



UFOP

Universidade Federal  
de Ouro Preto

UNIVERSIDADE FEDERAL DE OURO PRETO  
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



# Revisitando o Revenimento Paralelo: Computação de Alto Desempenho e Aplicação em Pesquisa Operacional

André Luís Barroso Almeida

Orientador: Prof. Dr. Marco Antonio Moreira de Carvalho

Coorientador: Prof. Dr. Joubert de Castro Lima

# AGENDA

**1**

**INTRODUÇÃO**

**2**

**REVISÃO SISTEMÁTICA  
DE LITERATURA**

**3**

**REVENIMENTO PARALELO**

**4**

**IMPLEMENTAÇÃO  
PARALELA PARA CPU**

**5**

**METODOLOGIA  
EXPERIMENTAL**

**6**

**PARALELIZAÇÃO DA  
FUNÇÃO DE AVALIAÇÃO**

**7**

**PRIMEIRO ESTUDO DE CASO**

**8**

**SEGUNDO ESTUDO DE  
CASO**

**9**

**TERCEIRO ESTUDO DE  
CASO**

**10**

**API REVENIMENTO  
PARALELO**

**11**

**PUBLICAÇÕES**

**12**

**CONCLUSÃO E TRABALHOS  
FUTUROS**

## Motivação

- Metaheurísticas são métodos generalistas com capacidade de explorar de forma eficiente e eficaz o espaço de busca;
- Metaheurísticas são, intrinsecamente, não escaláveis, tornando o uso do *paralelismo* um caminho natural;
- Apesar do crescente interesse em metaheurísticas paralelas, essa área de estudo é extremamente jovem e dinâmica;
- Metaheurísticas paralelas compõem uma linha de pesquisa extremamente desafiadora, interessante e contemporânea.

### Objetivos específicos

1. Elaborar uma rigorosa revisão sistemática da literatura relacionada à metaheurísticas paralelas baseadas em trajetória e identificar suas lacunas.
2. Gerar embasamento teórico sobre arquiteturas paralelas e distribuídas.
3. Projetar e implementar uma metaheurística paralela em CPU.
4. Avaliar o desempenho do método proposto em casos de estudo.
5. Revisitar as funções de avaliação dos estudos de caso.
6. Criar e disponibilizar uma API contendo o método proposto.
7. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos.

**2**

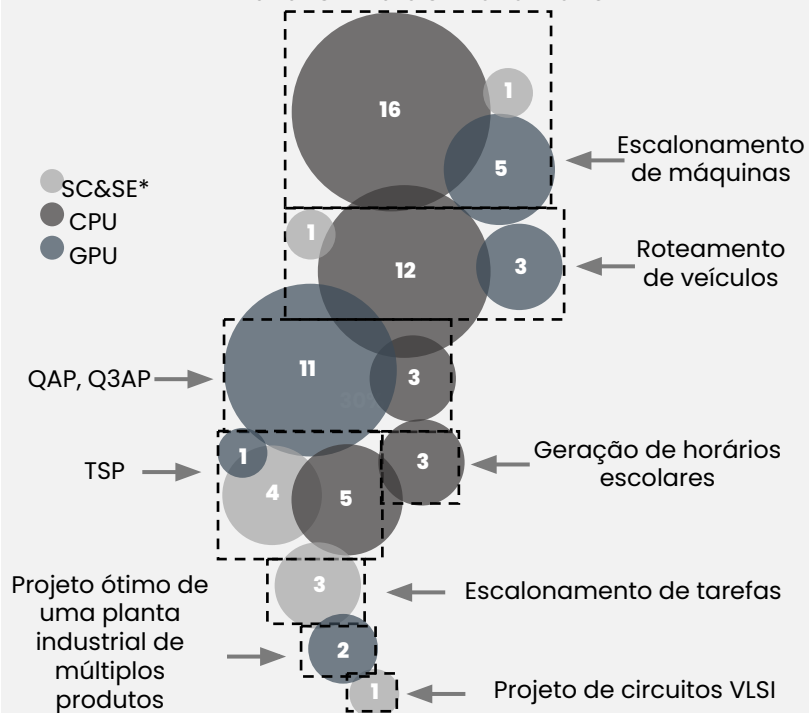
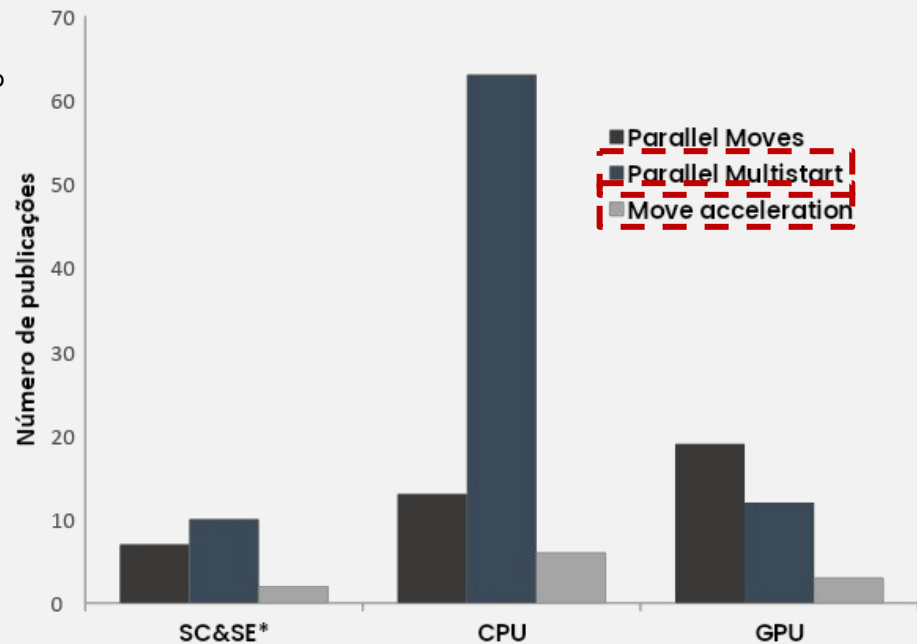
**REVISÃO SISTEMÁTICA  
DE LITERATURA**

**REVISÃO SISTEMÁTICA  
DE LITERATURA**

**Metaheurísticas paralelas baseadas em trajetória**



Revisão sistemática de literatura baseada nas propostas em [Brereton et al. \(2007\)](#) e [Kitchenham et al. \(2010\)](#).

Cinco problemas de otimização mais  
abordadosx Plataforma de *hardware*Modelo de paralelização x Plataforma de  
*hardware*

\*Supercomputadores e sistemas embarcados

## Achado durante a revisão de literatura

- Em outros contextos, como física e biologia, é utilizado com frequência a metaheurística Revenimento Paralelo (*Parallel Tempering*, PT);
- Segundo [Junghans e Hansmann \(2006\)](#), é sensato empregar o PT em conjunto com sistema *multi-core*;
- O PT apresenta notáveis resultados na área de simulação com uma convergência rápida se comparado com o SA;
- Geralmente é implementado de forma sequencial.



## Achado durante a revisão de literatura

- A origem do PT pode ser atribuída, segundo [Earl e Deem \(2005\)](#), ao trabalho de [Swendsen e Wang \(1986\)](#);
- Cinco anos depois, [Geyer \(1991\)](#) propôs um trabalho mais completo, sendo considerado a origem do PT;
- No mesmo ano, [Kimura \(1991\)](#) propôs o mesmo método aplicado a um problema de otimização;
- Utilização em otimização combinatória é rara, sendo abordado em 10 artigos.

3

REVENIMENTO  
PARALELO

REVENIMENTO  
PARALELO

Metaheurística conhecida como *Parallel Tempering*

## Introdução

- O PT mimetiza o processo físico de revenimento;
- Vastamente utilizado em simulações nas áreas de físico-química e biologia;
- O PT, por definição, é um método baseado no *Markov Chain Monte Carlo* (MCMC);
- Explora distribuições de probabilidade com várias dimensões.

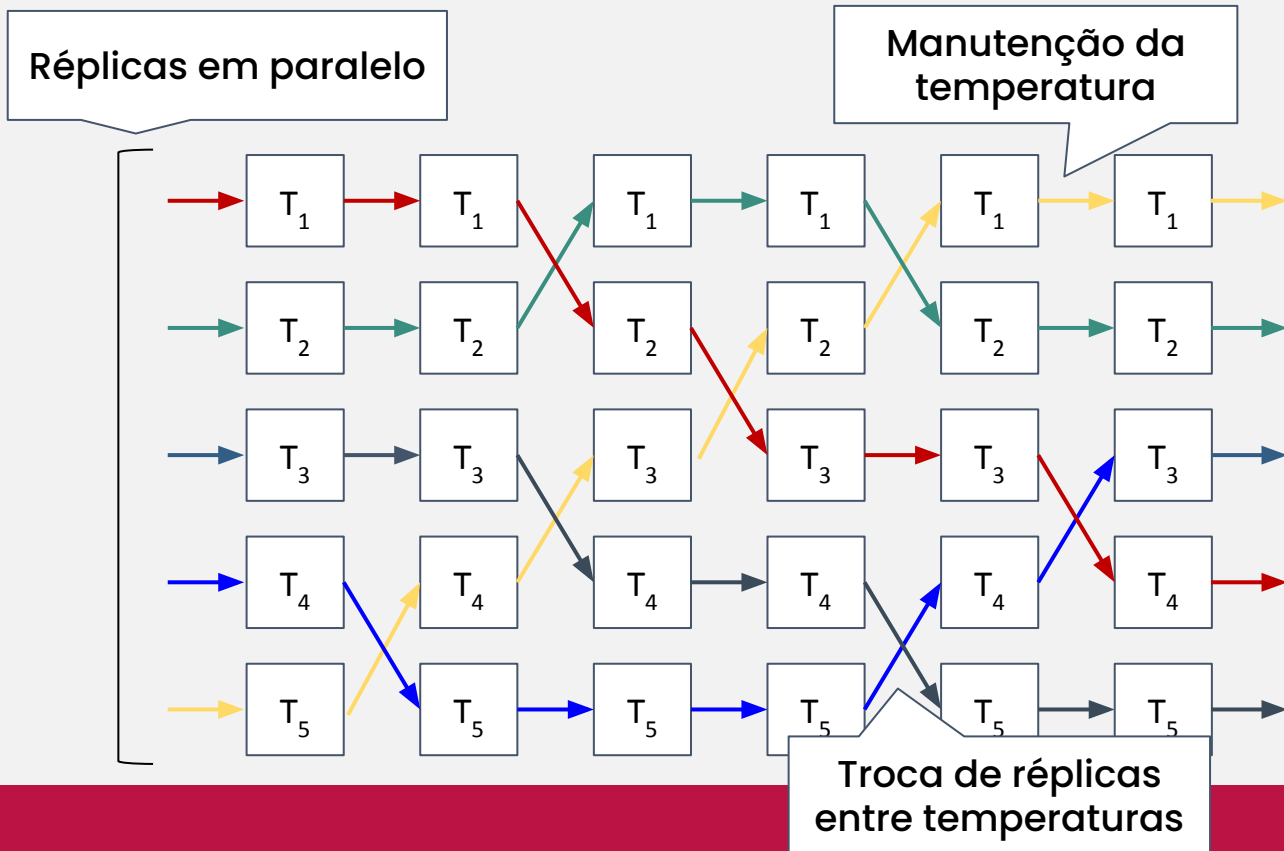
## Introdução

- Monte Carlo representa uma classe de algoritmos com o propósito de resolver problemas através de amostragens aleatórias;
- O MCMC consiste na geração de uma sequência dependente aleatória de soluções.

$$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \dots S_{n-1} \rightarrow S_n$$

## Características

- O PT consiste na coordenação da constante  $T$ , conhecida como temperatura na distribuição de *Boltzmann*;
- No algoritmo PT, réplicas ou cópias do sistema ( $R = \{r_1, r_2, r_3, \dots, r_n\}$ ) são simuladas em diferentes temperaturas, ou seja, diferentes valores de  $T$ ;
- Cada réplica, a uma temperatura fixa, simula o sistema em uma quantidade predefinida de passos;
- Após o processo anterior, uma troca de réplicas entre temperaturas adjacentes é proposta.



### Objetivos específicos

1. Elaborar uma rigorosa revisão sistemática da literatura relacionada à metaheurísticas paralelas baseadas em trajetória e identificar suas lacunas.
2. Gerar embasamento teórico sobre arquiteturas paralelas e distribuídas.
3. Projetar e implementar uma metaheurística paralela em CPU.
4. Avaliar o desempenho do método proposto em casos de estudo.
5. Revisitar as funções de avaliação dos estudos de caso.
6. Criar e disponibilizar uma API contendo o método proposto.
7. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos.

# 4

## IMPLEMENTAÇÃO PARALELA PARA CPU

### IMPLEMENTAÇÃO PARALELA DO PT EM CPU

Paralelização do PT utilizando o modelo *dataflow*



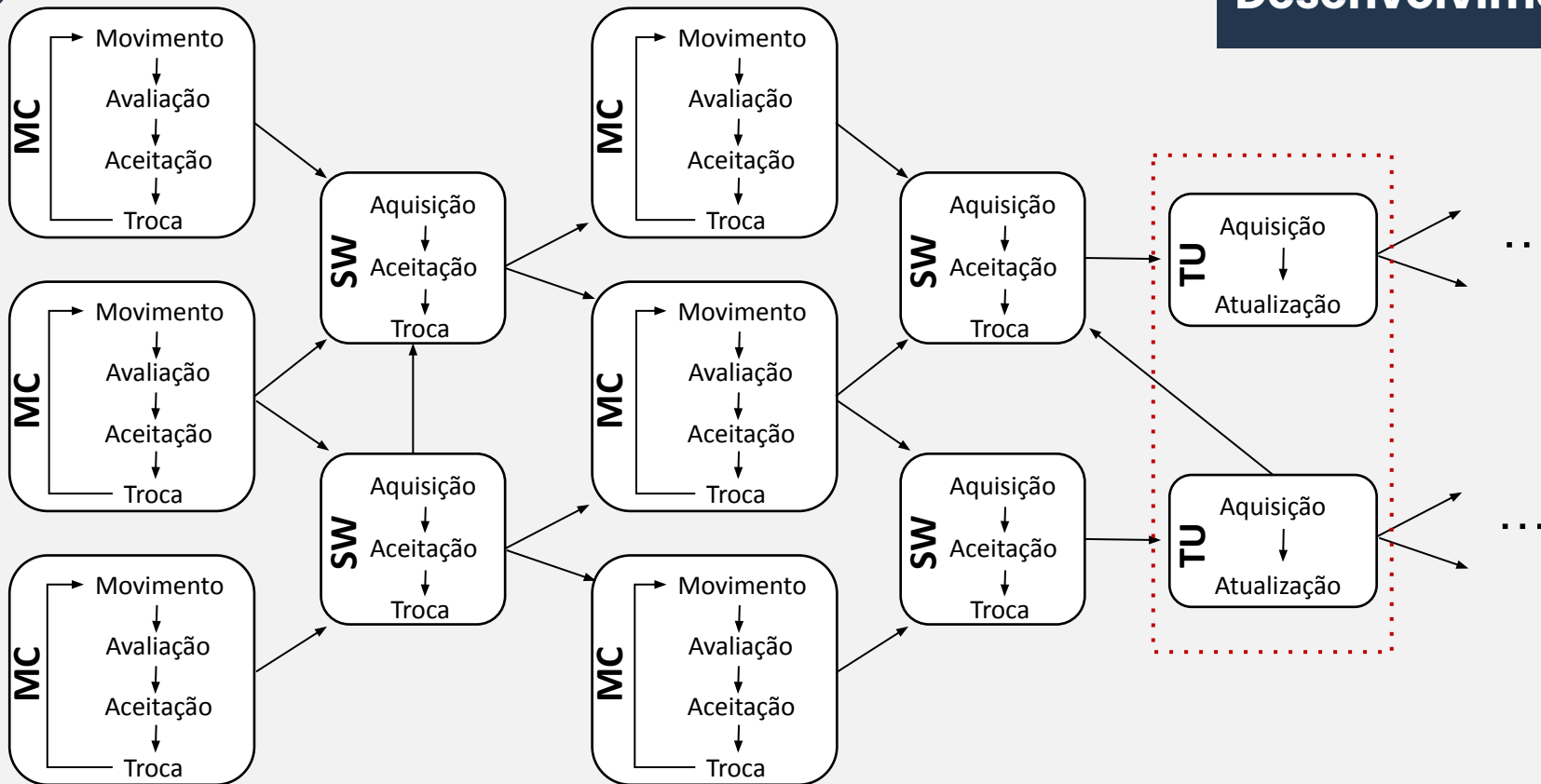
## Introdução

- Implementação paralela *coarse-grained*;
- Implementação que utiliza o modelo de programação paralela *dataflow*;
- Cada nó, cujas dependências tenham sido atendidas, é adicionado a uma fila que controla o fluxo de execução (*ThreadPool*);
- Paralelização segundo a taxonomia de [Crainic e Hail \(2005\)](#) do tipo *p-control* (pC);
- Percurso em grafos e um dos e 13 “dwarfs” listados no trabalho de [Asanović et al. \(2006\)](#).

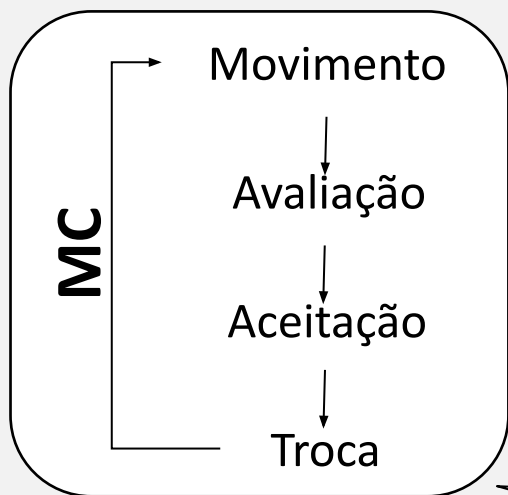
## 4

IMPLEMENTAÇÃO  
PARALELA PARA CPU

## Desenvolvimento



## Cálculo da cadeia de Markov homogênea



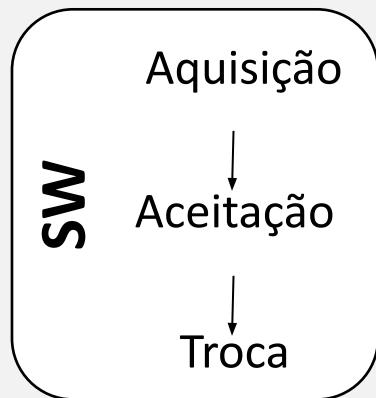
Uma solução vizinha é gerada de acordo com um movimento previamente definido e relacionado ao **problema abordado**.

Neste componente é calculado o valor referente e função de avaliação do **problema abordado**.

Neste componente um critério de aceitação baseado na distribuição de *Boltzmann* é aplicado

A nova solução gerada pode ou não se tornar a solução atual.

Executa a troca de réplicas entre temperaturas adjacentes



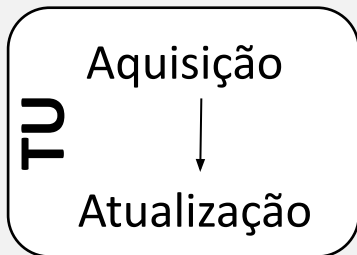
Na aquisição, a temperatura e as soluções vinculadas ao nó MC conectados ao nó SW são recuperadas e transferidas para o próximo passo.

Neste componente, as soluções e temperaturas são submetidas a um critério de aceitação probabilístico de troca baseado na equação

$$P(r_i \rightarrow r_j) = \min[1, \exp(\Delta\beta\Delta E)]$$

Troca de réplicas entre temperaturas adjacentes

Atualiza os valores da temperatura de forma dinâmica



Na aquisição, a temperatura, as taxas de aceitação e o fluxo relacionado ao *round-trip* são recuperadas e transferidas para o próximo passo.

Neste componente, ao receber os dados estatísticos as temperaturas são atualizadas de acordo com o método previamente escolhido.

### Objetivos específicos

1. Elaborar uma rigorosa revisão sistemática da literatura relacionada à metaheurísticas paralelas baseadas em trajetória e identificar suas lacunas.
2. Gerar embasamento teórico sobre arquiteturas paralelas e distribuídas.
3. Projetar e implementar uma metaheurística paralela em CPU.
4. **Avaliar o desempenho do método proposto em diferentes casos de estudo.**
5. Revisitar as funções de avaliação dos estudos de caso.
6. Criar e disponibilizar uma API contendo o método proposto.
7. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos.

**5**

**METODOLOGIA  
EXPERIMENTAL**

**METODOLOGIA  
EXPERIMENTAL**

**Recomendações do projeto experimental**



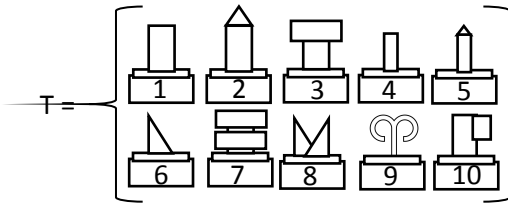


## SSP

- O Problema de minimização de trocas de ferramentas (SSP) vem recentemente recebendo atenção da comunidade científica;
- Diferentes variações do SSP podem ser encontradas na literatura, sendo caracterizadas de acordo com:
  - As características das ferramentas
  - As características das máquinas
  - As características das tarefas

## Problema de minimização de trocas de ferramentas

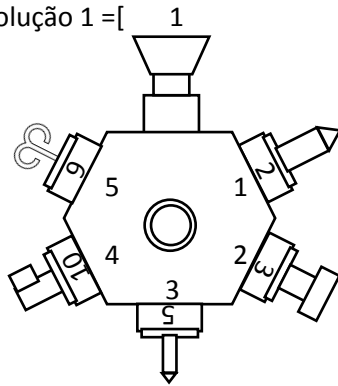
Conjunto de ferramentas (T)



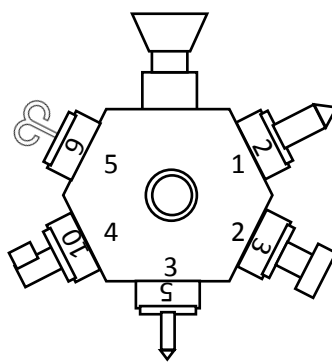
Conjunto de tarefas(J)

Tarefa	Ferramentas				
1	2	3	5	10	
2	1	5	7	8	
3	9	3	5	10	

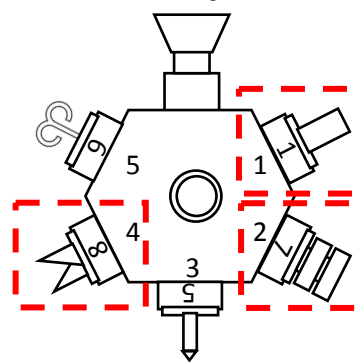
Solução 1 = [



3



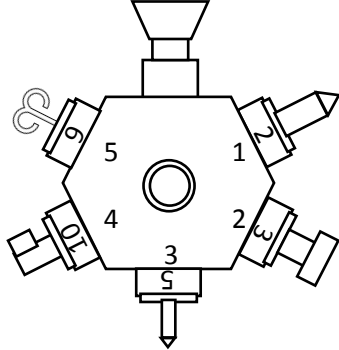
2 ]



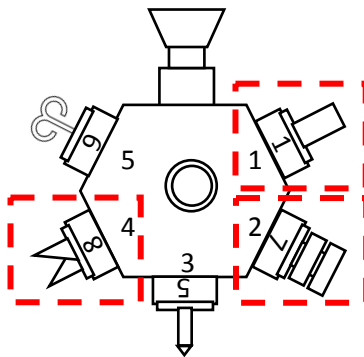
Total de trocas = 3

## Problema de minimização de trocas de ferramentas

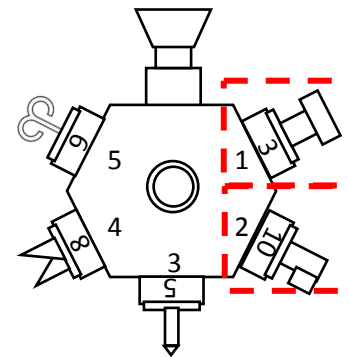
Solução 2 = [ 1



2



3]



Total de trocas = 5

### Objetivos específicos

1. Elaborar uma rigorosa revisão sistemática da literatura relacionada à metaheurísticas paralelas baseadas em trajetória e identificar suas lacunas.
2. Gerar embasamento teórico sobre arquiteturas paralelas e distribuídas.
3. Projetar e implementar uma metaheurística paralela em CPU.
4. Avaliar o desempenho do método proposto em casos de estudo.
5. Revisitar as funções de avaliação dos estudos de caso.
6. Criar e disponibilizar uma API contendo o método proposto.
7. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos.

**6**

## **PARALELIZAÇÃO DA FUNÇÃO DE AVALIAÇÃO**

**KTNS E GPCA**

**Implementações paralelas das funções de avaliação**

## Problema de alocação de ferramentas

- Frequentemente, ao abordar o SSP, é necessário calcular a quantidade de trocas de ferramentas;
- Felizmente existem dois algoritmos capazes de determinar esse valor em tempo polinomial, o *Keep Tool needed soonest* (KTNS) e o *Greedy Pipe Construction algorithm* (GPCA);
- É de conhecimento que o tempo para a função de avaliação pode representar até 90% do tempo total de uma metaheurística (Janiak, Janiak e Lichtenstein, 2008);

## KTNS paralelo

Sequência de processamento  
e ferramentas necessárias

			5	5
3	3	9	6	6
2	7	7	1	7
1		4		

↑  
Meio

KTNS da esq-dir



7	1
3	3
2	7
1	2

Trocas 0

KTNS da dir-esq

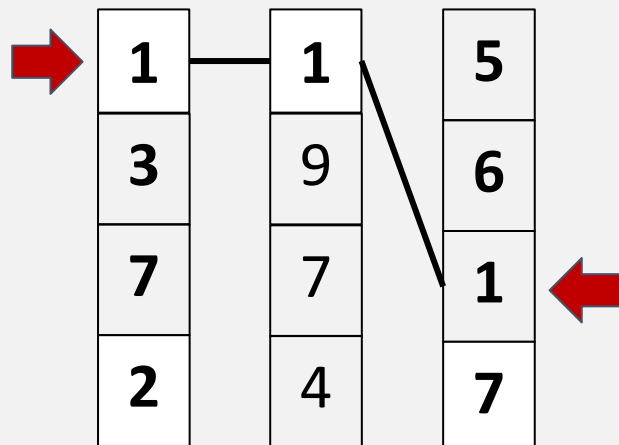


5	5
6	6
1	7
7	1

Trocas 0

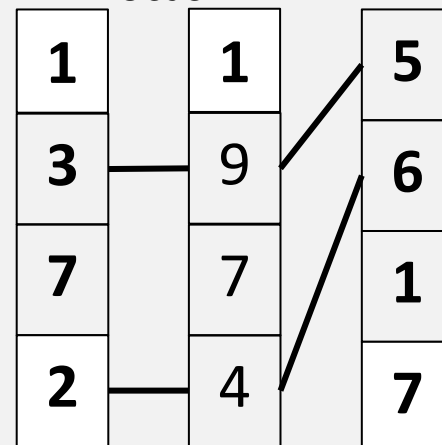
## KTNS paralelo

Preencher as lacunas



Determinar as trocas

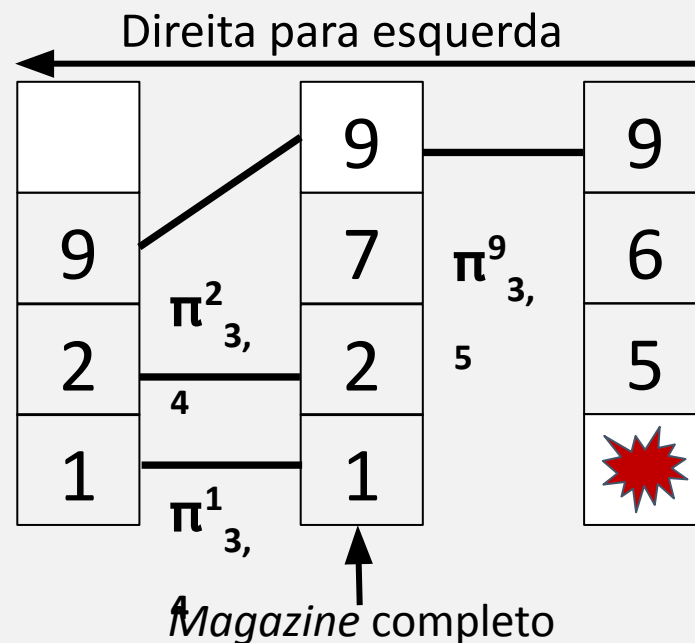
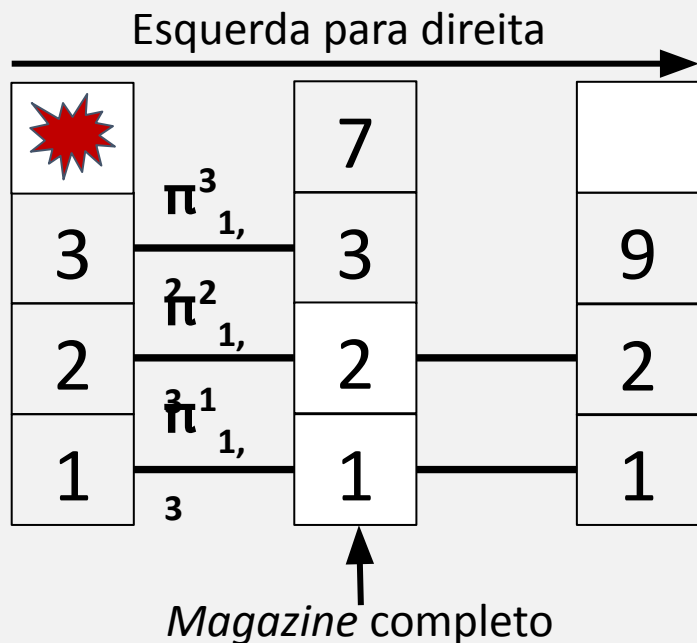
Trocas 2



Trocas 2

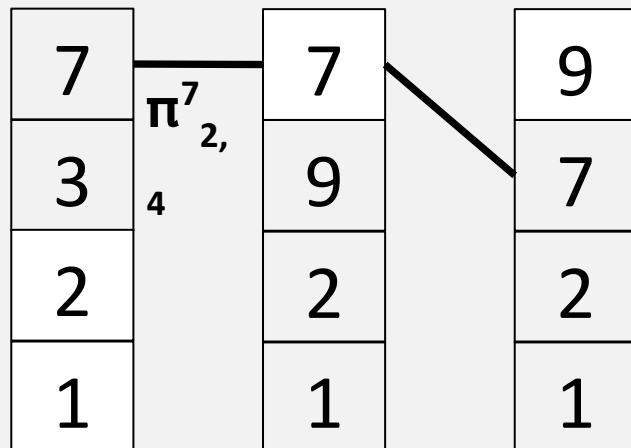


## GPCA paralelo



## GPCA paralelo

Etapa de união dos  
subproblemas



Ao final da etapa de união  
dos subproblemas, a  
quantidade total de trocas e  
dada por:

$$\sum_{i=1}^n |T_i| - C - |\pi|$$

## Experimentos

- Experimentos foram realizados com o objetivo de avaliar o tempo computacional exigido pelas implementações paralelas;
- Foram consideradas as instâncias propostas por [Catanzaro, Gouveia e Labbé \(2015\)](#) e [Mecler, Subramanian e Vidal \(2021\)](#);
- Ambas as implementações paralelas demonstraram uma redução significativa no tempo computacional, chegando a 93,80% e 96,57%;
- GPCA + *ToFullMag* chega a uma redução de até 85,42%.

## Experimentos

- O GPCA + ToFullMag paralelo obteve uma aceleração de 1,87 vezes próxima da linear para duas *threads* com uma eficiência de 94% para  $n = \{30, 40, 50, 60, 70\}$  ;
- O GPCA paralelo obteve uma aceleração de 1,38 vezes, sendo considerado sublinear para duas *threads*, culminando em uma eficiência de 69% para  $n = \{30, 40, 50, 60, 70\}$  ;
- O KTNS paralelo obteve uma aceleração de 7,12 vezes, sendo considerado superlinear para duas *threads*. Ademais, a eficiência foi de 356%;
- O KTNS tem uma acurácia média entre 76,96% e 99,84%.

**7**

**PRIMEIRO ESTUDO  
DE CASO**

**PRIMEIRO ESTUDO DE  
CASO**

**SSP uniforme**

## Visão Geral

- O SSP uniforme pertence à classe *NP-difícil* e possui instâncias recentes consideradas grandes
  - Somente uma máquina;
  - Tempo de troca idêntico;
  - Minimizar o número de trocas de ferramentas.
- Foram utilizadas 60 instâncias desenvolvidas por [Mecler, Subramanian e Vidal \(2021\)](#);
- O PT foi comparado com o *Hybrid Genetic Search* (HGS), o estado da arte para resolver o SSP uniforme proposto por [Mecler, Subramanian e Vidal \(2021\)](#).

## Resultados

- Em 7 dos 12 conjuntos de instâncias, o PT obteve resultados melhores ou similares que o HGS;
- Nenhuma distância percentual chegou a ultrapassar 0,16%;
- Em relação a média das 10 execuções, o PT supera o HGS em 9 dos 12 conjuntos de instâncias;
- Nos 3 conjuntos de instâncias restantes, a distância percentual não ultrapassou 0,09%.

## Resultados

- O PT demonstrou ser similar ao HGS em relação às melhores soluções geradas nos testes estatísticos;
- Na médias das soluções, o PT foi superior ao HGS nas instâncias com 50 e 60 tarefas e foi similar nas instâncias com 70 tarefas;
- Em relação ao tempo de execução, o PT foi claramente superior ao HGS, com uma redução de até 92,98%;
- Para uma das instâncias, o HGS demandou 7h e 30min enquanto o PT demandou 30min.



**SEGUNDO ESTUDO DE  
CASO**

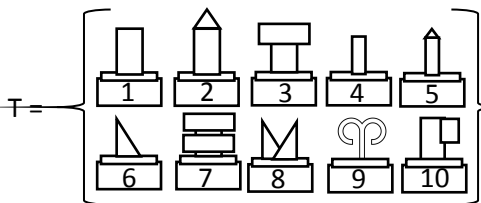
**SSP com máquinas paralelas idênticas com restrição de  
ferramentas (IPMTC)**

## Visão Geral

- O IPMTC pertence à classe *NP-difícil* e possui instâncias recentes consideradas grandes
  - Máquinas paralelas idênticas;
  - Minimizar o tempo de produção (*makespan*).
- Foram utilizadas 2880 instâncias desenvolvidas por [Beezão et al. \(2017\)](#);
- O PT foi comparado com uma hibridização do BRKGA com o VND, o algoritmo considerado o estado da arte para resolver o IPMTC proposto por [Soares e Carvalho \(2020\)](#).

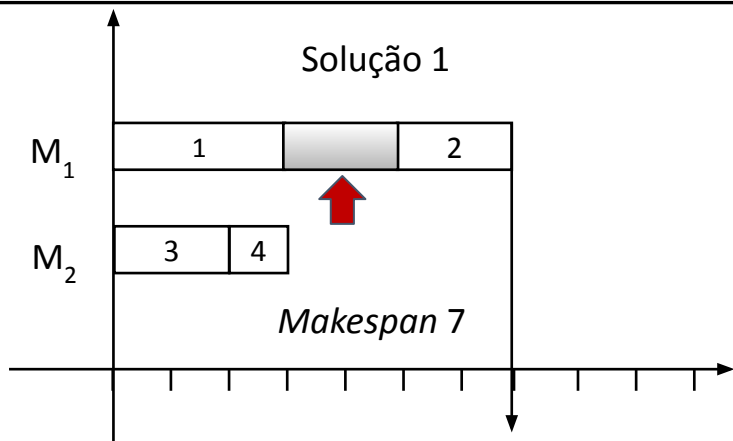
## IPMTC

Conjunto de ferramentas (T)

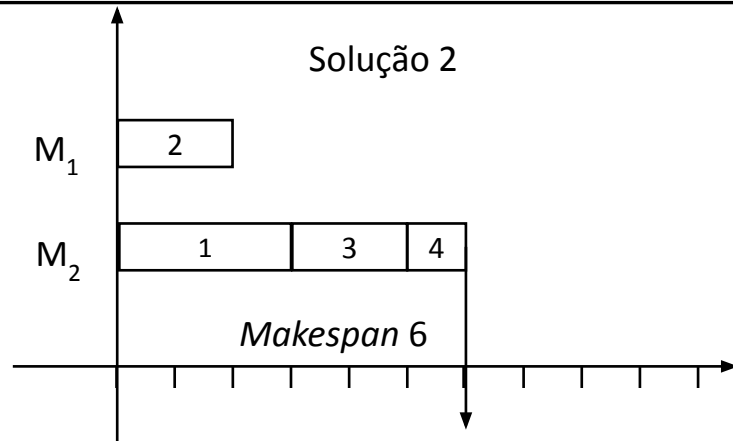


Tarefa	Ferramentas por tarefa ( $T_i$ )					Tempo ( $\tau_i$ )
1	2	3	5	10		3
2	1	5	7	8		2
3	9	3	5	10		2
4	2	3	5	10		1

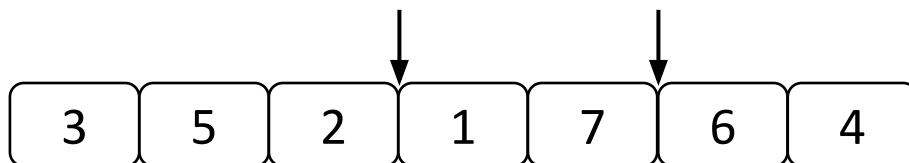
Solução 1



Solução 2



Vetor Principal:



Posição:

0 1 2 3 4 5 6

Vetor auxiliar início:



Vetor auxiliar fim:



Máquina 1 Máquina 2 Máquina 3

Máquina 1:

3 5 2

Máquina 2:

1 7

Máquina 3:

6 4

Máquina 1:

3 5

Máquina 2:

2 1 7

Máquina 3:

6 4

Máquina 1:

3 5

Máquina 2:

2 1

Máquina 3:

7 6 4

Máquina 1: 3 5 2

Máquina 2: 1 7

Máquina 3: 6 4

Máquina 1: 3 5 2

Máquina 2: 1 7 6

Máquina 3: 4

Máquina 1: 3 5 2 1

Máquina 2: 7 6

Máquina 3: 4

## Resultados

- O PT obteve resultados equivalentes ou melhores que o BRKGA para o conjunto de instâncias IPMTC-I, com 64 novas melhores soluções;
- Para o conjunto IPMTC-II o PT foi superior ao BRKGA em todos os 24 conjuntos de instâncias, variando *gap* entre -4,31% e -21,67%;
- Ao avaliar a média das 10 execuções, o PT foi superior ao BRKGA variando o *gap* entre -3,85% e -21,54%;
- Testes estatísticos comprovaram a superioridade do PT em relação ao BRKGA.

## Resultados

- O PT obteve 1274 novas melhores soluções para o IPMTC-II, o que representa 88% com conjunto;
- Em geral, o BRKGA apresenta valores de tempo de execução menores que PT;
- Este estudo de caso corrobora os indícios de qualidade do PT encontrados no primeiro estudo de caso.



**TERCEIRO ESTUDO DE  
CASO**

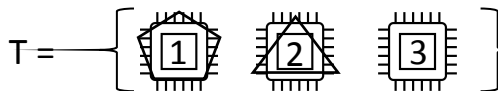
**SSP com máquinas paralelas com limitações de recursos  
(RCPMS)**

## Visão Geral

- O RCPMS pertence à classe *NP-difícil* e possui instâncias recentes consideradas grandes
  - Máquinas paralelas idênticas;
  - Minimizar o tempo de produção (*makespan*);
  - Compartilhamento de ferramentas.
- Foram utilizadas 270 instâncias desenvolvidas por [Soares e Carvalho \(2022\)](#);
- O PT foi comparado com o BRKGA hibridizado com procedimentos de busca local, o algoritmo considerado o estado da arte para resolver o RCPMS proposto por [Soares e Carvalho \(2022\)](#).

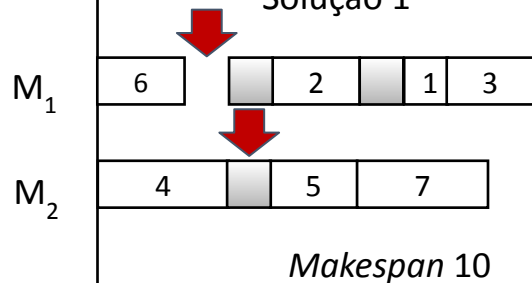
## RCPMS

Conjunto de ferramentas (T)

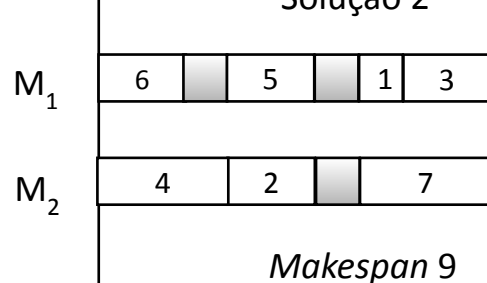


Ferramentas	Tarefa						
	1	2	3	4	5	6	7
1	✓		✓			✓	
2		✓		✓			
3		↑		↑	✓		✓
Tempo	1	2	2	3	2	2	3

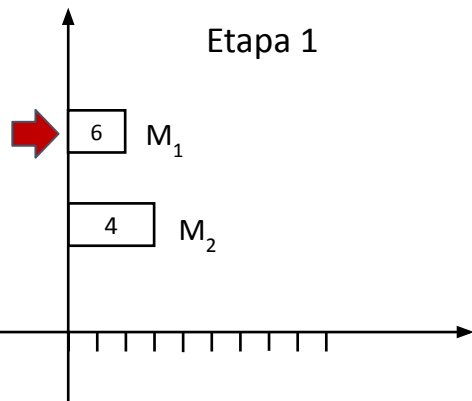
Solução 1



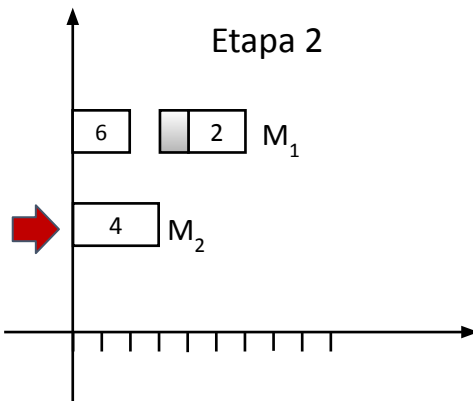
Solução 2



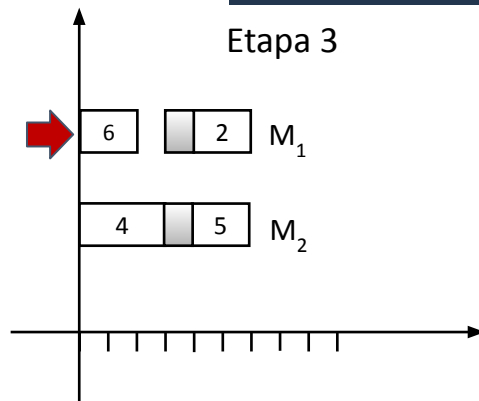
Etapa 1



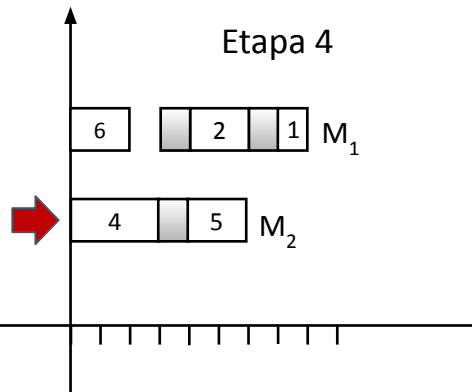
Etapa 2



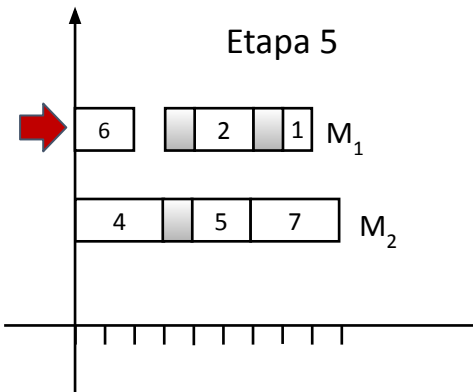
Etapa 3



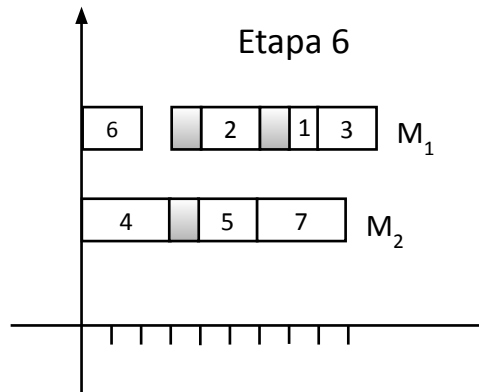
Etapa 4



Etapa 5



Etapa 6



## Resultados

- O PT igualou ou superou o BRKGA em relação às instâncias contidas no grupo RCPMS-I, com 4 novas melhores soluções;
- Para o conjunto de instâncias RCPMS-II, o PT obteve resultados iguais ou superiores ao BRKGA variando o *gap* entre -0,05% e -43,13%;
- Ao avaliar a média das 10 execuções, o PT foi melhor que o BRKGA em todos os 36 conjuntos de instâncias do RCPMS-II;
- Em relação ao tempo computacional, o BRKGA obteve menores valores em 27 dos 36 conjuntos de instâncias.

## Resultados

- Testes estatísticos confirmaram a superioridade do PT;
- Em nenhuma das 180 instâncias o PT obteve valores piores que o BRKGA para o conjunto RCPMS-II;
- O PT determinou 144 novas melhores soluções;
- O PT apresenta um desempenho notável nos três cenários abordados.

### Objetivos específicos

1. Elaborar uma rigorosa revisão sistemática da literatura relacionada à metaheurísticas paralelas baseadas em trajetória e identificar suas lacunas.
2. Gerar embasamento teórico sobre arquiteturas paralelas e distribuídas.
3. Projetar e implementar uma metaheurística paralela em CPU.
4. Avaliar o desempenho do método proposto em casos de estudo.
5. Revisitar as funções de avaliação dos estudos de caso.
6. Criar e disponibilizar uma API contendo o método proposto.
7. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos.

# **API DO REVENIMENTO PARALELO**

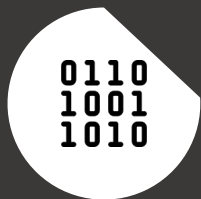
**Definições e funcionamento**



## Análise

- Necessidade de se popularizar as implementações paralelas;
- Difundir a utilização do PT na resolução de problemas de otimização;
- A API proposta apresenta uma interface de conexão simples e eficiente;
- Após a definição dos dados do problema, são necessárias somente três linhas para se executar o PT;
- Disponível para *download*\* sob a licença Creative Commons BY-NC (CC BY-NC).

\*<https://github.com/ALBA-CODES/PTAPI/>



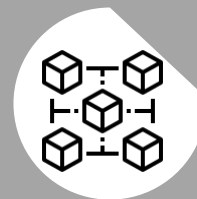
01

codificação e  
decodificação  
da solução



02

Codificação dos  
componentes  
dependentes ao  
problema (avaliação  
e movimento)



03

conexão com o  
PT

## Estratégias de definição das temperaturas

- Distribuição inicial das temperaturas
  - Linear;
  - Linear-inverso;
  - Progressão geométrica;
  - Exponencial.
- Estratégia dinâmicas de atualização das temperaturas
  - Igualar as taxas de troca das temperaturas adjacentes;
  - Igualar a taxa de troca a 23%;
  - *Feedback-optimized*.

## Estratégias Implementadas

- Movimentos
  - 2-opt;
  - 2-swap;
  - Inserção aleatória.
- Solução inicial
  - Aleatória.

### Objetivos específicos

1. Elaborar uma rigorosa revisão sistemática da literatura relacionada à metaheurísticas paralelas baseadas em trajetória e identificar suas lacunas.
2. Gerar embasamento teórico sobre arquiteturas paralelas e distribuídas.
3. Projetar e implementar uma metaheurística paralela em CPU.
4. Avaliar o desempenho do método proposto em casos de estudo.
5. Revisitar as funções de avaliação dos estudos de caso.
6. Criar e disponibilizar uma API contendo o método proposto.
7. Publicar os resultados deste trabalho de pesquisa em periódicos e eventos.

# Publicações

**Publicações aceitas e submetidas**

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. Systematic literature review on parallel trajectory-based metaheuristics. ACM Comput. Surv., Association for Computing Machinery, New York, NY, USA, jul 2022.

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. Revisitando o algoritmo Keep Tools Needed Soonest: implementações seriais e paralelas. In: ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2023, São José dos Campos.

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. Revenimento paralelo aplicado ao sequenciamento da produção em sistemas de manufatura flexível. In: ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2024, Fortaleza.

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. Revenimento paralelo aplicado ao sequenciamento de tarefas em máquinas flexíveis paralelas idênticas. In: ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2024, Fortaleza.

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. Revenimento paralelo aplicado ao sequenciamento em máquinas flexíveis paralelas com recursos compartilhados. In: ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2024, Fortaleza.

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. Revisiting the Parallel Tempering Algorithm: High-Performance Computing and Applications in Operations Research. Available at SSRN: <https://ssrn.com/abstract=4756904> or <http://dx.doi.org/10.2139/ssrn.4756904>

ALMEIDA, A. L. B.; LIMA, J. d. C.; CARVALHO, M. A. M. On Serial and Parallel Evaluation Functions for the Job Sequencing and Tool Switching Problems. Available at SSRN: <https://ssrn.com/abstract=4821662> or <http://dx.doi.org/10.2139/ssrn.4821662>



# **CONCLUSÃO**

**Contribuições e achados**

## Realizações

- O presente trabalho de pesquisa apresentou uma revisão sistemática da literatura, explorando lacunas e tendências;
- Esse estudo forneceu um embasamento teórico sobre otimização e arquiteturas paralelas;
- Foi implementada uma metaheurística paralela baseada no método revenimento paralelo (PT);
- Uma API foi desenvolvida com o objetivo de popularizar o PT.

## Realizações

- O PT foi avaliado em três estudos de caso distintos e baseados em problemas da classe *NP-difícil*;
- No primeiro estudo de caso, o PT obteve resultados similares ao estado da arte, porém em um tempo significativamente menor;
- No segundo estudo de caso, o PT superou o estado da arte, com uma redução de até 22%;
- No terceiro estudo de caso, o PT obteve um número significativo de novas melhores soluções, apresentando uma redução de até 43,13%.

# **TRABALHOS FUTUROS**

**Tópicos relevantes**

## Tópicos

- Um desdobramento do PT em contextos permutacionais diversos se apresenta como uma extensão natural deste projeto de pesquisa;
- Examinar variações do SSP em vários cenários não abordados durante a realização desta tese;
- Implementação paralela do PT em novas plataformas.



UFOP

Universidade Federal  
de Ouro Preto



*Obrigado!*