

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**PARALELIZAÇÃO DA METAHEURÍSTICA BUSCA TABU PARA SOLUCIONAR O  
PROBLEMA DE ROTEAMENTO DE CABOS EM PARQUES EÓLICOS**

Lara Lorrany Felipe Ferigatto

GOIÂNIA  
2023

Lara Lorrany Felipe Ferigatto

**PARALELIZAÇÃO DA METAHEURÍSTICA BUSCA TABU PARA SOLUCIONAR O  
PROBLEMA DE ROTEAMENTO DE CABOS EM PARQUES EÓLICOS**

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

GOIÂNIA  
2023

Lara Lorrany Felipe Ferigatto

**PARALELIZAÇÃO DA METAHEURÍSTICA BUSCA TABU PARA SOLUCIONAR O  
PROBLEMA DE ROTEAMENTO DE CABOS EM PARQUES EÓLICOS**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do grau de Bacharel em Ciência da Computação e aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, em 20 de junho de 2023.

---

Profa. Ma. Ludmilla Reis Pinheiro dos Santos  
Coordenadora de Trabalho de Conclusão de  
Curso

Banca examinadora:

---

Orientador: Prof. Me. Alexandre Ribeiro

---

Prof<sup>a</sup>. Dra. Maria José Pereira Dantas

---

Prof. Me. Max Gontijo de Oliveira

GOIÂNIA  
2023

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus por possibilitar que tudo acontecesse.

A minha mãe e minha avó que foram presentes por toda a minha vida me motivando a continuar.

Ao meu professor, orientador e amigo Alexandre Ribeiro, por tudo que fez por mim em todos esses anos, não me deixando desistir e me apoiando nos momentos mais difíceis, agradeço por me escutar durante várias horas e me orientar, por se preocupar e por me ajudar antes mesmo de eu pedir ajuda.

A todos os meus amigos que conheci durante o curso. Em especial, ao João Augusto, Jovânio Júnior e Omar. Vocês foram pessoas muito importantes para que eu conseguisse concluir essa etapa.

A Dona Lurdes e Dona Rosilda que me receberam na sua casa, cuidaram de mim e fizeram possível eu me dedicar exclusivamente a faculdade por um tempo.

A todos os professores da universidade. Obrigado por todos os ensinamentos e por continuarem passando seu conhecimento e experiências para todos.

## RESUMO

Foi registrado nos últimos anos a maior taxa de consumo de energia no mundo, com esse aumento, para atender a essa demanda, a produção de energia a carvão aumentou o que consequentemente aumentou os índices de emissão de  $CO_2$ . Pensando nisso, usinas livres de poluição tem recebido atenção de investidores. A energia eólica se destaca por ser uma das mais limpas. Neste trabalho de conclusão de curso foi abordado o Problema de Roteamento de Cabos em Parques Eólicos (WFCRP) com vários tipos de cabos e uma única subestação com limite de conexões. O objetivo do problema é encontrar a topologia de menor custo para ligar todas as turbinas a subestação a fim de coletar a energia produzida por cada turbina. Ao longo do trabalho foi proposto um algoritmo heurístico baseado na metaheurística Busca Tabu, juntamente, com técnicas de paralelização de busca local. O algoritmo foi executado uma vez para cada instância dada, tendo como solução inicial uma solução encontrada pela metaheurística simulated annealing, levando no máximo 2 minutos para fornecer a melhor solução encontrada. A busca tabu paralelizada apresentou uma melhora de 11% em média dos resultados obtidos por OLIVEIRA, RIBEIRO (2020), porem não conseguiu em nenhuma instância alcançar a melhor solução ficando em média 21% acima da melhor solução conhecida.

**Palavras-chave:** *Roteamento de cabos. Metaheurísticas. Metaheurística paralela.*

## ABSTRACT

The highest rate of energy consumption in the world has been registered in recent years, with this increase, to meet this demand, the production of coal-fired energy has increased, which consequently increased the emission rates of  $CO_2$ . Thinking about it, clean energy sources receive investors' attention. Wind energy stands out for being one of the cleanest. In this term paper the Wind Farm Cable Routing Problem (WFCRP) with several types of cables and a single substation with limited connections, was studied. The objective of the problem is to find the lowest cost topology to connect all the turbines to the substation in order to collect the energy produced by each turbine. Throughout the work, a heuristic algorithm based on the Tabu Search metaheuristic was proposed, along with local search parallelization techniques. The algorithm was executed once for each given instance, having as initial solution the best solution found by OLIVEIRA, RIBEIRO (2020), taking a maximum of 2 minutes to provide the best solution found. The parallelized tabu search showed an improvement of 11% on average in the results obtained by OLIVEIRA, RIBEIRO (2020), but in no instance was it able to reach the best solution, being on average 21% above the best known solution.

**Keywords:** *Cable routing. Metaheuristics. Parallel Metaheuristics*

## LISTA DE FIGURAS

Figura 1 - Topologia do parque eólico Barrow.....	12
Figura 2 - Topologia do parque eólico Barrow utilizando cabos diferentes .....	13
Figura 3 - Ótimos locais e ótimos globais .....	16
Figura 4 - Processo single thread e multithread.....	20
Figura 5 - Turbinas conectadas à um nó de Steiner .....	29
Figura 6 - Aplicação do movimento de troca de conexão .....	31
Figura 7 - Aplicação do movimento de troca de cabo .....	31
Figura 8 - Geração de Soluções usando threads.....	32
Figura 9 - Gráfico de Resultados .....	38

## LISTA DE TABELAS

Tabela 1 - Parques eólicos utilizados .....	34
Tabela 2 - Conjunto de cabos e número de turbinas suportadas por cada tipo de cabo. .....	35
Tabela 3 - Instâncias utilizadas neste trabalho.....	35
Tabela 4 - Resultados obtidos .....	37



## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>10</b>
1.1 Objetivos .....	14
1.2 Organização textual .....	14
<b>2. CONCEITOS PRELIMINARES .....</b>	<b>15</b>
2.1 Problemas de Otimização Combinatória .....	15
2.2 Abordagens Heurísticas .....	15
2.3 Busca Tabu .....	17
2.4 Programação Paralela .....	19
2.5 Programação Multicore .....	21
2.6 Paralelização de Metaheurísticas .....	22
<b>3. O PROBLEMA WFCRP .....</b>	<b>24</b>
3.1 Definição Formal .....	25
3.2 Formulação .....	26
<b>4. TRABALHOS RELACIONADOS .....</b>	<b>28</b>
<b>5. ABORDAGENS UTILIZADAS .....</b>	<b>30</b>
5.1 Busca Tabu Paralelizada .....	30
5.1.1 Representação das Soluções .....	30
5.2.2 Estrutura de vizinhança .....	30
5.2.1 Solução Inicial .....	32
<b>6. EXPERIMENTOS COMPUTACIONAIS .....</b>	<b>34</b>
6.1 Instâncias .....	34
6.2 Resultados .....	36
<b>7 CONCLUSÃO .....</b>	<b>39</b>
<b>REFERÊNCIAS .....</b>	<b>41</b>

## 1 INTRODUÇÃO

Foi registrado pela *International Energy Agency* (IEA) no ano de 2021, que a procura global de eletricidade teve um aumento de 1200TWh (5%), o maior aumento ano a ano na demanda global de eletricidade já registrado. Como consequência houve um aumento de 700Mt (milhões de toneladas) das emissões de  $CO_2$  pelo setor elétrico, atingindo um recorde histórico de mais de 14Gt (bilhões de toneladas). Isso foi impulsionado, principalmente, por um forte aumento na geração de eletricidade a carvão em comparação com o ano anterior (IEA, 2021).

A geração de eletricidade a carvão atendeu a cerca da metade do aumento da demanda global de eletricidade em 2021, sendo cerca de 32% do total de eletricidade gerada. A geração de energia a carvão foi impulsionada pelos altos preços do gás natural e os baixos custos de operação das usinas a carvão existente nos Estados Unidos. Esse aumento no consumo de energia e na emissão de  $CO_2$  causam grandes impactos ambientais e vão contra os esforços propostos pela meta Net Zero (cenário de emissão zero) (IEA, 2022). Net zero (*net zero carbon emissions*) é o compromisso feito entre empresas e governos de reduzir as emissões de gases de efeito estufa na atmosfera.

A energia eólica possui baixo impacto ambiental (CAZZARO et al., 2019), enquanto as usinas que utilizam carvão, gás ou petróleo são extremamente poluentes (SAIDUR et al., 2011). Existem outros tipos de energia limpa como, por exemplo, as usinas nucleares, mas há muita controvérsia, pois o processo de geração desse tipo de energia resulta na criação de substâncias tóxicas e alto risco em caso de acidentes. Hidrelétricas também são uma opção de energia limpa, porém como utilizam água como combustível, é necessária a criação de barragens que bloqueiam o curso de rios, podendo inundar áreas inteiras em caso de acidentes (OLIVEIRA; RIBEIRO, 2020).

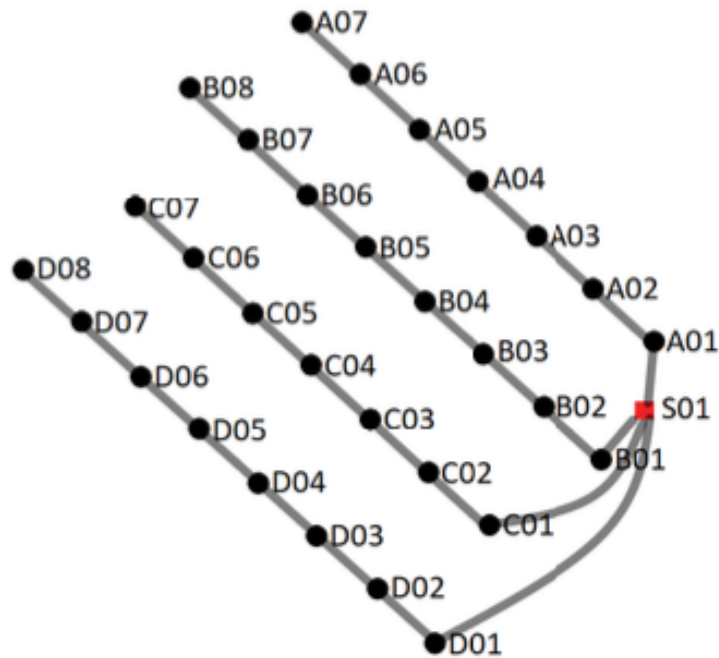
Em 2021 a produção de energia eólica cresceu 270TWh (aumento de 17%), um crescimento 55% maior do que em 2020, sendo uma das principais responsáveis por um recorde histórico na geração total de eletricidade renovável. Esse crescimento foi possível graças a um aumento sem precedentes nas adições de capacidade eólica, que atingiu 113GW em 2020. No entanto, ainda são necessários esforços maiores para alcançar os objetivos de emissão zero em 2050, o que considera que até 2030 é necessário produzir cerca de 7900TWh de energia eólica, para isso seria necessário

aumenta a capacidade média anual para quase 250GW, mais que o dobro do crescimento recorde de 2020. Para atingir esse nível de crescimento, as áreas mais importantes de melhoria seria facilitar a permissão para construção de parques eólicos em terra e a redução de custos para construção de parques eólicos marítimos (IEA, 2022).

O investimento para construir um parque eólico em terra está entre os mais baratos, já o investimento para construir um parque eólico no mar está entre os mais caros. Os gastos com infraestrutura elétrica na construção de parques eólicos marítimos variam entre 15% e 30% dos custos iniciais, 12% a 26% somente com os sistemas de coleta e transmissão de energia (OLIVEIRA; RIBEIRO, 2020). Sendo de suma importância para alcançar os objetivos do Net Zero, otimizar a forma com que os cabos são conectados para coletar energia, dando origem ao Problema de Roteamento de Cabos em Parques Eólicos (WFCRP - *Wind Farm Cable Routing Problem*).

O WFCRP é enunciado como, dado um conjunto de turbinas, com suas localizações já definidas no plano, encontrar o menor custo para conectar todas as turbinas a um ponto de coleta de energia, chamado subestação, sendo que as turbinas podem ser conectadas em séries, ou seja, uma turbina conecta a outra turbina, que pode estar conectada a outra, e assim sucessivamente até a subestação. A Figura 1 mostra a topologia de um parque real, Barrow, em que a subestação é representada por um quadrado, enquanto que os demais pontos são as turbinas.

Figura 1 - Topologia do parque eólico Barrow

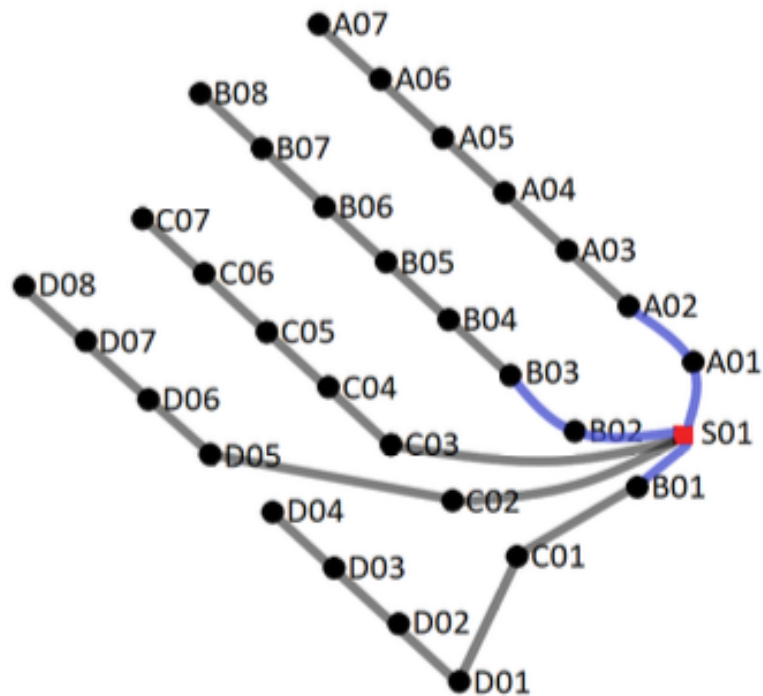


Fonte: OLIVEIRA, RIBEIRO (2018)

Uma solicitação comum das empresas é que os cabos não se cruzem, embora teoricamente seja possível deixar os cabos se cruzarem, existem dois motivos para essa restrição: primeiro, cabos de alta tensão geram calor, o que implica que dois cabos cruzados teriam que ser isolados. Em segundo lugar, os cabos são enterrados no fundo do mar para sua proteção. Se um cabo que está enterrado abaixo de outro precisar de manutenção, seria necessário desenterrar ambos os cabos para substituir o cabo com defeito, resultando em custos mais altos de manutenção (BAUER; LYSGAARD, 2015).

Podem ser utilizados um ou vários tipos de cabo para conectar parques eólicos. Na Figura 1 é possível ver o parque Barrow utilizando apenas um tipo de cabo. A Figura 2 apresenta uma outra possível topologia para o mesmo parque eólico, porém utilizando dois tipos de cabos diferentes, marcados cada um com uma cor diferente.

Figura 2 - Topologia do parque eólico Barrow utilizando cabos diferentes



Fonte: OLIVEIRA, RIBEIRO (2020)

O objetivo deste trabalho é minimizar os custos de cabeamento em parques eólicos utilizando a busca Tabu (BT) juntamente com técnicas de paralelização. BT é um método heurístico originalmente proposto por Glover em 1986, com o objetivo de permitir que os métodos de busca local superassem os ótimos locais. Glover não considerava a BT uma heurística, mas sim uma metaheurística, ou seja, uma estratégia geral para guiar e controlar heurísticas internas especificamente adaptadas ao problema em questão. Centenas de trabalhos nos últimos 30 anos apresentam aplicações da BT para problemas de otimização combinatória, onde em vários casos, o método fornece soluções muito próximas da ótima, estando entre os mais eficazes para esse tipo de problema (GENDREAU; POTVIN, 2019).

A Paralelização da Busca Tabu pode ser usada para acelerar as fases computacionais extensivas dessa metaheurística, e por isso foi adotada por este trabalho, visando aumentar e acelerar a etapa de busca local.

## **1.1 Objetivos**

O objetivo deste trabalho de conclusão de curso é apresentar uma abordagem metaheurística paralelizada para minimizar os custos do cabeamento em parque eólicos marítimos. Os resultados obtidos foram comparados com os resultados de OLIVEIRA, RIBEIRO (2020) e FISCHETTI, PISINGER (2018).

## **1.2 Organização textual**

O restante deste trabalho está textualmente assim organizado: No capítulo 2 são apresentados os conceitos preliminares sobre problemas de otimização combinatória, abordagens heurísticas, busca tabu, programação paralela, programação multicore e paralelização de metaheurística; o capítulo 3 é composto pela definição formal do problema e sua formulação matemática; o capítulo 4 apresenta os principais e mais recentes trabalhos relacionados ao problema WFCRP; o capítulo 5 descreve as abordagens utilizadas para solucionar o WFCRP; o capítulo 6 apresenta os resultados dos experimentos computacionais realizados; por fim o capítulo 7 conclui com algumas considerações finais.

## 2. CONCEITOS PRELIMINARES

Neste capítulo são apresentados os fundamentos teóricos necessários para o entendimento do trabalho. Na Seção 2.1 são apresentados os conceitos de otimização combinatória, necessário para a definição formal do problema. Em seguida na seção 2.2 e 2.3 contêm os conceitos base, sobre abordagens heurística e Busca Tabu. Na seção 2.4 e 2.5 são apresentados os conceitos sobre programação paralela e programação paralela necessários para o entendimento da seção 2.5, que aborda conceitos de metaheurística paralela, método usado neste trabalho.

### 2.1 Problemas de Otimização Combinatória

Na ciência da computação o segmento responsável por estudar problemas que têm por objetivo dentro de um conjunto discreto encontrar uma combinação de elementos que gere a melhor solução possível, é chamado de otimização combinatória (PAPADIMITRIOU; STEIGLITZ, 1998). De acordo com CARVALHO et al. (2001) um problema de otimização combinatória tem três elementos principais: um conjunto de instâncias, um conjunto  $sol(I)$  de soluções viáveis para cada instancia  $I$  e uma função objetivo  $val(S)$  que atribui um valor a cada solução viável  $S$ . Uma solução  $S$  é considerada viável caso ela atenda a todas as restrições do problema.

Problemas ditos de minimização estão interessados nas soluções viáveis de valor mínimo e problemas de maximização estão interessados nas soluções viáveis de valor máximo. Sendo assim, quando o problema for de minimização, uma solução  $S^* \in sol(I)$  é ótima se  $val(S^*) \leq val(S)$ , para todo  $S \in sol(I)$ , da mesma forma quando o problema se tratar de um problema de maximização,  $S^*$  é uma solução ótima se  $val(S^*) \geq val(S)$  para todo  $S \in sol(I)$ .

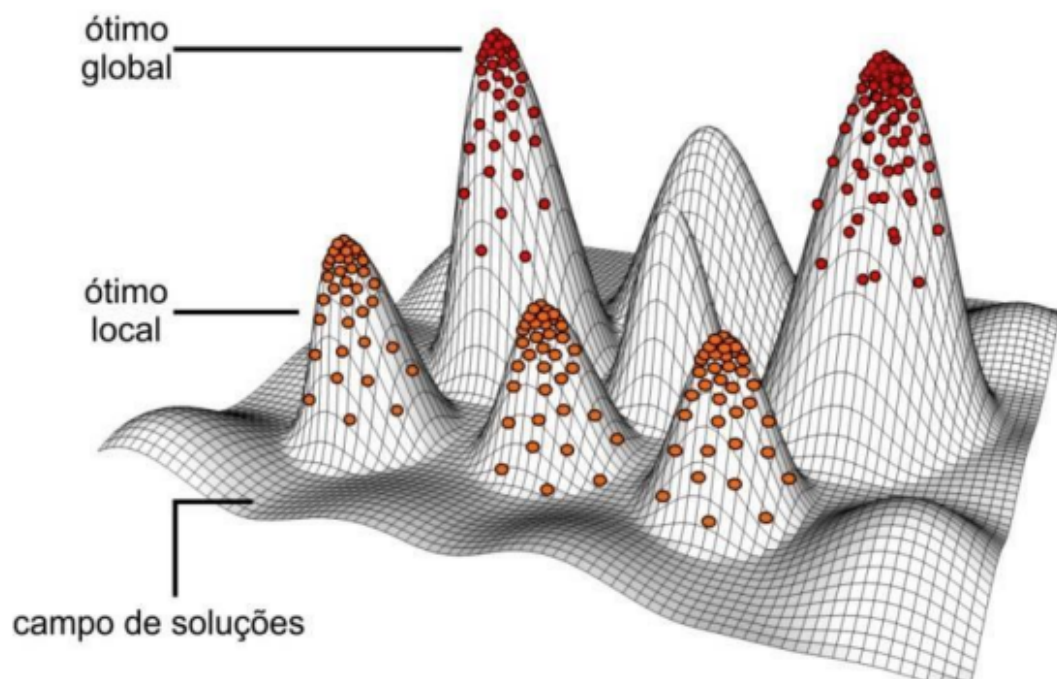
### 2.2 Abordagens Heurísticas

Abordagens heurísticas são técnicas de solução aproximada que têm sido usadas desde o início da pesquisa operacional, com o objetivo de lidar com problemas combinatórios difíceis. No início dos anos 70 com o desenvolvimento da teoria da complexidade, ficou evidente que como a maioria dos problemas combinatórios eram NP-

difícil, havia pouca probabilidade de encontrar procedimentos eficientes de soluções exatas aplicáveis a eles. Viu-se o importante papel das heurísticas para resolver problemas combinatórios encontrados em problemas reais, e que precisavam ser resolvidos, qualquer que fosse sua complexidade (GENDREAU; POTVIN, 2019).

Segundo GENDREAU, POTVIN (2019) muitas abordagens diferentes foram propostas e experimentadas, as mais populares eram baseadas em técnicas de aprimoramento de Pesquisa Local (PL). PL, de forma resumida, consiste em um procedimento de busca iterativa que parte de uma solução viável inicial e a melhora, progressivamente, aplicando modificações locais. A cada iteração a busca move-se para uma solução viável que difere ligeiramente da atual. Quando as transformações aplicadas não geram nenhuma solução melhor do que a já encontrada a busca se encerra. Essa solução é chamada de ótimo local, ou seja, a melhor solução dentro do espaço de busca local. Uma limitação do método PL é que esse ótimo local, normalmente, está bem distante do ótimo global, como pode ser visto na Figura 3, onde os círculos representam as soluções, o espaço de busca local de cada solução é onde os círculos estão contidos.

Figura 3 - Ótimos locais e ótimos globais



Fonte: GATE, 2018



Quando se tem uma heurística geral o suficiente para que possa ser modelada a específicos problemas ela é chamada de metaheurística. As metaheurísticas são de caráter geral e, comumente, o que as diferencia são os mecanismos usados para evitar que fiquem presas em ótimos locais, distantes do ótimo global (SOUZA, 2020).

### 2.3 Busca Tabu

O princípio básico da BT é continuar a busca local mesmo depois de encontrar um ótimo local, permitindo movimentos que não melhoram a solução atual. Para prevenir o retorno a soluções já visitadas e que dariam origem a um conjunto de soluções já avaliadas, é montado uma lista de movimentos, chamada de lista tabu. Uma BT básica poderia ser vista simplesmente como a combinação de um método de busca local com memorização de curto prazo (lista tabu) (GENDREAU; POTVIN, 2019).

Os elementos básicos da BT são o seu espaço de busca, a sua estrutura de vizinhança e os tabus. O espaço de busca é o conjunto de todas as soluções possíveis que podem ser consideradas durante uma busca. Soluções possíveis são aquelas que satisfazem todas as restrições do problema ao qual a BT está sendo aplicada, porém, dependendo do problema, pode ser necessário ou desejável expandir o espaço de busca para soluções inviáveis que possam, em iterações futuras, convergir em soluções viáveis (GENDREAU; POTVIN, 2019).

A definição da estrutura de vizinhança da BT está ligada à definição do espaço de busca. A cada iteração da BT as mudanças aplicadas à solução atual, chamada de  $S$ , geram um conjunto de soluções vizinhas a  $S$  no espaço de busca, denotado  $N(S)$ . Assim, formalmente,  $N(S)$  é um subconjunto do espaço de busca, constituído por todas as soluções geradas a partir de  $S$  (GENDREAU; POTVIN, 2019).

O significado da palavra tabu na Língua Portuguesa de acordo com o dicionário Aurélio é "Proibição absoluta e inviolável, imposta por preceitos religiosos, costumes, tradições ou convenções sociais, que determina a exclusão, a interdição ou a restrição de determinados atos, objetos, palavras, lugares, pessoas, etc." ou "Algo que é considerado impuro, profano ou perigoso e, portanto, deve ser evitado ou tratado com cautela". Esse significado define bem a própria função dos tabus na BT.

Os Tabus são usados para evitar ciclos e também evitar que a BT fique presa em um ótimo local, por meio de movimentos que não melhoram a solução atual. Isso é feito proibindo movimentos tabus que retornariam para movimentos recentemente visitados, o que daria origem uma vizinhança já avaliada. Os Tabus são armazenados em uma memória de curto prazo da pesquisa (lista tabu), normalmente, essa lista tem um tamanho fixo. Existem várias possibilidades em relação a qual informação deve ser armazenada. Pode-se armazenar soluções completas, porém isso exigiria muito armazenamento e verificar se uma solução é tabu tornaria o custo computacional elevado, portanto, é raramente usado. Os tabus mais utilizados são os que registram as últimas mudanças realizadas na solução atual, ocasionando na proibição de mudanças reversas (GENDREAU; POTVIN, 2019).

Podem ser usadas múltiplas listas tabus, quando, por exemplo, diferentes tipos de movimentos são usados para gerar uma vizinhança, assim mantem-se uma lista tabu para cada tipo de movimento. As listas tabus normalmente são implementadas utilizando uma lista circular de tamanho fixo (GENDREAU; POTVIN, 2019).

Apesar dos tabus serem úteis para evitar ciclos e estagnação em ótimos locais, eles podem, por outro lado, impedir movimentos desejados onde não há perigo de ciclagem ou levar a uma estagnação geral do processo de busca. Nesses casos é necessário utilizar métodos para ignorar o fato do movimento ser tabu. Estes métodos são chamados de critério de aspiração. O critério de aspiração mais utilizado em BT, consiste em permitir o movimento tabu caso a solução tenha um valor objetivo melhor do que o da solução atual mais conhecida (GENDREAU; POTVIN, 2019).

Como mostrado no Algoritmo 1, de forma geral, a BT explora conjuntos de soluções de um problema iterativamente, repetidas vezes, indo da solução  $S$  para  $s' \in N(s)$  (linha 11) onde  $N(s)$  é o conjunto de soluções vizinhas a  $S$ . Os movimentos de  $S$  para  $s'$  são feitos em busca de melhorar a melhor solução conhecida ( $S^*$ ). A avaliação de uma solução é feita pela função objetivo  $f(s)$  (linha 14). Após gerar o conjunto  $N(s)$ , cada solução é avaliada para determinar se é viável, ou seja, atende a todas as restrições do problema, gerando o conjunto  $\tilde{N}(s)$  de forma que se possa escolher a melhor solução de  $\tilde{N}(s)$  e mover-se para ela, assim que se move para uma nova solução é necessário

atualizar a lista tabu com o movimento feito. Caso a solução  $s'$  seja melhor  $S^*$ ,  $S^*$  é atualizado (linha 15). A busca termina quando o critério de parada é satisfeito.

Os Critérios de parada da BT mais usados são:

- Número fixo de iterações
- Número fixo de tempo de CPU
- Número fixo de iterações sem melhora
- Quando a função objetivo chega em um determinado valor

O Algoritmo 1 mostra um pseudocódigo da BT.

#### **Algoritmo 1: BT**

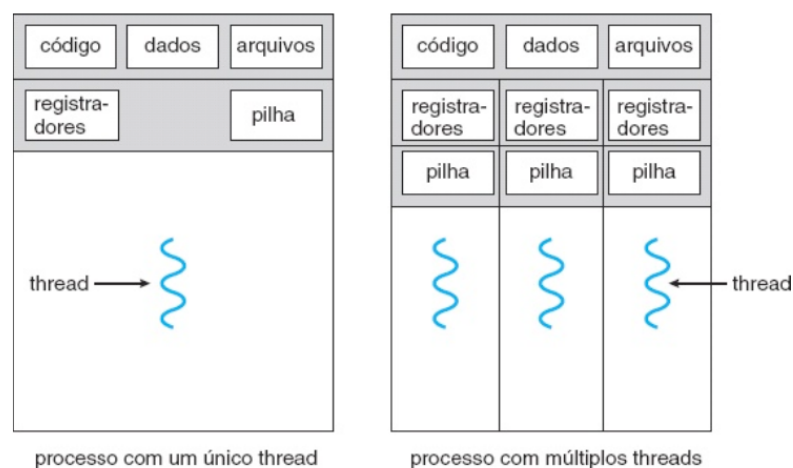
```
1  Entrada:
2  S - Solução atual
3   $S^*$  - Melhor Solução conhecida
4  N(S) - a vizinhança de S
5   $\tilde{N}(S)$  - Conjunto de Soluções viáveis de N(S)-Não tabus ou aceitas pelo critério de aspiração
6  T - Lista tabu
7  Inicialização:
8   $S'$  - Uma solução inicial válida
9   $S \leftarrow s'$ ,  $S^* \leftarrow S'$ , T  $\leftarrow$  vazio
10 enquanto (critério de parada não for satisfeito) faça:
11     Gerar o conjunto  $\tilde{N}(S)$  a partir N(S)
12      $S \leftarrow s'$  onde  $s'$  é a melhor solução do conjunto  $\tilde{N}(S)$ 
13     Atualizar a Lista Tabu T
14     se ( $f(s') < f(S^*)$ ) então
15          $S^* \leftarrow S'$ 
16     fim-se
17 fim-enquanto
18 retorne  $S^*$ 
```

## **2.4 Programação Paralela**

Segundo SILBERSCHAT, GALVIN, GAGNE (2015) um processo é um programa em execução com um único *thread* de controle. Basicamente, todos os sistemas operacionais modernos fornecem recursos que habilitam um processo a conter múltiplos *threads* de controle.

*Thread* é definido por SILBERSCHAT, GALVIN, GAGNE (2015) como uma unidade básica de utilização da CPU, sendo composto por um ID de *thread*, um contador de programa, um conjunto de registradores e uma pilha. *Threads* pertencentes ao mesmo processo compartilham entre si, sua seção de código, seção de dados e outros recursos do sistema operacional. Caso um processo tenha múltiplos *threads* de controle, isso significa que ele pode executar mais de uma tarefa ao mesmo tempo. A Figura 4 mostra a diferença entre um processo com um único *thread* e um processo com *multithread*.

Figura 4 - Processo single thread e multithread



Fonte: (SILBERSCHAT; GALVIN; GAGNE, 2015).

Atualmente, a maioria das aplicações de software são executadas em computadores *multithread*, sendo que, normalmente, uma aplicação é implementada como um processo separado com diversos *threads* de controle. Por exemplo, um processador de texto pode ter um *thread* para exibir elementos gráficos, outro para responder ao uso do teclado pelo usuário, e um terceiro para executar a verificação ortográfica e gramatical em *background*. Aplicações também podem ser projetadas para executar diversas tarefas em paralelo, com a finalidade de melhorar a capacidade de processamento em sistemas *multicore* (SILBERSCHAT; GALVIN; GAGNE, 2015).

A maioria, se não todos, dos *kernels* dos sistemas operacionais atuais, são *multithreaded*, isso significa que múltiplos *threads* operam no kernel e cada *thread* executa uma tarefa específica, como o gerenciamento de dispositivos, o gerenciamento de memória ou a manipulação de interrupções.

Segundo os SILBERSCHAT, GALVIN, GAGNE (2015) benefícios do uso de programação com *multithreaded* são divididos em quatro categorias principais:

- Capacidade de resposta. Permite que um programa continue executando, ainda que partes dele esteja bloqueada ou executando outra operação;
- Compartilhamento de recursos. Por padrão os *threads* compartilham a memória e os recursos do processo ao qual pertencem, o principal benefício de compartilhar código e dados é a possibilidade de vários *threads* executarem tarefas diferentes dentro do mesmo espaço de endereçamento.
- Economia. Alocação de recurso e criação de processo são operações de alto custo para o SO. Como os *threads* compartilham os recursos do processo ao qual pertencem, torna-se mais barato criar *threads* e permutar seus contextos.
- Escalabilidade. Em arquiteturas multiprocessadoras os benefícios são ainda maiores, pois nessas arquiteturas os *threads* podem ser executados em paralelo em diferentes núcleos do processador.

## 2.5 Programação Multicore

Com a necessidade de melhorar o desempenho da computação, sistemas com uma única CPU evoluíram para sistemas com múltiplas CPUs. Atualmente, múltiplos núcleos de computação no mesmo chip foram inseridos nos projetos de sistemas de computadores. Para o SO, cada núcleo é visto como um processador separado. O uso de múltiplos *threads* na programação oferece a possibilidade de uso mais eficiente desses múltiplos núcleos de computação e o aumento da concorrência. Considerando um sistema com um único núcleo, rodando uma aplicação com quatro *threads*, nesse caso a concorrência significa que os threads serão intercalados por meio de tempo, uma vez que o núcleo consegue executar apenas um thread de cada vez. Já em um sistema *multicore*, a concorrência significará que os *threads* podem ser executadas em paralelo, pois o sistema poderá atribuir um núcleo a cada *thread* (SILBERSCHAT; GALVIN; GAGNE, 2015).

Aqui vemos a diferença entre paralelismo e concorrência. Consideramos um sistema paralelo quando ele pode executar mais de uma tarefa simultaneamente, e um sistema é dito concorrente quando ele dá suporte a mais de uma tarefa permitindo que elas

progridam. Havendo assim concorrência sem paralelismo (SILBERSCHAT; GALVIN; GAGNE, 2015).

## **2.6 Paralelização de Metaheurísticas**

O desenvolvimento de metaheurísticas usando a computação paralela tem dois objetivos principais. O primeiro, assim como os esforços de desenvolvimento paralelo, é resolver mais rapidamente problemas com instâncias maiores, ou seja, alcançar soluções para problemas com instâncias maiores do que as alcançáveis em métodos sequenciais, fazendo isso em um tempo computacionalmente razoável. O segundo objetivo diz respeito à chamada robustez do método, ou seja, sua capacidade de oferecer um alto nível de desempenho em uma ampla variedade de configurações de problemas e características de instância (GENDREAU; POTVIN, 2019).

GENDREAU, POTVIN (2019) Adotam três dimensões de classificação para descrever as diferentes estratégias de paralelização de metaheurística. As três dimensões da classificação definem como o processo global de solução de problemas é controlado, como as informações são trocadas entre os processos e como são, eventualmente, criadas e por fim a diversidade de buscas envolvidas. A primeira dimensão, *Search Control Cardinality*, especifica se a busca global é controlada por um único processo ou por vários processos que podem colaborar entre si ou não. A segunda dimensão é relativa ao tipo de comunicação no controle da busca, abordando a questão de troca de informação ente processadores para controlar ou orientar a busca. Existem dois tipos de comunicação, chamadas de síncrona e assíncrona. Na comunicação síncrona os processadores envolvidos param em algum momento e trocam informações. Já na comunicação assíncrona, cada processo é responsável pela sua própria busca, assim sendo a busca global só termina quando cessam todas as buscas individuais. A terceira e última dimensão indica a diferenciação ou diversidade da pesquisa, os solucionadores partem das mesmas ou de soluções diferentes. Caracterizam-se dois solucionadores como “diferentes” mesmo quando baseados na mesma metodologia (por exemplo, duas buscas tabu ou dois algoritmos genéticos) se eles usarem diferentes estratégias de busca em termos de componentes (por exemplo, vizinhanças ou mecanismo de seleção) ou valores de parâmetros (GENDREAU; POTVIN, 2019).

Duas estratégias podem ser usadas para paralelizar a BT: manter caminhos de busca em paralelo; ou lidar com subproblemas menores por meio da decomposição do problema original. No primeiro caso, o processador executa a busca tabu e outros processadores executam subtarefas computacionais intensivas, como por exemplo gerar e avaliar a vizinhança da solução atual, nesse caso um alto grau de sincronização é necessário. No segundo caso vários processadores matem *threads* de pesquisa de caminho em paralelo, cada busca começa a partir de uma solução inicial diferente para explorar cada um uma região do espaço de busca, nesse caso os processadores são menos acoplados, porém um certo grau de sincronização ainda é necessário, para que as informações sobre as buscas sejam informadas ao processador central, o qual reinicializará os processos de busca (BADEAU; GUERTIN, 1997).

### 3. O PROBLEMA WFCRP

O problema WFCRP é um problema de otimização combinatória que tem por objetivo encontrar o custo mínimo para conectar todas as turbinas de um parque eólico a uma subestação a fim de coletar a energia produzida em cada uma das turbinas.

Considerando as seguintes restrições:

- Existe uma única subestação.
- Qualquer cabo disponível para ser usado no parque eólico tem capacidade maior ou igual ao total de energia produzido em qualquer turbina.
- Quando conectado a dois nós o cabo deve ter capacidade maior ou igual ao fluxo de energia que passa por ele.
- Não é permitido que cabos se cruzem.

As localizações das turbinas e subestação, assim como, o número máximo de cabos que podem ser conectados a uma subestação, são informações fornecidas.

O problema WFCRP pertence a classe NP-Difícil, é possível provar isso usando o problema da partição que é NP-Completo. O problema consiste em, dados  $n$  inteiros positivos  $x_1, x_2, \dots, x_n$ , determinar se existe um subconjunto  $Q \subset \{1, 2, \dots, n\}$  de modo que  $\sum_{i \in Q} x_i = \frac{1}{2} \sum_{i=1}^n x_i$  (OLIVEIRA; RIBEIRO, 2020).

**Teorema1.** O WFCRP é NP-Difícil mesmo se  $C = \infty$  e  $|T| = 1$ .

*Demonstração.* Seja a subestação  $V_0 = \{0\}$ , as turbinas  $V_T = \{1, 2, \dots, n\}$ , a produção de cada

turbina  $P_i = x_i, \forall i \in \{1, 2, \dots, n\}$ , e o cabo  $T = \{1\}$  com custo unitário  $u_i = 1$  e com capacidade  $k_1 = \frac{1}{2} \sum_{i=1}^n x_i$ . Assuma que a subestação está localizada no ponto  $(0,0)$  e todas as turbinas no ponto  $(0,1)$ . A resposta para o problema da partição é “sim” se, e somente se, a solução ótima para o WFCRP for 2. Se for possível encontrar a solução ótima para o WFCRP em tempo polinomial, então o problema da partição não pode ser NP-Completo, o que é uma contradição (OLIVEIRA; RIBEIRO, 2020).



### 3.1 Definição Formal

A definição do problema de roteamento de cabos usada nesse trabalho é a mesma mostrada por OLIVEIRA, RIBEIRO (2020), inicialmente proposta por FISCHETTI, PISINGER (2018), que consiste formalmente, em:

Dado um grafo  $G = (V, A)$ , um conjunto de cabos  $T$  e o número de máximo de conexões à subestação  $C$ , encontrar o custo mínimo para conectar as turbinas  $V_T$  à subestação  $V_0$ , direta ou indiretamente, de modo que não haja cabos cruzados  $[(i, j), (h, k)] \in \zeta$  e a energia  $f_{i,j}$  não exceda a capacidade do cabo utilizado na aresta  $(i, j) \in A$ .

Considere o grafo direcionado  $G = (V, A)$ , sem laços. No grafo, os nós em  $V$  são um conjunto de turbinas  $V_T$ , juntamente com a subestação  $V_0$ . As conexões possíveis são representadas pelo conjunto de arestas  $A$ , denotadas por  $(a, b)$ , indicando que é possível passar um cabo conectando os nós  $a, b \in V$ .

Todo nó  $v \in V$  possui coordenadas em um plano, permitindo o cálculo da distância entre eles e também determinar se um par de conexões  $(i, j)$  e  $(h, k)$  se cruzam. Não é considerado cruzamento se ele ocorrer em um nó, ou seja, nos pontos extremos da conexão. É permitido ter cabos paralelos, já que eles não se cruzam, mas não é permitido passar dois cabos entre um mesmo par de nós, visto que a energia é indivisível.

Também é definida a variável

$$\zeta = \{[(i, j), (h, k)] \mid (i, j), (h, k) \in A \text{ e } (i, j), (h, k) \text{ se cruzam}\},$$

representado o conjunto de pares de cabos que se cruzam.

Considere que  $P_v > 0$  é o total de energia produzida pelo nó  $v \in V_T$  e o fluxo  $f_{i,j} \geq 0$  é a energia total conduzida do nó  $i \in V$  para o nó  $j \in V$ . Considere também um conjunto de cabos  $T$ , em que cada cabo  $t \in T$  possui um custo  $u_t$  por unidade de comprimento e capacidade elétrica  $k_t$ . Assim, o custo para passar o cabo  $t$  na aresta  $(i, j)$  é

$$c_{i,j}^t = u_t \cdot dist(i, j),$$

em que  $dist(i, j)$  representa a distância euclidiana entre os nós  $i$  e  $j$ .

### 3.2 Formulação

Na formulação a seguir, a variável binária  $y_{i,j}$  define se existe um cabo na conexão  $(i,j)$ , enquanto a variável também binária  $x_{i,j}^t$  define se o cabo do tipo  $t$  é utilizado na conexão  $(i,j)$ . Como o grafo é direcionado,  $y_{i,j} = 1$  implica em  $y_{j,i} = 0$ .

A definição matemática do modelo será:

$$\text{minimizar } \sum_{(i,j) \in A} \sum_{t \in T} c_{i,j}^t \cdot x_{i,j}^t = \quad (1)$$

$$\sum_{i \in T} x_{i,j}^t = y_{i,j}, \quad (i,j) \in A \quad (2)$$

$$P_h + \sum_{i \in V} f_{i,h} = \sum_{j \in V} f_{h,j}, \quad h \in V_T \quad (3)$$

$$\sum_{t \in T} k \cdot x_{i,j}^t \geq f_{i,j}, \quad (i,j) \in A \quad (4)$$

$$\sum_{j \in V | j \neq h} y_{h,j} = 1, \quad h \in V_T \quad (5)$$

$$\sum_{j \in V | j \neq h} y_{h,j} = 0, \quad h \in V_0 \quad (6)$$

$$\sum_{i \in V | i \neq h} y_{h,j} \leq C, \quad h \in V_0 \quad (7)$$

$$y_{i,j} + y_{h,k} \leq 1, \quad [(i,j), (h,k)] \in \zeta \quad (8)$$

$$y_{i,j} \in \{0,1\}, \quad (i,j) \in A \quad (9)$$

$$x_{i,j}^t \in \{0,1\}, \quad (i,j) \in A, t \in T \quad (10)$$

A Equação (1) representa o objetivo do problema, que é minimizar o custo para conectar as turbinas. A Equação (2) garante que apenas um tipo de cabo seja utilizado em cada conexão. A Equação (3) garante que a energia flua para a subestação. A Equação (4) garante que a capacidade do cabo não vai ser excedida. A Equação (5) garante que um único cabo saia de cada turbina e (6) que nenhum cabo saia da subestação. A Equação (7) limita o número máximo de cabos que chegam na subestação. A Equação (8) impede que haja cabos cruzados. A Equação (9) garante que exista

exatamente uma ou nenhuma conexão entre dois nós, enquanto a Equação (10) garante que cada tipo de cabo seja utilizado exatamente uma ou nenhuma vez em cada conexão.

As restrições acima, implicam que as variáveis  $y$  definem um conjunto de componentes conectados, cada componente pode ser visto como uma árvore direcionada de grau 1 de saída, e sua raiz será a subestação

#### 4. TRABALHOS RELACIONADOS

Bauer, Lysgaard (2015) Definem o WFCRP como um o Problema de Roteamento de Veículo Aberto (OVRP), porem com demandas unitárias e restrições de planaridade, uma vez que os cabos não podem se cruzar e cada turbina é limitada a no máximo duas conexões. O OVRP é definido da seguinte forma: encontre um conjunto de rotas de veículos, minimizando os custos totais da rota, conectando cada cliente a um depósito, de forma que a capacidade do veículo não seja excedida e as rotas não se cruzem. O trabalho de BAUER, LYSGAARD (2015) adapta o algoritmo de CLARKE, WRIGHT (1964), inicialmente desenvolvido para o Problema de Roteamento de Veículos (VRP).

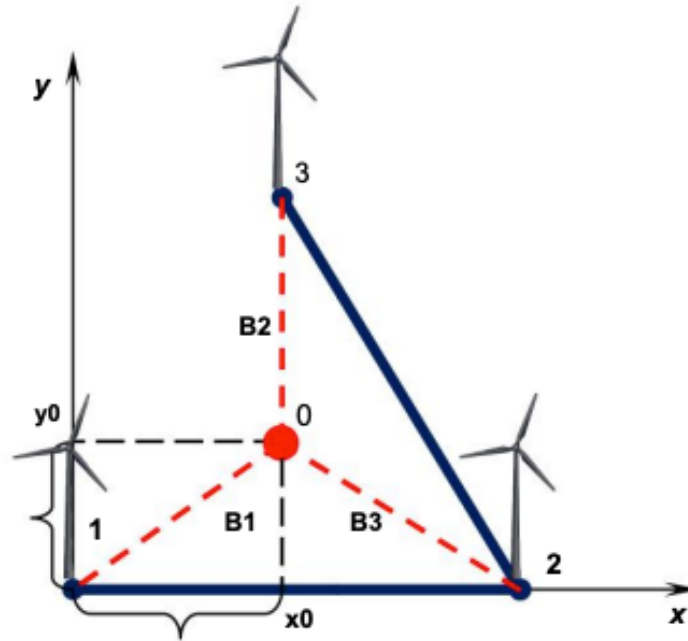
Fischetti, Pisinger (2018) propõe uma formulação semelhante a BAUER, LYSGAARD (2015) para o problema de Roteamento de Cabos em Parques Eólicos no Mar (Offshore Wind Farm Cable Routing – OWFCR), adicionando limites de conexão a subestação e diferentes tipos de cabos. O trabalho calcula o custo de conectar cada par de turbina, considerando custos diferentes para cada novo cabo conectado a uma mesma turbina. FISCHETTI, PISINGER (2018) utilizam o resolvedor CPLEX em conjunto com uma heurística. Os resultados obtidos por FISCHETTI, PISINGER (2018) são apresentados em mais detalhes na seção 4.6 deste trabalho.

O problema OWFCR é modelado por KLEIN, HAUGLAND (2019) de maneira semelhante ao proposto por FISCHETTI, PISINGER (2018), porém KLEIN, HAUGLAND (2019) lidam com obstáculos e áreas em que os cabos não podem passar, necessitando realizar um desvio.

No trabalho de NEAGU, GEORGESCU (2014) o problema é modelado como uma Minimum spanning tree (MST), adicionando a restrição referente a capacidade de cabos. Para solucionar o problema, o método usado foi a metaheurística Otimização por Colônia de Formigas (ACO), também foram adicionados ao parque pontos de Steiner, que são pontos opcionais, que podem ser usados para minimizar o custo para conectar as turbinas, ou desviar de obstáculos. O trabalho utilizou um parque eólico com 25 turbinas e uma subestação, totalizando 26 nós. Nessas configurações a abordagem utilizada conseguiu encontrar a configuração ótima para a instância usada.

A Figura 5 mostra a utilização de um ponto de Steiner, conectando todas as turbinas 1, e 3 ao ponto de Steiner 0, a fim de minimizar a distância dos três pontos, para conectá-lo posteriormente a outro ponto.

Figura 5 - Turbinas conectadas à um nó de Steiner



Fonte: NEAGU, GEORGESCU (2014)

Tifroute, Bouzahir (2019) modela o problema como a MST de Steiner, que também adiciona pontos opcionais e adiciona capacidade nos cabos. Nele, além dos pontos Steiner, foi utilizada uma abordagem híbrida de ACO e Sistema Imunológico Artificial (AIA) para solucionar o problema. Também são feitas partições no conjunto de pontos utilizando o Convex Hull (Casco Convexo, em tradução livre) para determinar os pontos de Steiner. A abordagem utilizada conseguiu encontrar a configuração ótima, para a instância de teste usada, em um curto espaço de tempo e com menos esforço computacional do que algoritmos exatos, como Kruskal e PRIM. TIFROUTE, BOUZAHIR (2019) também usou como instância um parque eólico com 26 turbinas.

## 5. ABORDAGENS UTILIZADAS

Neste capítulo serão apresentadas as abordagens utilizadas para resolver o problema WFCRP e buscar melhorar os resultados na solução proposta por Oliveira e Ribeiro (2020).

### 5.1 Busca Tabu Paralelizada

A estrutura geral da Busca Tabu proposta neste trabalho usa como base a estrutura apresentada no Algoritmo 1 no Capítulo 2, porém foi inserida a paralelização sobre a BT. A estratégia usada para paralelizar foi a de decomposição da BT em subproblemas menores apresentada na seção 2.5. Ao longo dessa seção é descrito como as soluções foram representadas e como as vizinhanças foram geradas, paralelamente.

#### 5.1.1 Representação das Soluções

Uma solução é constituída por três elementos: um grafo direcionado, representado por uma matriz de adjacência com dimensões de  $|V| \times |V| \times |T|$ , onde cada aresta  $a_{i,j,k}$  corresponde a uma conexão que representa que a turbina  $i$  flui energia para a turbina  $j$ , utilizando o cabo  $k$ ; o custo dessa solução; e qual foi a modificação feita na solução anterior para gerar essa solução. Sendo que os dois primeiros elementos representam a solução  $S$ , enquanto que as modificações podem ser armazenadas na lista tabu, caso a BT se mova para a solução  $S$ .

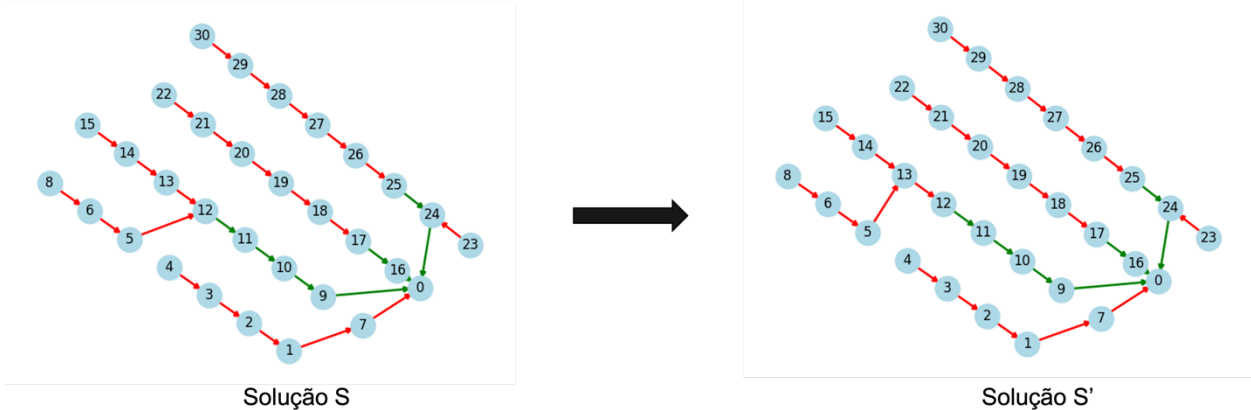
#### 5.2.2 Estrutura de vizinhança

Nesta seção são propostos dois tipos de movimentos para geração da vizinhança de uma solução, com o objetivo de possibilitar o maior o número de soluções viáveis.

O primeiro tipo de movimento consiste em trocar uma conexão entre uma turbina  $V_i$  com a turbina  $V_j$  para qualquer outra turbina  $V_{j'}$ , de forma que  $V_{j'} \in V$  e  $j' \neq j$ .

Na Figura 6 é possível ver um exemplo desse tipo de mudança, em que na Solução  $S$  a conexão  $a_{5,12,1}$ , ou seja, a turbina  $V_5$  flui energia para a turbina  $V_{12}$ , usando o cabo  $T_1$  é trocada para  $a_{5,13,1}$ , portanto, na solução  $S'$  a turbina  $V_5$  passa a fluir energia para a turbina  $V_{13}$ , usando o cabo  $T_1$ .

Figura 6 - Aplicação do movimento de troca de conexão

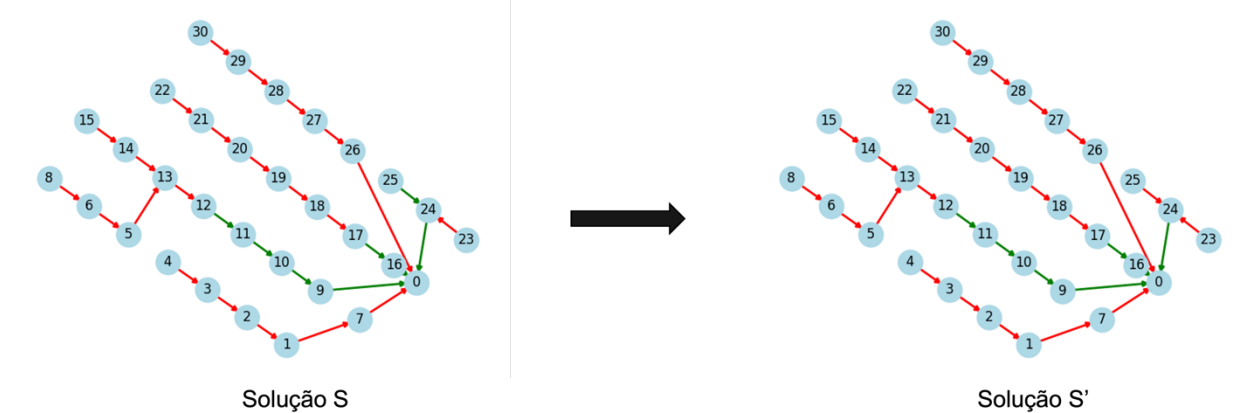


Fonte: elaborada pelos autores.

O segundo tipo de movimento é trocar o tipo de cabo utilizado na conexão entre duas turbinas, caso haja mais de um tipo de cabo disponível. Nesse tipo de mudança, a troca do cabo só acontece caso a capacidade do novo cabo seja maior ou igual quantidade total de energia que flui entre as duas turbinas, ou seja,  $k_t \geq f_{i,j}$ .

Na Figura 7 é mostrado um exemplo, na Solução S é possível ver que a conexão  $a_{25,24,2}$ , utiliza o cabo de tipo 2, ou seja, a turbina  $V_{25}$  flui energia para a turbina  $V_{24}$  usando o cabo  $T_2$  (verde), quando o movimento é aplicado em S a aresta  $a_{25,24,2}$  se torna a aresta  $a_{25,24,1}$ , portanto, na solução S' a turbina  $V_{25}$  passa a fluir energia para a turbina  $V_{24}$  usando o cabo  $T_1$  (vermelho).

Figura 7 - Aplicação do movimento de troca de cabo



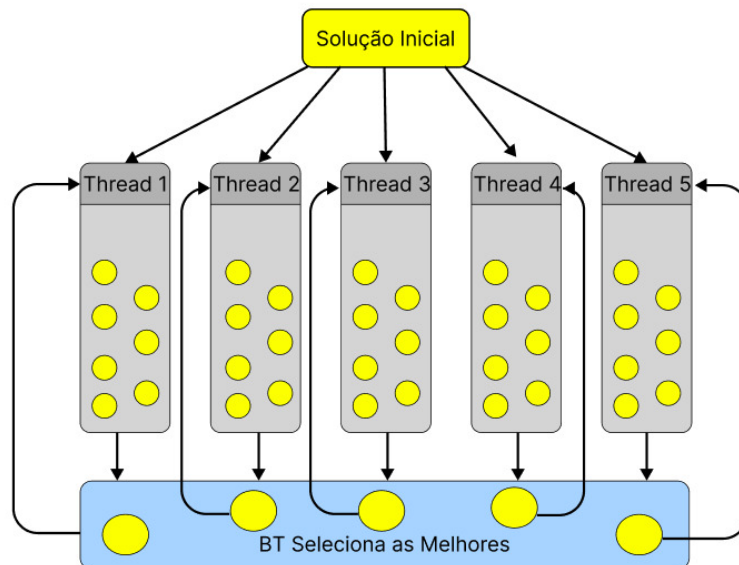
Fonte: elaborada pelos autores.

No algoritmo implementado, logo após a geração da solução candidata verifica se a solução é viável, ou seja, se  $S'$  atende a todas as restrições do problema, caso não atenda a solução não é avaliada pela função objetivo.

O algoritmo itera sobre cada vértice da solução  $S$  e tenta aplicar as mudanças e, após aplicá-las, ele verifica se a solução candidate é viável, caso seja, ele a adiciona ao conjunto de soluções viáveis  $\tilde{N}(S)$  e continua para a próxima vértice. As mudanças são aplicadas a cada vértice  $v$ , dessa forma cada Solução  $S$  pode gerar até, no máximo  $|A| \cdot (|T| - 1) \cdot 2$ , onde 2 é quantidade de tipos de movimentos possíveis, soluções, no cenário em que todas as soluções geradas pela aplicação das mudanças são validas, o que significa dizer que  $0 \geq |\tilde{N}(S)| \leq |A| \cdot (|T| - 1) \cdot 2$ .

Visando expandir o espaço de busca e, possivelmente, alcançar soluções melhores, threads de pesquisa de caminho foram implementadas. Como mostrado na figura 8, na primeira iteração cada thread parte da mesma solução inicial. Na iteração seguinte, cada thread partirá de uma solução diferente, que são as melhores soluções selecionadas na iteração atual, após a execução dos threads, a partir do conjunto de soluções gerado.

Figura 8 - Geração de Soluções usando threads



Fonte: elaborada pelos autores.

### 5.2.1 Solução Inicial

Este trabalho parte do trabalho proposto por (Oliveira; Ribeiro, 2020), no qual é proposta uma solução para o problema WFCRP usando a metaheurística *simulated*



*annealing* (SA). O trabalho compara os resultados obtidos pelo SA com os resultados obtidos por FISCHETTI, PISINGER (2018) que por sua vez utilizou uma abordagem exata em conjunto com uma heurística. O objetivo é utilizar a solução encontrada pelo trabalho de (OLIVEIRA; RIBEIRO, 2020) e, a partir dessa solução tentar encontrar soluções melhores usando outra metaheurística e técnicas de paralelização. Assim, a solução inicial usada é a melhor solução obtida por (OLIVEIRA; RIBEIRO, 2020), iniciando assim a BT paralelizada a partir do resultado obtido pelo SA. Essa técnica é usada para testar cada instancia.

## 6. EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são apresentados os experimentos computacionais feitos para comparar o método utilizado por este trabalho com os métodos utilizados por OLIVEIRA, RIBEIRO (2020) e FISCHETTI, PISINGER (2018).

O algoritmo proposto foi implementado na linguagem Java 8 com o JDK: openJDK versão 20.0.1.

Os experimentos foram realizados em um computador com 16GB de memória RAM e com um processador Apple M2 com CPU de 8 núcleos e GPU de 10 núcleos sobre o sistema operacional macOS Ventura versão 13.4.

### 6.1 Instâncias

As instancias utilizadas foram as disponibilizadas por FISCHETTI, PISINGER (2018), relativas a parques eólicos e cabos reais localizados no Reino Unido e na Dinamarca.

A Tabela 1 resume as informações relevantes de cinco diferentes parques reais, os quais foram utilizados para a definição da localização das turbinas, a tabela mostra o código de identificação da instancia, o nome do parque, o número de turbina e o limite de conexões permitidas à subestação, definido como C. No Parque wf02 não existe limitação de número de conexões à subestação, por isso C é definido como o numero de turbinas, pois caso nenhuma outra limitação seja violada todas as turbinas poderiam ser ligadas diretamente na subestação.

Tabela 1 - Parques eólicos utilizados

<b>Código</b>	<b>Nome do parque</b>	<b>Nº de turbina</b>	<b>C</b>
wf01	Horns Ver 1	80	10
wf02	Kentish Flats	30	30
wf03	Ormonde	30	4
wf04	Dan Tysk	80	10
wf05	Thanet	100	10

A Tabela 2 mostra as configurações dos conjuntos de cabos utilizados, cinco configurações foram utilizadas. Nessa Tabela está presente o código de identificação de cada conjunto de cabos (Código), os tipos de cabo presentes dentro do conjunto (Tipo), o custo por metro em mil euros de cada tipo de cabo (Custo), o número de turbinas

suportado por cada cabo em cada parque. Parques que não aceitam um determinado conjunto de cabos são apresentados como células vazias.

Tabela 2 - Conjunto de cabos e número de turbinas suportadas por cada tipo de cabo.

<b>Código</b>	<b>Tipo</b>	<b>Custo (<math>K_{euro}</math>)</b>	<b>wf01</b>	<b>wf02</b>	<b>wf03</b>	<b>wf04</b>	<b>wf05</b>
<b>cb01</b>	<b>1</b>	370	7	5	-	4	-
	<b>2</b>	393	11	8	-	6	-
	<b>3</b>	435	13	9	-	8	-
<b>cb02</b>	<b>1</b>	441	7	5	-	-	-
	<b>2</b>	450	12	8	-	-	-
<b>cb03</b>	<b>1</b>	407	-	-	5	-	-
	<b>2</b>	607	-	-	5	-	-
<b>cb04</b>	<b>1</b>	382	-	7	4	-	7
	<b>2</b>	630	-	15	9	-	15
<b>cb05</b>	<b>1</b>	440	10	7	-	6	7
	<b>2</b>	620	14	10	-	8	10

A Tabela 3 mostra as instâncias de teste, formadas pela combinação dos parques com o conjunto de cabos, contendo o número de identificação de cada instância, o código do parque e o conjunto de cabos que é utilizado nessa instância. Por exemplo, a instancia 01 é o parque wf01 combinado com o conjunto de cabo cb01.

Tabela 3 - Instâncias utilizadas neste trabalho.

<b>Identificador da instância</b>	<b>Código do Parque</b>	<b>Conjunto de Cabos</b>
01	wf01	cb01
02		cb02
03		cb05
04	wf02	cb01
05		cb02
06		cb04
07		cb05
08	wf03	cb03
09		cb04
10	wf04	cb01
11		cb05
12	wf05	cb04
13		cb05

## 6.2 Resultados

Para comparar os resultados obtidos pela BT com espaço de busca paralelizado, com os resultados obtidos pelo SA no trabalho de Oliveira, Ribeiro (2020) e pela heurística em conjunto com o resolvidor CPLEX desenvolvido por FISCHETTI, PISINGER (2018), foram utilizadas as 13 instâncias apresentadas na Tabela 3. Todos os três trabalhos utilizaram essas mesmas instâncias. Para isso foi montada a Tabela 4, em que nas quatro primeiras colunas é possível ver os custos obtidos para cada instância pela BT, Gurobi (OLIVIERA; RIBEIRO, 2020), SA (OLIVIERA; RIBEIRO, 2020), Heurística (FISCHETTI; PISINGER, 2018), respectivamente. Os valores ótimos encontrados por algum dos métodos são identificados em negrito. As três últimas colunas mostram a comparação dos resultados através da Equação 11, indicando qual seria o custo da construção encontrada pela BT em relação a outro resultado.

$$\frac{\textit{Resultado obtido pela BT} \cdot 100\%}{\textit{Resultado do método comparado}} \quad (11)$$

Na coluna 6 é mostrada a relação entre os resultados encontrados pelo Gurobi e pela BT, considerando o valor encontrado pelo Gurobi como 100%. Nessa coluna pode-se ver que o Gurobi se mostra melhor em todas as instâncias, sendo que a BT é em média 107% do valor do Gurobi. Na instância 8 é possível observar que a BT chega bem próximo do resultado obtido pelo Gurobi, chegando a 100,11% do resultado do Gurobi.

Na coluna 7 contém a relação entre a BT e o SA, tendo em vista que a BT parte da melhor solução obtida pelo SA, todos os custos obtidos pela BT são menores ou iguais aos custos obtidos pelo SA, os custos obtidos pela BT representam em média 89% dos melhores custos obtidos pelo SA, ou seja, sairia mais barato usar a BT 11% em média, sendo que na instância 11, o custo obtido pela BT seria 57% mais barato do que o obtido pela SA.

Por último, na coluna 8 está a relação entre os resultados obtidos pela BT e os obtidos pela heurística/CPLEX. A BT não conseguiu em nenhuma instância, gerar resultados melhores que a heurística/CPLEX. É possível observar que nas instâncias 4, 7 e 8, a BT chega bem próxima dos resultados obtidos pela heurística/CPLEX, com uma relação de

100,11%, 100,52% e 100,11%, para cada instância, respectivamente. A BT consegue alcançar em média 121% do valor da heurística/CPLEX.

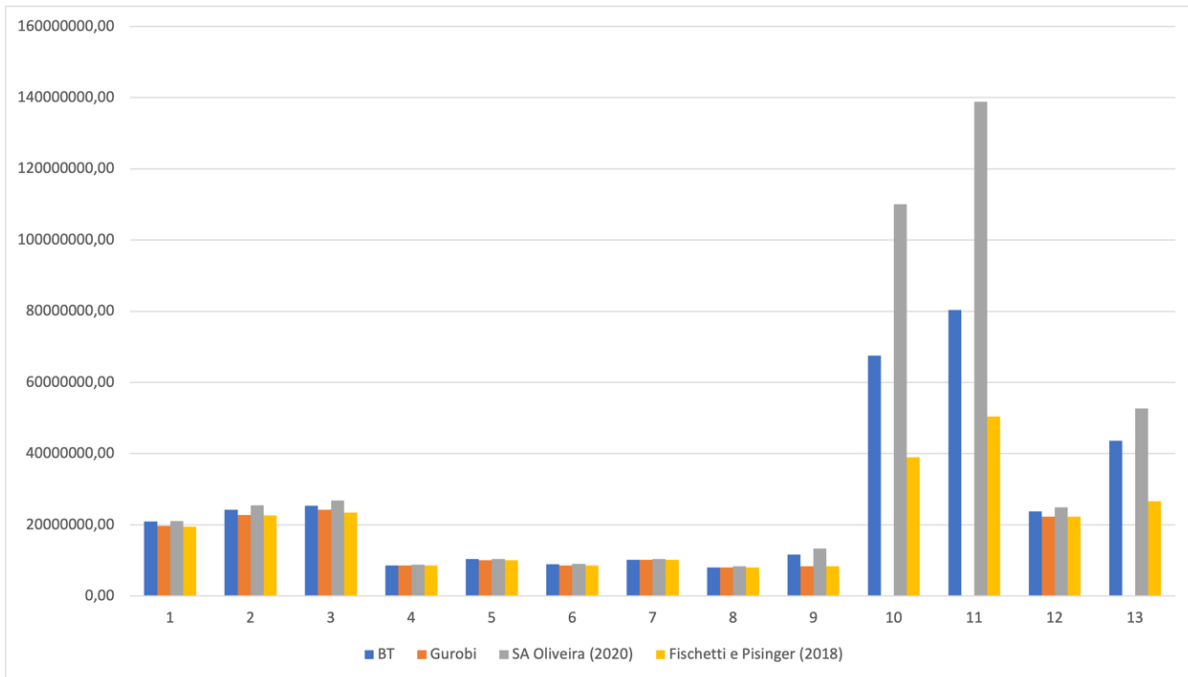
Tabela 4 - Resultados obtidos

Instância Nº	BT	Gurobi	SA Oliveira (2020)	Heurística/CPLEX	$\frac{BT \cdot 100\%}{Gurobi}$	$\frac{BT \cdot 100\%}{SA}$	$\frac{BT \cdot 100\%}{Heurística/CPLEX}$
1	20910629,33	19726498,65	21019558,82	<b>19436700,18</b>	106,00	99,48	107,58
2	24273964,92	22774240,65	25490593,12	<b>22611988,67</b>	106,59	95,23	107,35
3	25316187,63	24185686,53	26843761,73	<b>23482483,25</b>	104,67	94,31	107,81
4	8564990,30	8555171,40	8844337,97	<b>8555171,40</b>	100,11	96,84	100,11
5	10363806,10	10056670,31	10364132,25	<b>10056670,31</b>	103,05	100,00	103,05
6	8966792,98	8604208,93	9033173,72	<b>8604208,93</b>	104,21	99,27	104,21
7	10227244,04	10173931,59	10459605,98	<b>10173931,59</b>	100,52	97,78	100,52
8	8063476,67	8054844,90	8397726,03	<b>8054844,90</b>	100,11	96,02	100,11
9	11705993,13	8357195,76	13341885,09	<b>8357195,91</b>	140,07	87,74	140,07
10	67526271,91	-	110029974,69	<b>38977593,84</b>	-	61,37	173,24
11	80394103,08	-	138816418,19	50379247,34	-	57,91	159,58
12	23728889,79	22337934,77	24931796,15	<b>22337935,84</b>	106,23	95,18	106,23
13	43618798,96	-	52683071,04	26637602,25	-	82,79	163,75

Note que nas instâncias 4, 7 e 8 a BT chega bem próximo da solução ótima, já nas instancias 9, 10, 11 e 13 a BT fica longe dos valores encontrado por FISCHETTI, PISINGER (2018), no entanto, consegue melhorar bastante a solução original de OLIVEIRA, RIBEIRO (2020). Na instância 5 a BT não consegue melhorar a solução inicial de OLIVEIRA, RIBEIRO (2020), como o algoritmo da BT desenvolvido não se move para soluções invalidas, pode acarretar na BT ter ficado presa no ótimo local da solução inicial, não conseguindo expandir a busca.

A imagem da Figura 9 exibe de forma gráfica os resultados encontrados por cada método. No eixo x está representado cada instância, enquanto que no eixo y o menor custo encontrado por cada solução para cabear cada instância.

Figura 9 - Gráfico de Resultados



Fonte: elaborada pelos autores.

É possível visualizar a distância entre as soluções obtidas pela BT e as obtidas por FISCHETTI, PISINGER (2018) nas instâncias 9, 10, 11 e 13, porém é possível observar também que foram nessas instâncias em que a BT conseguiu melhorar mais o resultado obtido por OLIVEIRA, RIBEIRO (2020).

As instâncias 4, 5, 6 e 7, são configurações do parque eólico Kentish Flats, que é um parque com menor quantidade de turbinas, porém, mesmo sendo uma instância menor nem o SA nem a BT conseguiram chegar no ótimo global, ficando ambos presos em ótimos locais. Como a BT parte da solução do SA e não se move para soluções inválidas, ela acaba ficando presa em um espaço de busca limitado.

As instâncias 9, 10, 11 e 13 são instâncias em que a BT fica mais distante do ótimo. Isso acontece pela especificidade dessas instâncias. A instância 9 representa o parque wf03, que apesar de ser pequeno, apresenta uma limitação maior de quantas conexões podem ser feitas a subestação e o conjunto de cabos em que o cabo mais barato possui uma capacidade menor. As instâncias 10, 11 e 13 são instâncias maiores, partindo da solução inicial a BT ir pra soluções melhores.

## 7 CONCLUSÃO

Neste trabalho de conclusão de curso foi abordado um problema de otimização combinatória, chamado WFCRP. Para esse problema são dados um conjunto de instâncias que representam parques eólicos reais, é considerado que a localização das turbinas já foram definidos e são representados coordenadas no plano. O objetivo do problema, dado um conjunto de cabos disponíveis para cada instancia, é encontrar o menor custo possível para ligar todas as turbinas a subestação direto ou indiretamente, respeitando as restrições de cada parque.

Para resolver o problema foi proposta uma abordagem metaheurística BT, juntamente com a paralelização de caminhos de busca através de *threads*. Para cada instância foi utilizado como solução inicial a solução obtida por OLIVEIRA, RIBEIRO (2020), usando a metaheurística SA. A escolha de iniciar da solução de OLIVEIRA, RIBEIRO (2020) partiu da identificação de que as soluções encontradas pela metaheurística SA eram promissoras, levando em consideração o tempo de execução.

Através dos experimentos computacionais realizados foi identificado que apesar de, em várias instancia, a BT paralelizada melhorar a solução inicial em uma porcentagem razoável, ela ainda assim não conseguiu alcançar a solução ótima.

No trabalho de FISCHETTI, PISINGER (2018), o melhor valor conhecido é encontrado em menos de 30 minutos em 11 das 13 instâncias, utilizando heurísticas para acelerar o resolvidor CPLEX (OLIVEIRA; RIBEIRO, 2020). Já a BT paralelizada e o SA juntos executam em cerca de 2 minutos, com resultados atingindo em média 21% acima da melhor solução conhecida, sendo que em 9 instancias apresenta uma média de 4% acima da melhor solução.

Como mencionado apesar da paralelização a BT ficou rapidamente presa em um ótimo local. Para trabalhos futuros pode ser avaliado a possibilidade de permitir que a BT se mova para soluções invalidas que possam convergir para soluções validas, aumentando o espaço de busca, e assim a probabilidade de encontrar o ótimo global aumente.

Outra possível modificação seria a combinação de várias transformações em cada solução, uma abordagem parecida é a proposta por CROES (1958) para resolver o problema do Caixeiro Viajante (TSP). A heurística proposta por ele consiste em remover

$k$  arestas da solução e inserir outras  $k$ . Isso aumentaria o conjunto de soluções oriundas de  $S$ , o que poderia fazer a BT convergir para a próxima melhor solução.

Uma forma de melhorar a paralelização e aumentar o número de *threads*, seria desenvolver a BT como aplicação cliente/servidor, onde uma aplicação servidor seria desenvolvida e hospedada em nuvem, tornando os recursos computacionais escaláveis. O servidor na nuvem seria responsável por gerar o  $N(S)$  de soluções, recebendo uma solução e devolvendo um conjunto de soluções, cada solicitação poderia ser atendida por um *thread*, desacoplando totalmente as duas aplicações, o que poderia gerar melhor uso de recursos e um processamento totalmente paralelo, clusters poderiam ser criados e técnicas de *load balance* poderiam ser configuradas para aumentar o desempenho do servidor.



## REFERÊNCIAS

BADEAU, Philippe et al. **A parallel tabu search heuristic for the vehicle routing problem with time windows**. *Transportation Research Part C: Emerging Technologies*, v. 5, n. 2, p. 109-122, 1997.

BAUER, Joanna; LYSGAARD, Jens. The offshore wind farm array cable layout problem: a planar open vehicle routing problem. **Journal of the Operational Research Society**, v. 66, n. 3, p. 360-368, 2015.

CAZZARO, Davide; FISCHETTI, Martina; FISCHETTI, Matteo. Heuristic algorithms for the wind farm cable routing problem. **Applied Energy**, v. 278, p. 115617, 2020.

CLARKE, Geoff; WRIGHT, John W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, v. 12, n. 4, p. 568-581, 1964.

CROES, Georges A. A method for solving traveling-salesman problems. **Operations research**, v. 6, n. 6, p. 791-812, 1958.

FISCHETTI, Martina; PISINGER, David. **Optimal wind farm cable routing: Modeling branches and offshore transformer modules**. *Networks*, v. 72, n. 1, p. 42-59, 2018.

FISCHETTI, Martina; PISINGER, David. Optimizing wind farm cable routing considering power losses. **European Journal of Operational Research**, v. 270, n. 3, p. 917-930, 2018.

GLOVER, Fred W.; KOCHENBERGER, Gary A. (Ed.). **Handbook of metaheuristics**. Springer Science & Business Media, 2006.

IEA (2022), **Electricity Sector**, IEA, Paris <https://www.iea.org/reports/electricity-sector>, Licença: CC BY 4.0

IEA (2022), **Wind Electricity**, IEA, Paris <https://www.iea.org/reports/wind-electricity>, License: CC BY 4.0

KLEIN, Arne; HAUGLAND, Dag. Obstacle-aware optimization of offshore wind farm cable layouts. **Annals of Operations Research**, v. 272, n. 1-2, p. 373-388, 2019.

MARTINO, Jarryer Andrade de. **Algoritmos evolutivos como método para desenvolvimento de projetos de arquitetura**. 2015. 2015. Tese de Doutorado. Tese (Doutorado em Engenharia Civil, Arquitetura e Urbanismo) - Universidade Estadual de Campinas, Campinas, São Paulo.

NEAGU, Bogdan Constantin; GEORGESCU, Gheorghe. **Wind farm cable route optimization using a simple approach**. In: 2014 International Conference and Exposition on Electrical and Power Engineering (EPE). IEEE, 2014. p. 1004-1009.

RIBEIRO, Alexandre; OLIVEIRA, Bruno. **Abordagem Exata E A Metaheurística Simulated Annealing Para O Problema De Roteamento De Cabos Em Parques Eólicos**. LII Simpósio Brasileiro de Pesquisa Operacional. 2020.

SAIDUR, Rahman et al. Environmental impact of wind energy. **Renewable and sustainable energy reviews**, v. 15, n. 5, p. 2423-2430, 2011.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. **Fundamentos de Sistemas Operacionais**. Grupo Gen-LTC, 2000.

SOUZA, M. J. F. **Inteligência Computacional para Otimização: metaheurísticas** Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto, Minas Gerais, 2022, disponível em <http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>> Acesso em 09 jun. 2023

TIFROUTE, Mohamed; BOUZAHIR, Hassane. **Optimization for Onshore Wind Farm Cable: Connection Layout using ACO-AIA algorithm**. MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics, p. 67-82, 2019.