

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU
FACULDADE DE CIÊNCIAS INTEGRADAS DO PONTAL – FACIP
GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

JANAINA MARIA DA SILVA

**UMA META-HEURÍSTICA BASEADA EM BUSCA EM VIZINHANÇA VARIÁVEL
PARA O PROBLEMA DE *JOB SHOP***

**ITUIUTABA
JUNHO DE 2016**

JANAINA MARIA DA SILVA

**UMA META-HEURÍSTICA BASEADA EM BUSCA EM VIZINHANÇA VARIÁVEL
PARA O PROBLEMA DE *JOB SHOP***

Trabalho de conclusão de curso exigido como requisito obrigatório para a obtenção do título de bacharel em Engenharia de Produção da Universidade Federal de Uberlândia.

Área de Concentração: Pesquisa Operacional.

Orientador: Prof. Dr. Jorge von Atzingen dos Reis.

**ITUIUTABA
JUNHO DE 2016**

JANAINA MARIA DA SILVA

Uma meta-heurística baseada em busca em vizinhança variável para o problema de *Job Shop*

Trabalho de conclusão de curso aprovado para obtenção do título de bacharel em Engenharia de Produção da Universidade Federal de Uberlândia pela banca examinadora formada por:

Banca de Avaliação:

Prof. Dr. Jorge von Atzingen dos Reis - UFU

Orientador

Prof. Dr. Antonio Álvaro de Assis Moura - UFU

Membro

Prof. Dr. Fernando de Araujo – UFU

Membro

Ituiutaba, 21 de Junho de 2016.

DEDICATÓRIA

Dedico este trabalho a todos os meus familiares, namorado e amigos que de todas as formas me ajudaram a chegar até aqui.

Janaina Maria da Silva.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus que me deu forças para seguir meu caminho até aqui, ao meu orientador Jorge Reis que me auxiliou com muita dedicação do começo ao fim desta jornada e à UFU e PROPP/CNPq pelo financiamento da bolsa.

RESUMO

O planejamento e controle da produção (PCP) em uma empresa tem como objetivo reduzir os custos relacionados a produção. Um das tarefas do PCP é determinar a melhor sequência de produção, problema conhecido na literatura como *Job Shop Problem* (JSP). A determinação do sequenciamento é uma tarefa complexa devido às diversas restrições que necessitam ser consideradas. Este trabalho tem como objetivo principal criar uma sequência de produção para uma linha de produção de modo a maximizar o lucro. A avaliação dos resultados é dada pela comparação entre os valores das soluções iniciais e finais após a aplicação da meta-heurística VNS - *Variable Neighborhood Search*, que utiliza estruturas de vizinhança para buscar soluções otimizadas. Os dados utilizados foram gerados aleatoriamente baseados nos dados da literatura.

Palavras-chave: VNS. meta-heurística. Job Shop Problem.

ABSTRACT

The production planning and control (PPC) in a company aims to reduce costs related to production. One of the tasks of the PPC department is to determine the best production sequence, this problem is known in the literature as Job Shop Problem (JSP). Determining the best sequence is a complex task due to several restrictions that need to be considered. This paper aims to create a production sequence for a production line in order to maximize the profit. The evaluation of the results is given by comparing the values of the initial and final solutions after the application of VNS metaheuristics - Variable Neighborhood Search, which uses neighborhood structures to find optimal solutions. The data were randomly generated based on literature.

Keywords: VNS. metaheuristics. Job Shop Problem.

LISTA DE ILUSTRAÇÕES

Figura 1 - Programação da Produção.....	3
Figura 2 - Modelo Matemático do <i>Job Shop Problem</i>	5
Figura 3 - Pseudocódigo Meta-heurística Busca Tabu.....	9
Figura 4 - Pseudocódigo da Meta-heurística ILS.....	10
Figura 5 - Ilustração da Meta-heurística ILS.....	11
Figura 6 - Ilustração da Meta-heurística VND.....	12
Figura 7 - Pseudocódigo da Meta-heurística VNS.....	14
Figura 8 - Ilustração da Meta-heurística VND.....	15
Figura 9 - Pseudocódigo da Heurística Construtiva.....	19
Figura 10 - Ilustração das Estruturas de Vizinhaça.....	20
Figura 11 - Pseudocódigo da Função Objetivo.....	21
Figura 12 - Ilustração do funcionamento do <i>Quicksort</i>	23

LISTA DE TABELAS

Tabela 1 - Valores dos Benefícios para Todos os Produtos.....	17
Tabela 2 - Estruturas de Vizinhança.....	21
Tabela 3 - Resultados para os Produtos da Cor Preta.....	24
Tabela 4 - Resultados para os Produtos da Cor Laranja.....	24
Tabela 5 - Resultados para os Produtos da Cor Verde.....	25
Tabela 6 - Resultados para os Produtos da Cor Vermelha.....	26
Tabela 7 - Resultados para os Produtos da Cor Roxa.....	26

SUMÁRIO

1. INTRODUÇÃO.....	1
2. REFERENCIAL TEÓRICO.....	2
2.1. Planejamento e Controle da Produção.....	2
2.2. <i>Job Shop Problem</i>	4
2.2.1. JSP e aplicações.....	6
2.3. Busca Tabu.....	8
2.4. ILS.....	9
2.5. VND.....	11
2.6. VNS.....	12
2.6.1. Aplicações do VNS.....	15
3. METODOLOGIA.....	16
3.1. Dados de Entrada.....	16
3.2. Heurística Construtiva.....	18
3.3. Estruturas de Vizinhança.....	19
3.4. Função Objetivo.....	21
3.5. <i>Quicksort</i>	22
4. RESULTADOS OBTIDOS.....	24
5. CONSIDERAÇÕES FINAIS.....	27
REFERÊNCIAS.....	29

2. INTRODUÇÃO

Com o objetivo de aumentar o rendimento e garantir a posição no mercado as empresas buscam cada vez mais alternativas capazes de melhorarem seus sistemas produtivos. As companhias investem nas suas áreas mais defasadas de modo a cortar gastos e obter um processo mais eficiente.

A área de gestão e planejamento de uma empresa é responsável por identificar quais os pontos do processo que necessitam ser melhorados. Os problemas de uma fábrica não são exclusivos dos processamentos e do chão de fábrica, muitas vezes o problema acontece antes destas etapas, contudo as consequências só são observadas ao final do processo produtivo.

São vários os problemas que podem ocorrer em um processo produtivo, alguns são difíceis de serem previstos, entretanto outros são relacionados ao planejamento e controle da produção (PCP) e devem ser evitados. O PCP é a área da empresa que desenvolve o processo do início ao fim, acompanhar e controlar todas as etapas é necessário para verificar se o mesmo caminha conforme o planejado.

Alguns dos problemas de PCP são tão comuns que há estudos desenvolvidos somente para auxiliar a identificação e solução no mesmo. Um desses casos é o problema de sequenciamento, conhecido na literatura como *Job Shop Problem*.

A Pesquisa Operacional (PO) é formada por um aglomerado de técnicas de solução para diversos tipos de problemas, e a determinação da solução é baseada na complexidade do problema (TAHA, 2008).

Algumas das técnicas da PO são a programação linear, a programação inteira, a programação dinâmica, otimização em redes, a programação não linear e as heurísticas e meta-heurísticas. As heurísticas e meta-heurísticas diferentemente das outras técnicas podem não apresentar uma solução ótima, mas sim uma solução boa ou melhorada, dado que elas são utilizadas na solução de problemas que possuem modelos matemáticos muito complexos (TAHA, 2008).

Há diversos tipos de meta-heurísticas disponíveis atualmente, e determinar qual utilizar depende não somente do problema, dado que uma mesma meta-heurística é capaz de solucionar diferentes tipos de problemas, assim como um mesmo problema pode ser solucionado por mais de um tipo de meta-heurística. É necessário avaliar então qual metodologia apresenta uma qualidade de resultado mais próxima do esperado.

Neste trabalho a meta-heurística discutida é a *Variable Neighborhood Search* (VNS ou Busca em Vizinhança Variável) que é um método de busca de melhores soluções a partir da vizinhança, age através de trocas de estruturas afim de encontrar novas soluções com resultados superiores às anteriores, trata-se de uma meta-heurística já otimizada de outras, o que classifica seus resultados como mais refinados.

O objetivo deste trabalho é apresentar como a meta-heurística *Variable Neighborhood Search* pode melhorar a solução de um problema de *Job Shop Problem*. Para tal, será solucionado um problema teórico de sequenciamento, e os resultados serão avaliados de modo a identificar as melhorias.

3. REFERENCIAL TEÓRICO

Nesta seção do trabalho estão descritos os principais fundamentos para a compreensão do trabalho, assim como as referências bases para o desenvolvimento do mesmo, que justificam a escolha do tema.

3.2. Planejamento e Controle da Produção

Tubino (2009) define como Planejamento e Controle da Produção a atividade responsável por organizar os dados e as tomadas de decisões considerando todos os fatores do processo produtivo. O PCP precisa definir a melhor maneira de coordenar e aplicar os recursos da empresa de modo a atender os planos estabelecidos nos níveis estratégico, tático e operacional.

Vollmann et. al. (2006) classifica as atividades do PCP como tarefas de longo, médio e curto prazo. As tarefas de longo prazo são as relacionadas às tomadas de decisões sobre demandas e capacidades futuras. As de médio prazo fornece informações sobre a combinação entre suprimentos, volumes de demanda e *mix* de produtos. E as de curto prazo indicam as programações de recursos, ou seja, como tratar dos materiais, equipamentos, tempo e instalações.

Slack et. al. (2009) também analisa as tarefas do PCP como longo, médio e curto prazo. No longo prazo concentram-se mais as tarefas de planejamento, realizando previsão de

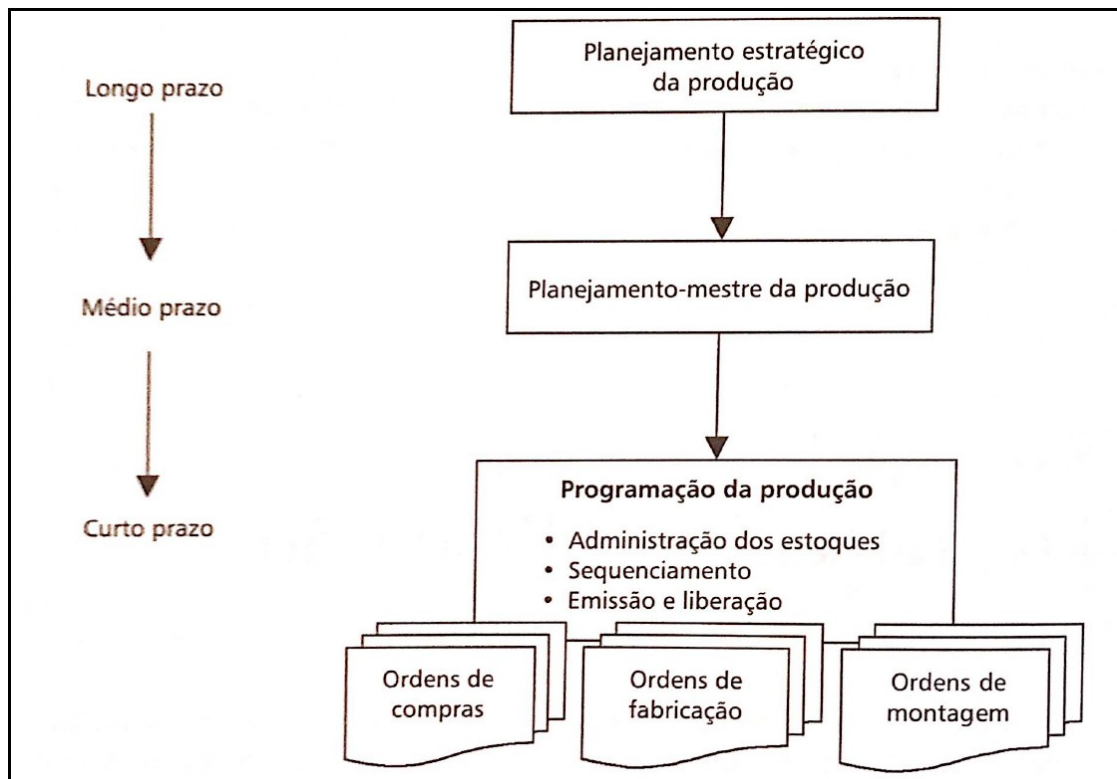
demanda e estabelecendo objetivo; no médio prazo, as tarefas de planejamento já são mais detalhadas; e por fim, no curto prazo determinam-se os recursos e considera-se a demanda real.

Para Tubino (2009) as atividades mais importantes do PCP são a previsão da demanda, o planejamento estratégico, o planejamento-mestre, a programação da produção e o acompanhamento e controle da produção.

A programação da produção é dividida em três grupos: a administração de estoques, o sequenciamento e a emissão e liberação de ordens. O foco deste trabalho é a fase de sequenciamento, na qual gera-se um programa de produção que combina a fabricação e montagem dos produtos com qualidade e baixo custo (TUBINO, 2009).

A Figura 1 representa as atividades, destacando as atividades da programação da produção.

Figura 1 - Programação da Produção



Fonte: Tubino (2009)

Para Lustosa et. al. (2008) a programação da produção é mais detalhada e dividida em sete categorias, sendo elas: Designação, Sequenciamento, Programação, Despacho, Controle, Apressamento e Carregamento de Oficinas. A partir dessas categorias é possível identificar onde a tarefa será executada, a sequência da tarefa, quando deve ser iniciada e terminada, para

quem e quando será enviada, como fazer o acompanhamento, como acelerar as ordens de modo a atender às prioridades e por fim os roteiros e programação da empresa.

De acordo com Tubino (2009) as principais informações que devem conter nas ordens de sequenciamento são em relação as listas de ordens de fabricação e em relação aos recursos disponíveis. Isso significa considerar tanto as ordens de planejamento, como as características do produto e dos clientes, estoque, tempos de processamento, como também taxa de produção, capacidade e tempo de *setup* dos recursos.

Para definir as ordens de sequenciamento é possível basear-se em algumas regras de prioridade. As mais utilizadas segundo Tubino (2009) e Lustosa et. al. (2008) são: PEPS – Primeiro que entra, primeiro que sai; MTP - Menor tempo de processamento; e MDE – Menor data de entrega. Estas ordens de prioridade não são somente regras prontas e definidas, elas podem ser criadas a partir de índices importante para a empresa e serem usadas como índices críticos.

Bremer & Lenza (2000) destaca o uso de sistemas de programação na área de programação da produção. Segundo eles, esses sistemas são responsáveis pelo gerenciamento das ordens de produção e requisições de compras definidas no planejamento. Os sistemas de produção são sistemas computacionais que utilizam algoritmos matemáticos a fim de diminuir os tempos de transporte, de fila e de processamento.

3.3. *Job Shop Problem*

Durante o planejamento dos processos produtivos um dos principais problemas é a alocação dos recursos para assegurar a data de conclusão dos projetos. Esse é um problema de sequenciamento da produção, conhecido na literatura como *Job Shop* ou *Job Scheduling*. É parte importante do Planejamento e Controle da Produção, devido às empresas necessitarem de um plano eficiente afim de elevar sua produtividade utilizando recursos restritos, como máquina e rotinas de trabalho. Muitas empresas aplicam multas e restrições em caso de atrasos, logo, o planejamento eficiente é fator determinante para uma empresa (TUBINO, 2009).

O *Job Shop Problem* ou *Job Scheduling Problem* (JSP) é um problema de sequenciamento no qual há uma quantidade n de tarefas que devem ser processadas em um número m de máquinas. Em cada tarefa i existe uma sequência de m de operações (O_{i1} , O_{i2} , O_{i3} , ..., O_{im}). O princípio do JPS é que cada tarefa deve ser processada sem interrupções, nas

máquinas previamente definidas e por um tempo pré-determinado (GAO, SUN & GEN, 2007).

As tarefas de um processo produtivo são independentes entre si, o que significa que cada uma possui seu próprio fluxo nas máquinas. Comum a todos os processamentos é que cada máquina pode processar uma tarefa por vez e cada tarefa só pode ser processada por uma máquina por vez. O tempo de processamento de todas as tarefas é conhecido como *makespan* (C_{max}) e um dos objetivos de solucionar um JSP é tornar este tempo o menor possível (JAIN & MEERAN, 1998).

Figura 2- Modelo Matemático do *Job Shop Problem*

$$\begin{aligned}
 & \min \sum_{i=1}^n C_{im} \\
 & C_{i,maq_1} \geq p_{i,maq_1} \quad \forall i=1,\dots,n \\
 & C_{i,maq_{k+1}} \geq C_{i,maq_k} + p_{i,maq_{k+1}} \quad \forall i=1,\dots,n \quad \forall k=1,\dots,m-1 \\
 & C_{jk} \geq C_{ik} + p_{jk} - M(1-x_{ijk}) \quad i \neq j \quad \forall i,j=1,\dots,n \quad \forall k=1,\dots,m \\
 & C_{ik} \geq C_{jk} + p_{ik} - M x_{ijk} \quad i \neq j \quad \forall i,j=1,\dots,n \quad \forall k=1,\dots,m \\
 & C_{ik} \geq 0 \quad \forall i=1,\dots,n \quad \forall k=1,\dots,m \\
 & x_{ijk} \in [0,1] \quad \forall i,j=1,\dots,n \quad \forall k=1,\dots,m
 \end{aligned}$$

Fonte: Adaptado de Arenales et al. (2007)

A Figura 2 apresenta a formulação clássica para um modelo matemático do JSP (*Job Shop Problem*). Esta formulação clássica do *Job Shop* visa minimizar o *makespan*. Em pesquisa operacional, o *makespan* de um projeto é o tempo total decorrido entre o início e o término, ou em outras palavras, é a diferença de tempo entre o início e o final de uma sequência de *jobs* ou tarefas. A função objetivo é minimizar o somatório de todos os C_{im} , onde C_{im} representa o instante de término da tarefa i na máquina m .

No modelo, as variáveis de decisão são C_{ik} e x_{ijk} . C_{ik} representa o horário de término da tarefa i na máquina k , e x_{ijk} assume valores binários para indicar se a tarefa i precede a tarefa j na máquina k , caso preceda assume valor 1, se não, assume 0. M , maq_i , p_{ik} são os parâmetros do modelo, M é um número suficientemente grande em relação aos dados disponíveis, maq_i representa a máquina que processa a i -ésima tarefa e p_{ik} é o tempo de processamento da tarefa i na máquina k .

A primeira restrição garante que a tarefa i termina somente quando o tempo de processamento é concluído, ou seja, quando o valor de $C_{i,maq}$ é igual ou superior ao tempo de processamento da tarefa i na máquina.

A segunda restrição garante que a tarefa $k+1$ só é concluída depois do término da tarefa k e depois do tempo de processamento da tarefa $k+1$, o que significa que o tempo de término da tarefa $k+1$ deve ser igual ou superior à soma do tempo de processamento da tarefa $k+1$ mais o tempo de término da tarefa k .

A terceira e quarta indicam se na máquina k a tarefa i precede a tarefa j ou vice-versa, de acordo com o valor de x_{ijk} . O x_{ijk} está sendo utilizado no modelo como fator multiplicador de M , na forma direta ou através da expressão $(1 - x_{ijk})$, o que influencia diretamente o resultado do lado direito da restrição. Se x_{ijk} for 1, o fator multiplicador de M resultará em 0 na terceira restrição, o que implica que a quarta restrição é desativada, caso x_{ijk} seja 0, o fator multiplicador de M torna-se nulo para a quarta restrição, implicando assim na desativação da terceira restrição.

Um algoritmo é uma alternativa de solução desenvolvida a fim de auxiliar os tomadores de decisões a identificar a melhor solução para um problema. Algoritmos são capazes de encontrar a solução ótima em tempo razoável para problemas pequenos, no entanto à medida que o tamanho do problema cresce, como o caso do JSP, a rápida obtenção de uma solução ótima é extremamente difícil em um tempo computacional aceitável. Por se tratar de um problema de otimização combinatorial o JSP tem seu nível classificado como *NP-hard* (*não polinomial difícil*), o que significa que não há um algoritmo capaz de encontrar a melhor solução em tempo polinomial (MELO; RONCONI, 2012).

Para problemas, que assim como o JSP, não são solucionados por algoritmos, existem diversos tipos de meta-heurísticas. Algumas delas podem utilizar o mesmo princípio de solução, como no caso das meta-heurísticas baseadas em busca - Busca Tabu (*Tabu Search*), *Iterated Local Search* (ILS), *Variable Neighborhood Descent* (VND) e VNS que serão apresentadas nas seções 2.3, 2.4 e 2.5.

3.3.1. JSP e aplicações

Nesta subseção, encontram-se alguns trabalhos realizados com o foco de representar a importância de solucionar o JSP e quais foram as técnicas utilizadas.

Muller & Gómez (2006) realizaram um estudo para solucionar o problema do *Job Shop* em um sistema de manufatura flexível utilizando a meta-heurística Busca Tabu. O modelo consistia basicamente de uma estrutura de dados, um mecanismo de aceitação ou rejeição da nova configuração e de um mecanismo responsável pela intensificação e diversificação das configurações. O resultado alcançado neste trabalho mostrou que o modelo

é uma alternativa para o JSP, contudo há a necessidade de melhorar a relação de datas de entrega e produtividade para o caso avaliado.

Melo & Ronconi (2012) desenvolveram um trabalho afim de solucionar o JSP e como método de solução utilizaram uma Heurística Construtiva, fundamentada em regras de prioridades pré-determinadas e que gerasse uma solução de qualidade dentro de um espaço de tempo razoável. O resultado encontrado se mostrou satisfatório por atingir todas as regras implementadas e por gerar soluções factíveis.

Silva et al. (2012) elaborou um trabalho com o objetivo de identificar as melhores regras de sequenciamento de produção em um ambiente de *Job Shop* e *Flow Shop*. Para tal, foi utilizado uma simulação computacional e indicadores de desempenho. Os indicadores utilizados foram em relação ao tempo médio de fluxo, atraso total e número de ordens atrasadas. Os resultados mostraram que não há uma regra de sequenciamento considerada melhor e mais indicada para a maioria dos casos, dado que eles variam de acordo com os indicadores e ambiente, entretanto as regras que mais se repetiram foram a de menor data de entrega (EDD) e a de menor tempo de processamento iminente (SIPT).

Toso e Morabito (2005) realizaram um estudo em torno do problema de sequenciamento em uma fábrica de rações. O objetivo do trabalho era determinar a quantidade a ser produzida de cada tipo de produto considerando a sequência de produção, a demanda, a minimização dos custos e estoques. Para solucionar o problema foi desenvolvido um modelo matemático e resolvido no *software* GAMS/CPLEX e o resultado encontrado apresentou uma sequência de produção melhor do que a praticada pela empresa e ressaltou a importância da comunicação entre o planejamento e controle da produção com o chão de fábrica para a identificação das restrições e alterações na produção. O trabalho mostrou também que para problemas de sequenciamento o modelo matemático escolhido não foi o mais eficiente devido à demora na obtenção da solução computacional e sugeriu o uso de outros métodos, como heurísticas construtivas, buscas locais e meta-heurísticas.

Figueiredo et al. (2013) desenvolveu um trabalho para avaliar o problema de sequenciamento em uma linha de usinagem industrial, para tal utilizou a meta-heurística *Simulated Annealing* e também o método exato, para verificar a qualidade da solução obtida pela primeira. A diferença entre o método exato e o *Simulated Annealing* é que o primeiro apresenta a melhor solução e o segundo uma solução ótima, que pode não ser a melhor, porém com menor tempo computacional. Para o caso aplicado tanto o método exato quanto a meta-heurística encontraram soluções com o mesmo *makespan* para dois casos distintos, ambos envolvendo cinco máquinas e 6 tarefas, porém com demandas diferentes. A grande diferença

entre os métodos é que o tempo computacional do método exato foi cinco vezes maior e isso piora para casos envolvendo maior número de soluções.

3.4. Busca Tabu

A meta-heurística Busca Tabu (*Tabu Search* - TS) foi desenvolvida por Glover em 1986 e recebeu este nome por ser uma metodologia que permite que a busca local evite ótimos locais já avaliados, os chamados tabus. Para evitar melhorias que já foram realizadas previamente, esta meta-heurística utiliza as chamadas listas tabus (*tabu lists*), uma espécie de memória que armazena um histórico de busca (GLOVER & KOCHENBERG, 2003).

As listas são preenchidas com os resultados dos movimentos realizados recentemente, e permanecem nas mesmas durante um período de tempo determinado. O fato desses movimentos estarem registrados nas listas impede que movimentos iguais sejam realizados, evitando assim a ocorrência de ciclos (REIS & CUNHA, 2010).

A Figura 3 apresenta o pseudocódigo da meta-heurística Busca TABU (TS – TABU Search) para um problema de minimização. A partir de uma solução inicial S previamente determinada por uma heurística construtiva gulosa, é realizada uma busca local em toda a vizinhança da solução S . Esta busca local visa determinar qual a melhor solução vizinha S' da solução inicial S . A vizinhança de uma solução é definida como todas as soluções que podem ser obtidas a partir de um movimento previsto na estrutura de vizinhança determinada. A solução vizinha S' é armazenada na lista tabu.

O valor da função objetivo (FO) da solução vizinha S' é comparado ao valor da FO atual, referente à solução S , caso a FO da solução vizinha seja menor que o valor da FO atual, a solução S' torna-se a nova solução, denominada solução corrente.

Depois de atualizada a lista tabu e a solução corrente, um novo processo de busca é iniciado, como descrito anteriormente, exceto que a partir de então sempre haverá uma verificação da lista tabu antes da atualização da solução, de modo a evitar soluções já analisadas, ou seja, caso uma solução encontre-se na lista significa que a mesma já foi verificada, independentemente de ter sido ou não adotada como solução corrente já teve seu valor de FO considerado.

O critério de parada da meta-heurística Busca Tabu, pode ser definido como um valor máximo de iterações, ou limite mínimo entre os resultados das funções objetivos da nova solução e da solução corrente, evitando assim que o processo se estenda por muito tempo e a melhora seja pouco significativa.

Figura 3 - Pseudocódigo Meta-heurística Busca Tabu

```

Procedimento Busca Tabu
1.  $s^* \leftarrow s$ ;
2.  $Iter \leftarrow 0$ ;
3.  $MelhorIter \leftarrow 0$ ;
4. Esvazie a lista tabu;
5. Inicialize a função de aspiração;
6. enquanto ( $FO(s) > FO\_min$  e  $Iter - MelhorIter \leq max\_Iter$ ) faça
7.      $Iter \leftarrow Iter + 1$ ;
8.     Encontre o melhor elemento da vizinhança cujo movimento
        não esteja na lista tabu ou atenda o critério de aspiração;
9.     Atualize a lista tabu;
10.     $s \leftarrow s'$ ;
11.    se ( $FO(s) < FO(s^*)$ ) então
12.         $s^* \leftarrow s$ ;
13.         $MelhorIter \leftarrow Iter$ ;
14.    fim-se;
15.    Atualize a função de aspiração;
16. fim-enquanto;
17.  $s \leftarrow s^*$ ;
18. Retorne  $s$ ;

Fim

```

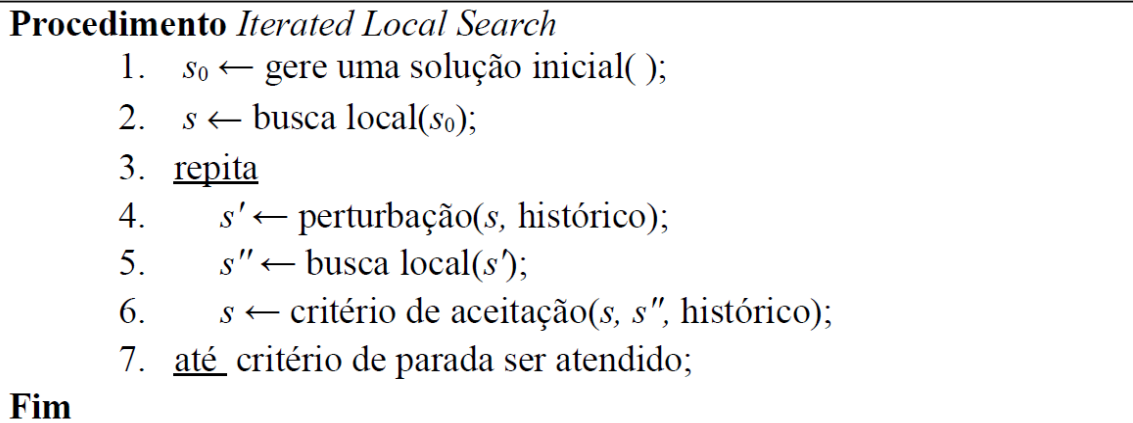
Fonte: Reis & Cunha (2010)

3.5. ILS

O VNS foi desenvolvido a partir dos princípios de outras meta-heurísticas semelhantes, como o ILS (*Iterated Local Search*), procedimento para otimização de soluções baseados na busca local. A solução local sofre perturbações afim de encontrar uma nova solução melhor que a inicial (GLOVER & KOCHENBERG, 2003), ou seja, há uma variação dos pontos para a nova busca local.

Na Figura 4, o pseudocódigo, retirado de Glover & Kochenberg (2003), representa o fundamento da meta-heurística ILS.

Figura 4 - Pseudocódigo da Meta-heurística ILS



Fonte: Reis & Cunha (2010)

O procedimento da meta-heurística ILS (*Iterated Local Search*) necessita de uma solução inicial, assim como na Busca Tabu esta pode ser gerada a partir de uma heurística construtiva gulosa ou alguma outra heurística equivalente. Definida a solução inicial S é realizada uma busca local na sua vizinhança visando encontrar uma solução S' que melhore o valor da função objetivo da solução inicial S . Para este caso a vizinhança também é definida por toda e qualquer solução que possa ser atingida através de um único movimento de vizinhança de acordo com a estrutura de vizinhança proposta.

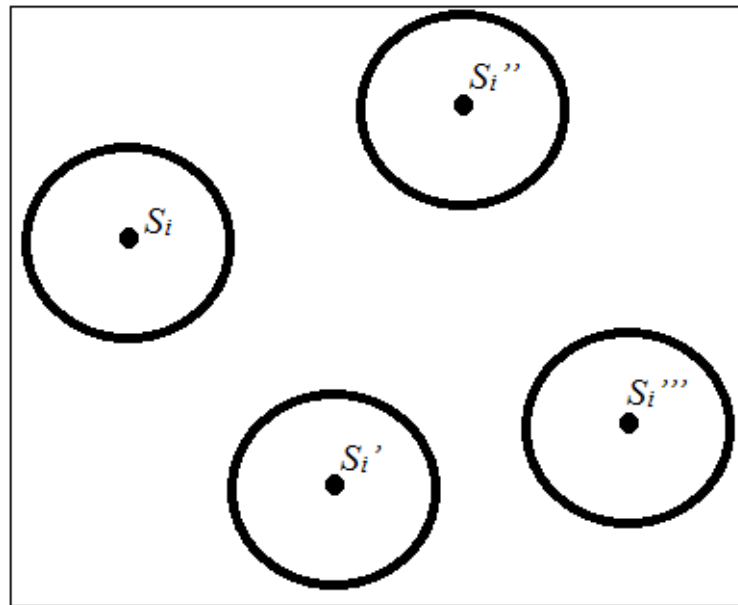
Após encontrada a solução S' , seu valor de função objetivo deve ser comparado com o valor de função objetivo da solução S , que até o momento é valor corrente. Caso a FO de S' seja melhor a FO de S , ou seja, caso seja inferior para um problema de minimização ou superior para um problema de maximização, a S' assume então o papel de solução corrente.

Antes de uma nova busca ser realizada, ocorre uma perturbação na vizinhança da solução S que faz com que o ambiente no qual a solução se encontra e que estão sendo realizadas as buscas se modifique. Depois da perturbação, uma nova busca local é realizada na nova vizinhança e novamente há uma comparação das funções objetivos de modo a encontrar a melhor solução.

O critério de parada, assim como na busca tabu, pode ser definido como uma quantidade máxima de iterações, ou um intervalo mínimo entre as funções objetivo das soluções, nova e corrente.

A Figura 5 ilustra o funcionamento da meta-heurística ILS para três iterações, nas quais S_i é a primeira solução, S_i' a solução após a primeira iteração, e assim sucessivamente até S_i''' .

Figura 5 - Ilustração da Meta-heurística ILS



Fonte: Autoria Própria

3.6. VND

Outra meta-heurística que fundamentou o desenvolvimento do VNS é o VND (*Variable Neighborhood Descent*), que tem como propósito também a busca local, porém auxiliado por métodos determinísticos (MLADENOVIC & HANSEN, 1997). Nesta meta-heurística, a cada iteração ocorre uma ampliação do espaço da busca local, afim de encontrar uma melhor solução.

Assim como nas outras meta-heurísticas citadas, o procedimento da meta-heurística VND (*Variable Neighborhood Descent*) necessita de uma solução inicial, que também pode ser obtida através de uma heurística construtiva ou alguma outra heurística equivalente.

Definida a solução inicial S é realizada uma busca local na sua vizinhança a fim de determinar uma nova solução S' que possa otimizar o valor da função objetivo, ou seja, que possa aumentá-la no caso de problemas de maximização ou diminuí-la no caso de problemas de minimização.

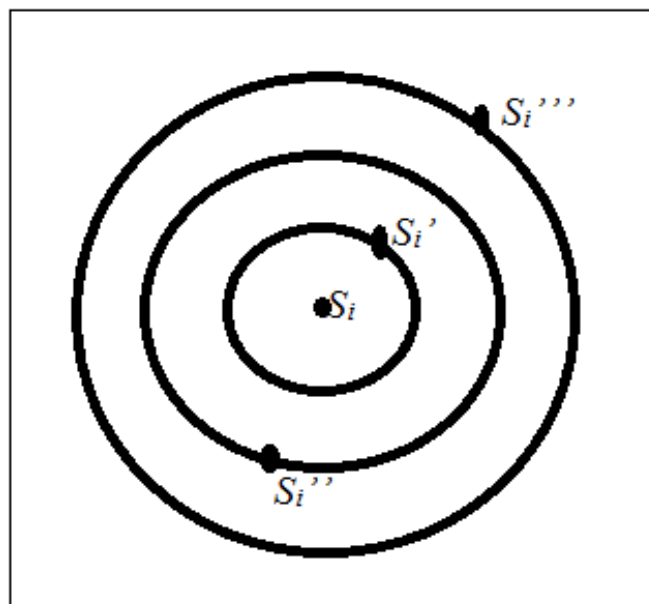
No procedimento do VND, ocorre uma ampliação na estrutura de vizinhança a fim de aumentar o espaço de buscas. Depois que a solução S' for encontrada, a partir da solução S , e a primeira ampliação na vizinhança for realizada o processo de busca ocorre novamente.

A busca realizada em um espaço maior tem maior chance de apresentar um melhor resultado, porém também aumenta o gasto computacional, o que faz com que seja necessário utilizar um eficiente critério de parada.

Um exemplo de como seria a ampliação da vizinhança é imaginar um espaço circular, que seria a vizinhança inicial, onde se encontram as soluções, o processo consistiria em aumentar o número de círculos mantendo o mesmo centro, e variando o diâmetro dos mesmos, o que implicaria num maior espaço de soluções, no qual cada círculo é uma nova vizinhança, como ilustrado na Figura 6.

A Figura 6 apresenta a variação na solução até a terceira iteração, cada solução é representada por S . A primeira solução é S_i e a última S_i''' .

Figura 6 - Ilustração da Meta-heurística VND



Fonte: Autoria Própria

3.7. VNS

Mladenovic e Hansen (1997), propuseram a meta-heurística VNS a fim de demonstrar que era possível um método de solução simples e eficiente através de mudanças sistemáticas de vizinhança combinando-as com uma busca local. O nome *Variable Neighborhood Search* explica-se pelo fato de, ao contrário de outras buscas locais que seguem uma trajetória única, este método avalia uma vizinhança que pode ser maior ou menor dependendo da precisão a ser almejada da solução.

O princípio do *VNS* é simples: criar conjuntos de soluções e combiná-las entre si para chegar a melhor solução possível. Ele age explorando as soluções através de trocas sistemáticas de estruturas de vizinhança, e focaliza a busca em torno de uma nova solução, a

qual é alcançada somente quando um movimento de melhora é identificado (MLADENOVIC & HANSEN, 1997).

As estruturas de vizinhança de uma solução são as modificações que podem ser feitas na solução atual, de tal forma a gerar uma nova solução distinta e melhorada. Para se determinar se a solução vizinha é uma melhora de uma anterior é utilizada a função de avaliação, que tem como objetivo quantificar cada solução e selecionar as de valores superiores (REIS & CUNHA, 2010).

A princípio o método VNS necessita de uma solução inicial e uma vizinhança $N^k(s)$, na qual k é um número natural maior que 1 e menor que o valor determinado como critério de parada, que pode ser algo como o tempo máximo de processamento ou o número máximo de iterações, e s é o conjunto de soluções (MLADENOVIC & HANSEN, 1997).

Um vizinho s' é submetido a uma busca local o que resulta em uma solução s'' , esta solução é então comparada com a atual solução, chamada de solução corrente. Dentre as duas soluções a menor é descartada e a maior assume a posição de solução corrente (MLADENOVIC & HANSEN, 1997). O pseudocódigo que representa esse processo encontra-se na Figura 7.

A partir do pseudocódigo é possível descrever o processo da meta-heurística VNS. Assim como nas outras meta-heurísticas citadas, parte-se de uma solução inicial, que também pode ser obtida por uma heurística, sendo ela construtiva ou não.

Para o caso do VNS é necessário que as estruturas de vizinhança também sejam desenvolvidas antes de iniciar o procedimento. As estruturas são criadas pelo responsável pela elaboração do procedimento, e como citado, são trocas possíveis de serem realizadas pelo modelo, de forma a melhorar o resultado.

Com a solução inicial e as estruturas de vizinhança inicia-se o procedimento. As estruturas devem receber um código de identificação, a forma mais simples é numerar cada estrutura. O procedimento ocorre enquanto houverem estruturas a serem analisadas.

Dentro de um grupo de estrutura de vizinhança, é possível que existam diversas trocas, ou seja, para cada iteração, somente uma troca será feita, e de acordo com os resultados, as outras podem ser também analisadas ou não. É escolhida uma estrutura de vizinhança aleatoriamente através da busca local dentro na vizinhança. A estrutura escolhida assume então o papel de segunda solução.

Com a solução corrente, que no primeiro caso é a solução inicial, e a segunda solução, que foi obtida através da busca local, faz-se a comparação dos resultados de suas funções

objetivos. Para o modelo do pseudocódigo o objetivo é a minimização, então escolhe-se como solução corrente a solução que apresentar a menor função objetivo.

Caso a segunda solução seja a escolhida, o procedimento continuará avaliando os outros itens da estrutura, caso contrário, outra estrutura de vizinhança será escolhida e o processo se repetirá.

O procedimento chega ao fim quando atingir o critério de parada, e este pode ser quando todas as estruturas forem avaliadas, ou para casos no qual haja um número muito elevado de estruturas e deseje-se diminuir o tempo computacional, quando a diferença entre as soluções atinja um limite muito pequeno, o que significa que a melhora não é significativa o bastante para compensar o gasto computacional.

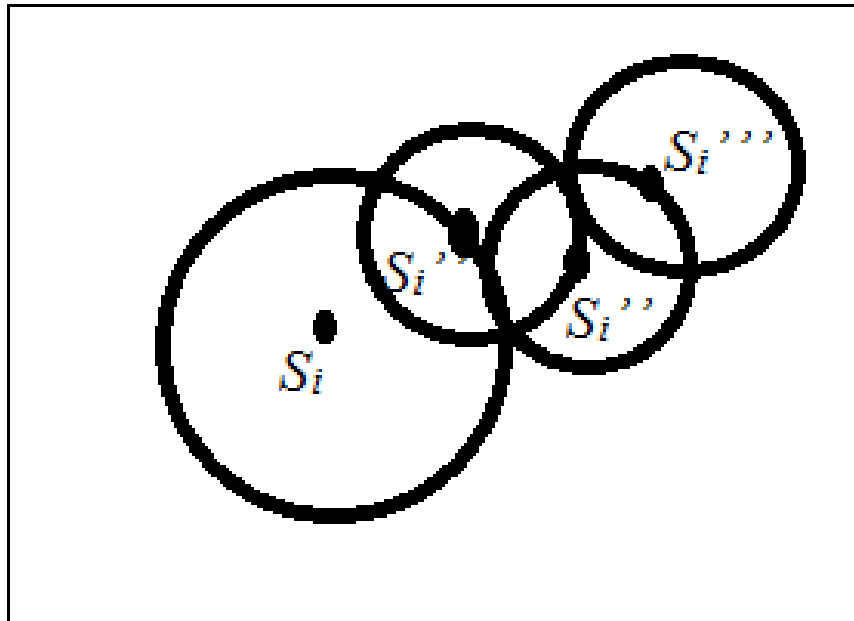
Figura 7 - Pseudocódigo da Meta-heurística VNS

Procedimento <i>Variable Neighborhood Search</i>	
1.	Seja s_0 uma solução inicial e r o número de estruturas de vizinhança;
2.	$s \leftarrow s_0$; {Solução corrente}
3.	<u>Enquanto</u> (Critério de parada não satisfeito) <u>faça</u>
4.	$k \leftarrow 1$; {Tipo de estrutura de vizinhança}
5.	<u>Enquanto</u> ($k \leq r$) <u>faça</u>
6.	Gere um vizinho qualquer $s' \in N^{(k)}(s)$;
7.	$s'' \leftarrow \text{Busca_Local}(s')$;
8.	<u>Se</u> $f(s'') < f(s)$
9.	<u>Então</u> $s \leftarrow s''$;
10.	$k \leftarrow 1$;
11.	<u>Senão</u> $k \leftarrow k + 1$;
12.	<u>Fim-se</u> ;
13.	<u>Fim-enquanto</u> ;
14.	<u>Fim-enquanto</u> ;
15.	Retorne s ;
Fim	

Fonte: Reis & Cunha (2010).

A Figura 8 ilustra como é o resultado da aplicação do VNS e deixa mais claro como esta meta-heurística é resultado das meta-heurísticas ILS e VND, é possível perceber pelo resultado, que é como se essas duas meta-heurísticas tivessem sido aplicadas juntas. Ou seja, há um crescimento no espaço da vizinhança, a partir da solução corrente, como indicado na meta-heurística VND, porém não pela ampliação do diâmetro, mas pela variação da posição no espaço, como o que apresenta a meta-heurística ILS.

Figura 8 - Ilustração da Meta-heurística VND



Fonte: Autoria Própria

3.7.1. Aplicações do VNS

Nesta subseção, estão apresentados sucintamente alguns estudos que utilizaram a meta-heurística VNS, ou adaptações da mesma, para solucionar diversos tipos de problemas de planejamento.

Freitas e Montané (2008) utilizaram uma combinação de meta-heurísticas, VNS-VND e GRASP-VNS, para solucionar um problema de roteamento de veículos com coleta e entrega simultânea. O problema consistia na determinação da melhor rota considerando os custos de transporte e as distâncias percorridas, variando o número de clientes entre 50 e 199. Os autores chegaram à conclusão que a meta-heurística VNS-VND apresenta resultados superiores devido a diversificação causada pelas estruturas de vizinhança.

O trabalho realizado por Paula et al. (2006) avaliou a qualidade da meta-heurística VNS também para a solução do problema de sequenciamento, considerando ainda os tempos de preparação das máquinas paralelas com relação à dependência de sequência. Para comparar e verificar a qualidade da solução a partir do VNS, foram utilizados três algoritmos baseados na abordagem GRASP. O estudo consistia em avaliar 8 conjuntos com diferentes números de tarefas variando a data de entrega. O resultado encontrado comprovou a eficiência da meta-heurística quando comparada aos algoritmos, principalmente para atividades com mais de 60 tarefas.

Outro estudo em que o VNS foi utilizado para solucionar problema de roteirização foi o do Polacek et al. (2004). Neste trabalho a meta-heurística do VNS se mostrou superior à meta-heurística Busca Tabu, uma das mais utilizadas na solução de problemas de roteirização. Tratou-se de um problema de roteirização com múltiplos pontos e com tempo limitado, e segundo os autores o VNS apresentou 10 novas soluções e 6 empates, contra somente 4 novas soluções da Busca Tabu.

O trabalho realizado por García-López et al. (2002) teve como objetivo encontrar a solução para o problema *p-Median* utilizando o VNS. O *p-Median Problem* consiste na localização de plantas industriais, armazéns, centros de distribuição e/ou outros de modo a minimizar os custos de transporte e atender às demandas dos clientes. Neste estudo a meta-heurística também foi considerada uma alternativa de solução satisfatória.

A meta-heurística VNS foi utilizada por Volpato et al. (2013) para determinar uma escala de trabalho de enfermeiros. Optou-se pelo uso da meta-heurística devido à grande quantidade de restrições e preferências que tornam o escalonamento uma atividade complexa. Foi utilizado uma base de dados de 250.000 instâncias e os resultados obtidos pelo VNS foram comparados com resultados tratados na literatura para verificar a eficiência dos mesmos e chegou-se à conclusão que os resultados eram bons e ainda melhores quando as instâncias eram de grande porte.

4. METODOLOGIA

Para a realização do trabalho foi desenvolvido um código em linguagem C++ para encontrar a solução para o problema de sequenciamento. Os dados do problema foram retirados da literatura.

4.2. Dados de Entrada

Os dados de entrada necessários para a realização do código em C++ foram adaptados a partir dos problemas existentes no site da *OR-Library*. Foram eles: capacidade das máquinas, tempos de setup, demandas, custos, margens de lucro, tamanhos e cores dos produtos. Os valores foram dispostos em um arquivo do tipo texto utilizado como fonte de entrada do programa.

O modelo do problema foi retirado da biblioteca citada e os valores foram gerados a partir de um código de geração de números randômicos. O problema consiste na produção de 20 tipos de produtos que diferem entre cores e tamanhos. A sequência fixa de produção inicia-se na máquina i ou na j , segue para máquina i caso tenha sido iniciada na j e vice-versa e então finaliza a montagem na máquina k . O produto só é finalizado caso cumpra toda a sequência de produção. Cada máquina deve ser preparada para atender à cada tipo de produto, variando-se a cor da matéria prima, o controlador de dimensão da máquina e os acessórios compatíveis com os mesmos. Alguns valores gerados randômicamente encontram-se a seguir:

- Capacidade das máquinas em unidades:
 - I – 10300000
 - J – 3545
 - K – 1890

- Tempo de Setup das máquinas em minutos – 68

A Tabela 1 apresenta as margens de lucro de cada produto e os valores são utilizados no cálculo da Função Objetivo.

Tabela 1 - *Valores dos Benefícios para Todos os Produtos*

Tamanho	Cores				
	Preto	Laranja	Verde	Vermelho	Roxo
250 cm ³	0,75	0,57	0,54	0,57	0,54
600 cm ³	0,81	0,65	0,62	0,65	0,62
1000 cm ³	1,27	1,13	0,99	1,13	0,99
2000 cm ³	1,77	1,44	1,32	1,44	1,32

Fonte: Autoria Própria

4.3. Heurística Construtiva

A solução inicial é o ponto de partida para a meta-heurística do VNS, e a mesma foi obtida através de uma heurística construtiva de modo que atendesse as restrições de viabilidade, tal como, o produto só pode ser manufaturado se a máquina tiver capacidade disponível, caso haja embalagem para o produto e se houver demanda para o mesmo. A Figura 9 apresenta o pseudocódigo da heurística construtiva.

O pseudocódigo da solução inicial indica o processo de determinação da solução inicial. No início é atribuído para cada produto o seu benefício, em função da sua demanda e do seu custo, ou seja, para produtos que tenham alta demanda o benefício será maior, e por isso será prioritário em relação aos outros produtos, dado que o custo é fixo para cada produto.

A partir dos valores dos benefícios de cada produto é necessário identificar os que apresentam melhores resultados, então utiliza-se de algoritmo de ordenação para definir a priorização dos produtos. Neste caso o algoritmo utilizado foi o *Quicksort*.

Depois de ordenado os valores, inicia-se a etapa de definição da quantidade a ser produzida de cada produto. A quantidade a ser produzida depende da capacidade da máquina. A capacidade é determinada pela quantidade de matéria prima disponível. Para os produtos com maiores benefícios a máquina era reabastecida, para os de menores a quantidade de matéria prima utilizada era menor.

O critério de parada adotado para esta heurística foi quando a demanda fosse atingida e não houvesse matéria prima para ser processada.

Figura 9 - Pseudocódigo da Heurística Construtiva

```

Procedimento Solução Inicial
1. Para todas as cores faça
2.   Para todos os tamanhos faça
3.     benefício ← FO + demanda * custo;
4.   Fim_para;
5. Fim_para;
6. Para todas as cores faça
7.   quicksort(benefício);
8. Fim_para;
9.  $k \leftarrow$  cores;
10. Para todas as estruturas de vizinhança faça
11.   quantidade_produto ← capacidade_maquina;
12.   Se  $k = 1$ 
13.     quantidade_produto ← 2 * quantidade_produto;
14.   Fim_se;
15.   Se  $k = 2$  OU  $k = 4$ 
16.     quantidade_produto ← 0,5 * quantidade_produto;
17.   Fim_se;
18.   Se  $k = 3$  OU  $k = 5$ 
19.     quantidade_produto ← 0,25 * quantidade_produto;
20.   Fim_se;
21. Para todas as cores faça
22.   Enquanto (demanda > produto_finalizado) E (tamanho ≤ quantidade_produto)
23.     produto_finalizado++;
24.     quantidade_produto ← quantidade_produto - tamanho;
25.   Fim_enquanto;
26. Fim_para;
27. Fim_para;
Fim

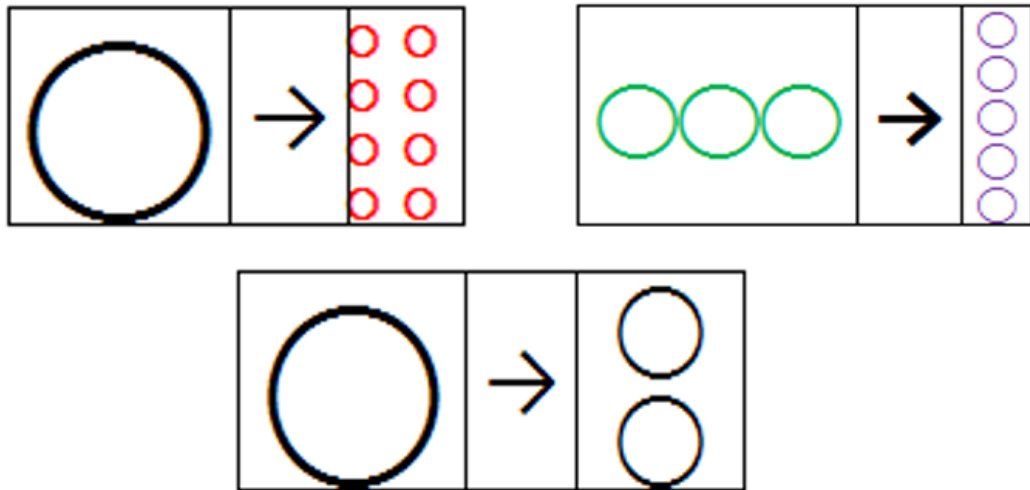
```

Fonte: Autoria Própria

4.4. Estruturas de Vizinhança

A estrutura de vizinhança é o local onde se encontram substituições viáveis e que serão utilizadas pelo VNS afim de encontrar novas soluções. Essa estrutura é composta por combinações de trocas possíveis de serem realizadas, como por exemplo, a substituição da fabricação de um produto preto de 2000 cm³ por oito produtos da mesma cor de tamanho 250 cm³ ou até mesmo dois produtos da mesma cor de tamanho 1000 cm³, ou simplesmente a substituição das cores. Na estrutura de vizinhança encontram-se diversas substituições equivalentes que serão avaliadas pela função objetivo na intenção de melhorar a solução. A Figura 10 ilustra como podem ser as estruturas de vizinhança.

Figura 10 - Ilustração das Estruturas de Vizinhaça



Fonte: Autoria Própria

Na imagem da Figura 10, o primeiro quadro representa a troca de um produto da cor preta de tamanho x por oito produtos da cor vermelha de tamanho $1/8$ de x cada. O segundo quadro indica a troca de três produtos da cor verde de tamanho y por cinco produtos da cor roxa de tamanho $3/5$ de y . O terceiro e último quadro representa a troca de um produto da cor preta de tamanho x por dois produtos da mesma cor de tamanho y . O objetivo da imagem é representar que as estruturas de vizinhaça podem utilizar diversos tipos de parâmetros para realizar a troca, neste caso, foram utilizados tamanhos e cores.

Para este trabalho, seis estruturas de vizinhaças foram criadas, e estão apresentadas na tabela 2. Para aumentar o número de estruturas, além da multiplicidade dos valores, ou seja, dobrar, triplicar ou multiplicar por valores aleatórios cada conjunto de estrutura, também foi realizado uma variação nas cores, de maneira randômica os números que representam cada uma das cores podem ou não serem trocados. O que significa que os produtos referentes as duas primeiras colunas podem ser de uma cor e os produtos referentes as outras duas colunas de outra.

Tabela 2 - Estruturas de Vizinhança

Atual		Troca	
Quantidade	Tamanho	Quantidade	Tamanho
1	2000 cm ³	8	250 cm ³
1	2000 cm ³	2	1000 cm ³
3	2000 cm ³	10	600 cm ³
1	1000 cm ³	4	250 cm ³
3	1000 cm ³	5	600 cm ³
5	600 cm ³	12	250 cm ³

Fonte: Autoria Própria

4.5. Função Objetivo

A Função Objetivo (FO) é a função responsável por avaliar as soluções e verificar quais apresentam os melhores resultados, tornando-as momentaneamente como solução corrente. A FO para este caso específico foi formulada considerando o lucro adquirido com a produção e venda do produto, descontando o prejuízo causado com o não atendimento da demanda. A Figura 11 apresenta o pseudocódigo da Função Objetivo.

Figura 11 - Pseudocódigo da Função Objetivo

Procedimento Função Objetivo	
1.	Para todas as cores faça
2.	Se produto_finalizado \geq demanda
3.	FO \leftarrow FO + demanda * lucro;
4.	Fim_se;
5.	Se produto_finalizado $<$ 0,5 * demanda
6.	FO \leftarrow FO + produto_finalizado * lucro;
7.	FO \leftarrow FO + (demanda - produto_finalizado) * lucro;
8.	Fim_se;
9.	Se (produto_finalizado $<$ demanda) E (produto_finalizado $>$ 0,5 * demanda)
10.	FO \leftarrow FO + produto_finalizado * lucro;
11.	FO \leftarrow FO - (demanda - produto_finalizado) * lucro/2;
12.	Fim_se;
13.	Fim_para;
Fim	

Fonte: Autoria Própria

Para a função objetivo partiu-se do conceito de que há lucro para os itens produzidos e que foram entregues, de acordo com a demanda, ou seja, se a quantidade produzida de um produto for maior ou igual ao valor da demanda para este produto, significa que há estoque suficiente para atendê-la, porém somente será gerado lucro pelos produtos que atingem até a quantidade demandada, caso haja excesso, o mesmo não será contabilizado na função objetivo, por isso que para o cálculo da função objetivo é multiplicado o valor da demanda pelo lucro, e não o valor da quantidade produzida pelo lucro.

Para os casos no qual a quantidade produzida não seja suficiente para atender toda a demanda será gerado lucro pelos produtos que forem entregues, porém haverá uma perda pelos produtos faltantes. A função objetivo soma a quantidade produzida vezes seu lucro, e subtrai a quantidade faltante, demanda menos a quantidade produzida, pelo seu valor equivalente.

Para o caso de não atendimento da demanda em mais de cinquenta por cento, o valor equivalente é o mesmo do lucro, ou seja, a perda de um produto é igual ao lucro gerado pela entrega do mesmo, neste caso a função objetivo será negativa, pois haverá mais perda do que lucro. Esta formulação foi utilizada para tentar evitar o não atendimento da demanda.

Para o caso de não atendimento da demanda em menos de cinquenta por cento, o valor multiplicador é metade do lucro, o que significa que ainda há uma perda, porém esta é menor do que no caso anterior.

4.6. *Quicksort*

O *Quicksort* é um algoritmo de ordenação utilizado para rearranjar conjunto de valores, como vetores e tabelas, em ordem crescente ou decrescente (FARIAS, 2012). O *Quicksort* foi o método escolhido para organizar os resultados de forma a apresentá-los na ordem decrescente.

O método de funcionamento do algoritmo é simples, um valor dentro do conjunto é selecionado, o chamado pivô, a partir dele é iniciado o método de separação. Caso a ordem crescente seja escolhida, os valores, menores que o pivô, serão considerados pequenos, e os maiores como grandes, e conseqüentemente ficarão a esquerda e a direita do pivô, respectivamente. Caso a ordem decrescente seja a escolhida a disposição será oposta (FARIAS, 2012).

Para representar graficamente o funcionamento, a Figura 12 apresenta um grupo de valores representados por colunas que são ordenados a partir do algoritmo *Quicksort*, o pivô

está representado sempre pela coluna em vermelho.

Figura 12 - Ilustração do funcionamento do *Quicksort*



Fonte: Autoria Própria

5. RESULTADOS OBTIDOS

As tabelas a seguir mostram os valores obtidos pelo programa desenvolvido.

Tabela 3 - Resultados para os Produtos da Cor Preta

Preto		
Valor da FO inicial		R\$ 6.268,00
Valor da FO final		R\$ 8.608,00
Tamanho	Demanda	Quantidade Produzida
250 cm ³	2.179	4.056
600 cm ³	2.656	1.330
1000 cm ³	2.493	0
2000 cm ³	8.292	7.278

Fonte: Autoria Própria

Para a cor preta houve uma melhora de 37,33% no valor da Função Objetivo e as quantidades produzidas representam 186,14%, 50,08%, 0,00% e 87,77% da demanda dos produtos de tamanho 250 cm³, 600 cm³, 1000 cm³, 2000 cm³, respectivamente.

Tabela 4 - Resultados para os Produtos da Cor Laranja

Laranja		
Valor da FO inicial		R\$ 1.823,00
Valor da FO final		R\$ 4.452,00
Tamanho	Demanda	Quantidade Produzida
250 cm ³	684	1.400
600 cm ³	4.388	4.210
1000 cm ³	1.754	1.750
2000 cm ³	3.577	2.702

Fonte: Autoria Própria

Para a cor laranja houve uma melhora de 144,21% no valor da Função Objetivo e as quantidades produzidas representam 204,68%, 95,94%, 99,77% e 75,54% da demanda dos produtos de tamanho 250 cm³, 600 cm³, 1000 cm³, 2000 cm³, respectivamente.

Tabela 5 - Resultados para os Produtos da Cor Verde

Verde		
Valor da FO inicial		-R\$ 723,00
Valor da FO final		R\$ 1.673,00
Tamanho	Demanda	Quantidade Produzida
250 cm ³	2.708	962
600 cm ³	5.384	4.784
1000 cm ³	2.419	2.659
2000 cm ³	1.673	1.675

Fonte: Autoria Própria

Para a cor verde houve uma melhora no valor da Função Objetivo de modo que fez com o retorno destes produtos deixassem de gerar prejuízo e para gerar lucro, as quantidades produzidas representam 35,52%, 88,86%, 109,92% e 100,12% da demanda dos produtos de tamanho 250 cm³, 600 cm³, 1000 cm³, 2000 cm³, respectivamente.

Tabela 6 - Resultados para os Produtos da Cor Vermelha

Vermelho		
Valor da FO inicial		R\$ 2.437,00
Valor da FO final		R\$ 3.294,00
Tamanho	Demanda	Quantidade Produzida
250 cm ³	1.979	600
600 cm ³	1.933	0
1000 cm ³	1.415	708
2000 cm ³	2.736	2.586

Fonte: Autoria Própria

Para a cor vermelho houve uma melhora de 35,17% no valor da Função Objetivo e as quantidades produzidas representam 30,32%, 0%, 50,04% e 94,52% da demanda dos produtos de tamanho 250 cm³, 600 cm³, 1000 cm³, 2000 cm³, respectivamente.

Tabela 7 - Resultados para os Produtos da Cor Roxa

Roxo		
Valor da FO inicial		R\$ 1.065,00
Valor da FO final		R\$ 1.207,00
Tamanho	Demanda	Quantidade Produzida
250 cm ³	3.348	127
600 cm ³	5.037	4.962
1000 cm ³	2.166	2.196
2000 cm ³	1.207	1.215

Fonte: Autoria Própria

Para a cor roxo houve uma melhora de 13,33% no valor da Função Objetivo e as quantidades produzidas representam 3,79%, 98,51%, 101,39% e 100,66% da demanda dos produtos de tamanho 250 cm³, 600 cm³, 1000 cm³, 2000 cm³, respectivamente.

As Tabelas de 3 a 7 indicam as quantidades que devem ser produzidas de todos os

produtos. É possível observar pelos valores das quantidades produzidas que não é simples identificar a lógica utilizada na determinação dos resultados, como por exemplo, produzir mais dos produtos que apresentam maior benefício, ou produzir a mesma quantidade dos produtos que tenham semelhança, o que representa como é difícil determinar o sequenciamento de produção manualmente.

Os valores das soluções obtidas antes e depois da aplicação da meta-heurística VNS indicam como é benéfico programar o sequenciamento utilizando um método de solução auxiliar, como algoritmos e meta-heurística, para problemas de grande dimensão. O lucro total calculado antes da aplicação do VNS foi de R\$ 10.870,00, enquanto que o calculado após foi de R\$ 19.234,00, o que representa uma melhora de 76,95% de retorno monetário.

As quantidades produzidas inferiores a demanda são justificadas pelo fato de que a condição do problema, ou seja a Função Objetivo, era maximizar o lucro e não atender ao máximo das demandas, porém a perda de não atender à alguma demanda foi contabilizada no cálculo da Função Fobjetivo. O aumento monetário significativo, de quase 77%, mostra que o modelo atingiu sua condição e maximizou o lucro.

É importante ressaltar que o resultado apresentado por este modelo, assim como em outros casos, deve ser analisado pelo tomador de decisão. Não há necessidade de se produzir mais de 100% da demanda para este caso, porque neste modelo não há estoque, porém o valor em excesso aparece porque o mesmo foi desenvolvido sem considerar interrupção da máquina por meio externo, somente caso a capacidade fosse atendida, logo alguns valores das quantidades produzidas são superiores a demanda.

Em situações na qual o produto não tem um ciclo de vida muito curto, e o produto final seja de fácil armazenamento, a produção até a capacidade máxima pode ser vantajosa, entretanto é tarefa do tomador de decisão analisar os custos relacionados ao armazenamento.

6. CONSIDERAÇÕES FINAIS

Este estudo apresenta como a meta-heurística VNS é útil na resolução de problemas de grande dimensão, como o caso do JSP. O modelo desenvolvido apresentou resultado satisfatório, visto que atendeu a maior restrição, a de maximizar o lucro, fato observado nos valores das funções objetivos e principalmente na porcentagem otimizada do lucro total, que representa um aumento de 76,95%.

O JSP foi escolhido por se tratar de um dos problemas mais comuns dentro no planejamento e controle da produção e de difícil solução, dado que o mesmo pode não apresentar solução ótima a partir de algoritmos simples.

A meta-heurística VNS foi a opção escolhida por se tratar de uma meta-heurística mais refinada e flexível, o que permite que os resultados encontrados sejam confiáveis e o modelo passível de adaptações, afim de solucionar outros problemas.

Os resultados apresentados mostram a melhora significativa nas soluções após a aplicação da meta-heurística. A mesma conseguiu encontrar os valores a serem produzidos de modo que o lucro obtido fosse otimizado.

Os resultados apresentados também contribuem para a avaliação de outras atividades presente nas empresas, como a questão de produzir mais que a demanda afim de estocar o produto, ou avaliar se pelas quantidades produzidas a empresa está atingindo sua capacidade máxima e haja necessidade de flexibilizar o serviço, abrindo uma nova janela de trabalho para os responsáveis pelo planejamento e controle de produção.

Este trabalho contribui para a realidade atual por se tratar de um modelo genérico para um problema comum, ou seja, o modelo desenvolvido é passível de ser ajustado para solucionar os problemas das empresas ou melhorar seus processos. É necessário modificar a formatação dos dados de entrada e criar estruturas de vizinhanças compatíveis com o problema, com as tarefas e com a produção.

REFERÊNCIAS

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. Pesquisa operacional para cursos de engenharia. Campus Elsevier, Rio de Janeiro, 2007.

BREMER, C. F.; LENZA, R. P. Um modelo de referência para gestão da produção em sistemas de produção *assembly to order* – ATO e suas múltiplas aplicações. *Gestão & Produção*, v. 7, p. 268-282, dez 2000.

FARIAS, F. H.; SILVA, F. B. M.; NETO, J. B. *Quicksort* e *Quicksort* Aleatorizado: Um estudo comparativo. Anais do Congresso de Matemática Aplicada e Computacional, CMAC Nordeste 2012.

FIGUEIREDO, M. V.; LUCHE, J. R. D.; FRANCO, B. C.; SALGADO, A. M. P. Sequenciamento da produção em uma linha de usinagem industrial utilizando a Meta-heurística *Simulated Annealing*. *Gestão e Tecnologia para a Competitividade*. Outubro, 2013.

FREITAS, L. M. B.; MONTANÉ, F. A. T. Meta-heurísticas VNS-VND e GRASP-VND para problemas de roteamento de veículos com coleta e entrega simultânea. SPOLM, Rio de Janeiro, Brasil, agosto de 2008.

GAO, J.; SUN, L.; GEN, M. *A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems*. *Computer & Operations Research* 2007.

GARCÍA-LÓPEZ, F.; MELIÁN-BATISTA, B.; MORENO-PÉREZ, J. A.; MORENO-VEJA, J. *The Parallel Variable Neighborhood Search for the p-Median Problem*. *Journal of Heuristics*, 8: 375-388, 2002.

GLOVER, F.; KOCHENBERGER, G. A. *Handbook of metaheuristics*. *International Series in Operations Research & Management Series, Kluwer's International Series, Stanford University*, 2003.

JAIN, A. S.; MEERAN, S. *Deterministic job-shop scheduling: past, present and future*. *Department of Applied Physics and Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, UK, DD1 4HN*. 1998.

LUSTOSA, L.; MESQUITA, M. A.; QUELHAS, O.; OLIVEIRA, R. Planejamento e Controle da Produção. Rio de Janeiro : Elsevier, 2008.

MELO, E. L.; RONCONI, D. P. Heurísticas construtivas para a minimização do atraso total no ambiente *job shop* flexível. Simpósio Brasileiro de Pesquisa Operacional. Setembro, 2012.

MLADENOVIC, N.; HANSEN, P. *Variable neighborhood search*. *Computers and*

MULLER, G. I.; GÓMEZ, A. T. Utilização da Busca Tabu para a geração de um modelo aplicado do *Job-shop Scheduling Problem* considerando um sistema de manufatura flexível. PIPCA – Programa Interdisciplinar de Pós-Graduação em Computação Aplicada. São Leopoldo, RS – Brasil, 2006.

OR-Library. Disponível em: <<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>>. Acesso em: 01 de maio de 2012.

PAULA, M. R.; RAVETTI, M. G.; PARDALOS, P. M. Abordagem *Variable Neighborhood Search* para o problema de sequenciamento com máquinas paralelas e tempos de preparação dependentes da sequência. XXXVIII Simpósio Brasileiro de Pesquisa Operacional. Goiânia, GO, 2006.

POLACEK, M.; HARTL, R. F.; DOERNER, K. *A Variable Neighborhood Search for the multi depot vehicle routing problem with time windows*. *Journal of Heuristics*, 10: 613-627, 2004.

REIS, J. V. A.; CUNHA, C. B. Uma heurística baseada em VNS para o problema bidimensional de binpacking com frota heterogênea. In: XVI PANAM, Lisboa, Portugal. Julho, 2010.

SILVA, E. B.; COSTA, M. G.; SILVA M. F. S.; PEREIRA, F. H. Avaliação de regras de sequenciamento da produção em ambientes *Job shop* e *Flow shop* por meio de simulação computacional. *Exacta*, São Paulo, v. 10, n. 1, p. 70-81, 2012.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. Administração da produção. 3. ed. – São Paulo : Atlas, 2009.

TAHA, Hamdy A. Pesquisa operacional: uma visão geral / Hamdy A. Taha ; tradução Arlete Simille Marques; revisão técnica Rodrigo Arnaldo Scarpel. – São Paulo: Pearson Prentice Hall, 2008.

TOSO, E. A. V.; MORABITO, R. Otimização no dimensionamento e sequenciamento de lotes de produção: estudo de caso numa fábrica de rações. *Gestão & Produção*, v. 12, p. 203-217, mai.-ago. 2005.

TUBINO, D. F. Planejamento e controle da produção: teoria e prática / Dalvio Ferrari Tubino. - 2. ed. – São Paulo: Atlas, 2009.

VOLLMANN, T. E.; BERRY, W. L.; WHYBARK, D. C.; JACOBS, F. R. Sistemas de planejamento e controle da produção para o gerenciamento da cadeia de suprimentos. Tradução Sandra de Oliveira – 5. ed. – Porto Alegre : Bookman, 2006.

VOLPATO, L. T.; CONSTANTINO, A. A.; ROMÃO, W.; SAVINIEC, L. Meta-heurística VNS aplicada ao problema de escalonamento de enfermeiros. XLV Simpósio Brasileiro de Pesquisa Operacional. Natal/RN, setembro, 2013.