424,35	2,07	464.507
+LB	421,70	425,04	1,94	369.373