# ***Máquinas Paralelas Não Relacionais Com tempo de Setup Dependente da Sequência***

Hesau Hugo Leite *(Autor)*

Ingrid Cardoso de Souza (*Autor*)

Gustavo Valentim Loch (orientador) (*Autor*)

2021

***Resumo***— **Este artigo terá como objeto de estudo Máquinas Paralelas Não Relacionais para minimização do tempo de Makesapn, desenvolvendo uma breve explicação sobre o conceito de Máquinas Paralelas, explicação de termos importantes, definição do problema abordado e a demonstração da formulação matemática e resultados atingidos com a implementação da Vizinha Swap e Insert.**

Palavras-chave: Máquinas Paralelas Não Relacionais; Definições e Conceitos; Vizinha Swap e Insert.

***Abstract — This article will study Parallel Non-Relational Machines for Makesapn time minimization, developing a brief explanation of the concept of Parallel Machines, explanation of important terms, definition of the problem addressed and demonstration of the mathematical formulation and results achieved with the implementation of Neighborhood Swap and Insert.***

***Keywords: Parallel Non-Relational Machines; Definitions and Concepts;Neighbor Swap and Insert.***

# Introdução

Atualmente diante da complexidade de tarefas e problemas para otimizar os resultados, a problemática das Máquinas Paralelas Não Relacionais é um formidável desafio a ser estudado e resolvido em diversos setores do mercado e acadêmico. Mais facilmente encontrado em ambiente fabril, mas não limitado a este, o uso de Máquinas Paralelas Não Relacionais é comum na produção de bens e serviços devido, muitas vezes, a mudanças de tecnologia, custos, mão de obra, etc., na qual a alocação correta do recurso, de acordo com a máquina adequada e a tarefa a ser executada, trazem a dificuldade ao tema.

Diversas podem ser as maneiras para alocar com recursos, seja com o objetivo de maximizar ou minimizar alguma coisa. Neste artigo será abordado o uso da implementação da Vizinha Swap e Insert, afim de minimizar o tempo de Makespan, ou seja, diminuir o tempo total preparação e processamento/execução de uma tarefa.

Assim, por meio da aplicação da literatura conhecida e conhecimento acadêmico, o artigo contribuirá com as discussões a respeito da implementação da Vizinha Swap e Insert para resolver uma situação em ambiente de Máquinas Paralelas Não Relacionais.

# Fundamentação Teórica

Nessa seção serão apresentados os objetivos, definições do Problema de Máquinas Paralelas não relacionais com tempo de Setup Dependente da Sequência.

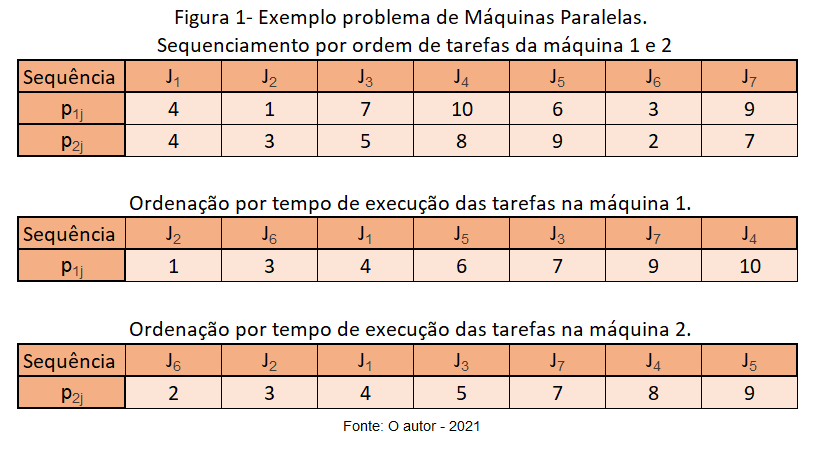
## Máquinas Paralelas

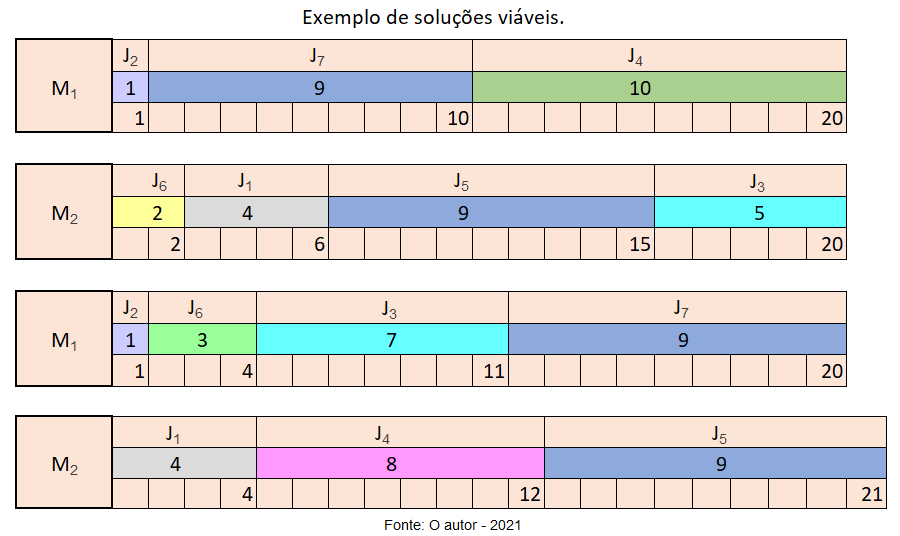
Paralelo, em matemática segundo o dicionário significa que, estando num mesmo plano, não se cortam, conservando sempre a mesma distância, entretanto no sentido figurado o dicionário define que é mantém uma atividade (emprego, função etc.) juntamente com outra(s), a partir desta definição vamos conceituar máquinas paralelas.

De acordo com Almasi e Gottelib, 1989- Máquina paralela é um conjunto de processadores que se comunicam e cooperam para a resolução de um problema de grande porte.

O problema de máquinas paralelas se trata da alocação de um determinado número de tarefas independentes, com tempos de execução conhecidos, para um outro número de máquinas paralelas. As máquinas paralelas podem ser classificadas em idênticas, uniformes e não relacionadas. As idênticas possuem o mesmo tempo de processamento e preparação. As uniformes o tempo de processamento e preparação são proporcionais. Já as não relacionadas, não existem relação entre o tempo de processamento e preparação das diferentes máquinas.

Nesse trabalho vamos abordar sobre máquinas paralelas não relacionadas, um exemplo está ilustrado na figura 1 com um exemplo de solução viável.





De acordo com Pinedo (2016), as máquinas possuem classificações, algumas estão definidas a seguir:

* Job Shop: são máquinas diferentes em ordem diferentes;
* Máquinas paralelas: Os Jobs precisam de apenas um processamento e independe da máquina;
* Setup: O tempo de preparação de uma tarefa a outra na máquina;
* Makespan: O tempo total de execução de cada tarefa, ou seja, tempo de processamento somado ao tempo de setup.

## Definição do Problema

No problema UPMSP (unrelated parallel machine scheduling problem with sequence dependente setup times) existe um conjunto de tarefas pré-estabelecidas que devem ser processadas em um conjunto de máquinas já conhecidas, sendo que cada tarefa deve ser processada somente em uma máquina. O tempo de execução de cada tarefa está associado a cada máquina. Esse tempo pode variar dependendo do equipamento escolhido. Entre duas tarefas há também, um tempo de setup, o qual depende da sequência dessas tarefas e da máquina associada. O objetivo é minimizar o tempo de conclusão de todas essas tarefas, conhecido como makespan.

## Formulação Matemática

***Parâmetros:***

*N*: conjunto de tarefas que serão processadas;

*M*: conjunto de máquinas paralelas não relacionadas;

parâmetro que armazena o tempo de processamento da tarefa *j* na máquina *i* ;

: parâmetro que armazena o tempo de preparação quando a tarefa *j* precedem, imediatamente a tarefa *k* na máquina *i* ;

***Variáveis:***

:

variável que armazena o tempo de conclusão da tarefa j na máquina i ;  
 variável que armazena o tempo total de processamento de todas as tarefas, ou seja, o valor do makespan.

***Função Objetivo:***

*Sujeito a:*

A função objetivo é representa por (1), inclui-se problema uma tarefa fictícia, com o tempo de preparação e conclusão igual zero, para vir no início da programação como mostra a restrição (2). A restrição (3) determina o tempo de preparação. A restrição (4) faz com que o tempo de conclusão seja zero para todas as tarefas fictícias. A restrição (5) faz com que uma única tarefa seja processe em somente uma máquina. A restrição (6) determina 1 para cada tarefa sucessora e a última tarefa zero, pois não tem sucessora. A restrição (7) complementa a restrição (6), permitindo que 1 seja o número máximo de sucessoras das tarefas fictícias. A restrição (8) verifica se a tarefa j é processada na máquina i, se sim, existe uma h predecessora na máquina i. A restrição (9) determina que se xijk = 1, o tempo para completar a tarefa k deve ser maior ou igual a tempo para completar a tarefa j, o que significa que o próximo conjunto de tempo de tarefa mais setup deve ser maior ou igual ao conjunto anterior. A restrição (10) garante a não negatividade para o tempo de conclusão das tarefas. A restrição (11) determina o tempo de conclusão máximo.

# Resolução do Problema

Para a resolução do problema apresentamos as heurísticas da vizinhança, insert e heurística construtiva e seguida a apresentação dos experimentos computacionais.

## Heurística

#### Vizinhanças

De acordo com BLUM e ROLI, 2003 para melhorar uma solução e/ou escapar do ótimo local, pode-se fazer uma busca local, que corresponde em percorrer interagindo com espaço de busca partindo de uma solução inicial para o uma solução vizinha. Para essa exploração de busca local, pode-se aplicar diversas vizinhanças que são funções que representam cada solução para um subconjunto de vizinhos.

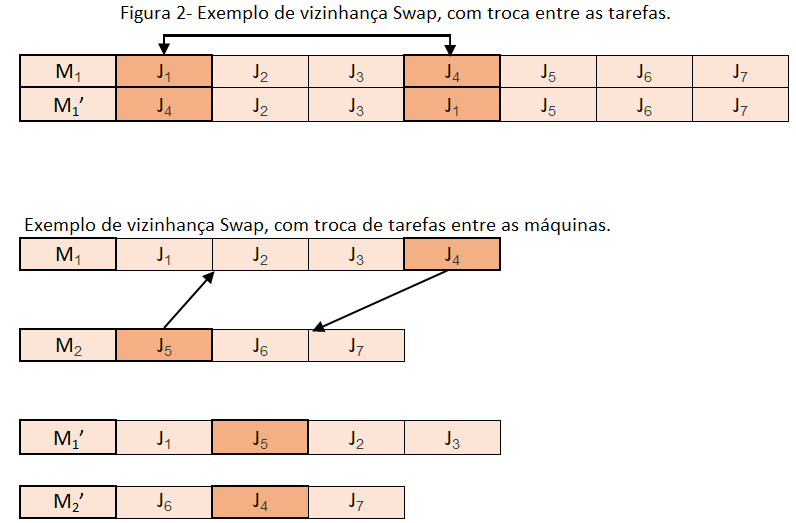
Para esse trabalho foi implementado a vizinha swap e Insert.

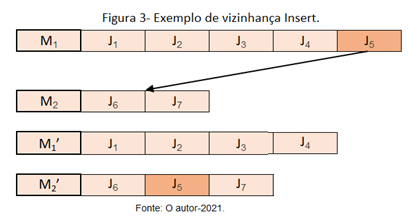
#### Vizinhanças Swap

Gera vizinhos com movimentos de trocas, nesse caso pode-se ser feita a troca entre as máquinas na literatura são conhecidas também como Switch e/ou troca entre as tarefas em uma e/ou nas duas. A figura 2 ilustra um exemplo.

#### Vizinhanças Insert

Nesse caso geram vizinhos através das retiradas uma tarefa da máquina que tem o maior makespan e alocada na que tem o menor makespan. A figura 3 ilustra um exemplo.





## Heurística Construtiva

* Será inserido as tarefas em cada máquina de forma que todas sejam atendidas;
* A quantidade incluída será um sorteio de uma posição entre a metade da quantidade de tarefas remanescentes;
* Será incluída na máquina a quantidade relativa à posição sorteada.

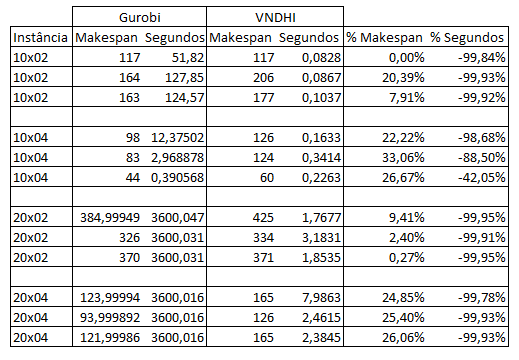
## Experimentos Computacionais

Para o desenvolvimento da solução exata dos experimentos foi usado a linguagem de programação C# e o solver Gurobi 9.1. Para as meta-heurísticas foi usado python. As instâncias foram usadas de Vallada e Ruiz(2011) apenas as small. O computador no qual os experimentos foram realizados foi em um Dell SSD x64 com processador Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz e 8GB de RAM.

# Resultados

Conforme representado na Figura 3 pode-se concluir que a meta-heurística desenvolvida tem uma performance muito rápida se comparada com o solver, porém apenas para a classe de instâncias small 20x02 o algoritmo obteve melhores gaps.

Figura 3 -Comparação Gurobi x VNDHI



# Conclusão

A meta-heurística desenvolvida obteve êxito em tempo de execução para as instâncias small testadas, o seu gap ficou por volta de 25% do ótimo. Para estudos futuros será interessante testar mais classes de instâncias.

##### Referências

1. Artigo Máquinas Paralelas: Revisão Sistemática da Literatura -Lorena Carvalho da Paz, José Eduardo Pécora Júnior-2019.
2. Moura, Rodrigues, Costa, Santana “MÁQUINAS PARALELAS: MULTICOMPUTADORES”, 2018.
3. Müller,  Dias, Araújo, “ Algoritmo para o problema de seqüenciamento em máquinas paralelas não-relacionadas”, 2007.
4. Patrícia Belfioret, Luiz Paulo Fávero, “Pesquisa Operacional Para cursos de engenharia ,” 2013.
5. Ravetti, “Problemas de seqüenciamento com máquinas paralelas e tempos de preparação dependentes da seqüência.”, 2003.