

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



**META-HEURÍSTICAS APLICADAS AO PROBLEMA DE
ROTEAMENTO DE CABOS EM PARQUES EÓLICOS**

BRUNO ALMÊDA DE OLIVEIRA

GOIÂNIA

2020

BRUNO ALMÊDA DE OLIVEIRA

**META-HEURÍSTICAS APLICADAS AO PROBLEMA DE
ROTEAMENTO DE CABOS EM PARQUES EÓLICOS**

Trabalho de Conclusão de Curso I 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 Engenharia da Computação.

Orientador(a): Alexandre Ribeiro

GOIÂNIA

2020

RESUMO

Este trabalho apresenta algo.

Palavras-chave: Palavra 1. Palavra 2. Palavra 3.

SUMÁRIO

1	INTRODUÇÃO	4
1.1	Objetivos	6
2	CONCEITOS BÁSICOS	8
2.1	Otimização Combinatória	8
2.2	Programação Linear	8
2.2.1	Resolvedor de Programação Linear	9
2.3	Heurísticas e Meta-heurísticas	10
2.3.1	Simulated Annealing	10
3	O PROBLEMA DE ROTEAMENTO DE CABOS	11
3.1	Definição formal	11
3.2	Formulação matemática	12
4	ANÁLISE COMPUTACIONAL	14
4.1	Instâncias	14
4.2	Testes	15
5	RESULTADOS PRELIMINARES	16
	REFERÊNCIAS	17

1 INTRODUÇÃO

Em 2017, o consumo de energia elétrica no mundo foi de 21372 *TWh* (IEA, 2019a). É estimado que esse consumo atinja cerca de 27000 *TWh* em 2030 e 37000 *TWh* em 2040 (IEA, 2019b). Esse aumento é esperado devido à expectativa no aumento de veículos elétricos utilizados em todo o mundo. Com o aumento no consumo de energia, serão necessários mais fontes de energia, possivelmente causando ainda mais impactos na natureza.

A energia eólica contribui na redução do impacto ambiental causado pela produção de energia elétrica (CAZZARO; FISCHETTI; FISCHETTI, 2019). Já as usinas que utilizam carvão, gás ou petróleo contribuem bastante para a emissão de poluentes na atmosfera (SAIDUR et al., 2011). As usinas nucleares entram na lista de energia limpa por não liberar substâncias na atmosfera (SADEKINA et al., 2019), mas há controvérsias devido a criação de substâncias tóxicas e ao alto risco em caso de acidente (NIRS, 2011; GREENPEACE, 2017). As hidrelétricas também são uma fonte limpa de energia, utilizando como combustível a água, mas necessita de barragens que bloqueiam o curso dos rios, alteram o fluxo de água e podem inundar áreas inteiras em caso de acidente (KOONTZ, 2015).

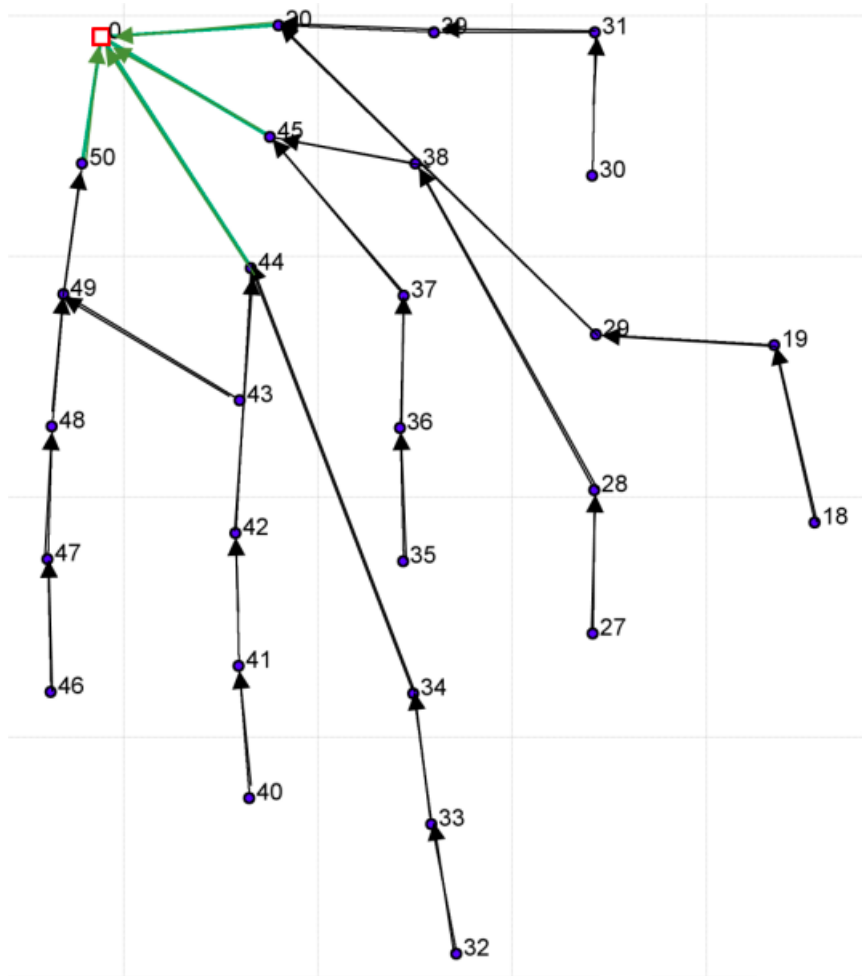
Apesar de limpa, o parque eólico possui uma produção de energia bastante variável e dependente do clima (FANTIDIS et al., 2012). Já as hidrelétricas são bem menos sujeitas a variações, inclusive podendo armazenar água em um reservatório para suprir uma possível alta na demanda de energia (FOLK, 2018). A fonte de energia mais estável é a nuclear, permanecendo em sua capacidade máxima cerca de 92,6% do tempo, contra as hidrelétricas que estão em sua capacidade máxima em 42,8% do tempo e os parques eólicos em 37,4% (Robin Office of Nuclear Energy, 2018).

O investimento para construção de uma parque eólico está entre as mais baratas se construída em terra e entre as mais caras se construída no mar (EIA, 2020). Gastos com a infraestrutura elétrica na construção de parques eólicos no mar variam entre 15% e 30% dos custos iniciais, 12% a 26% somente com os sistemas de coleta e transmissão de energia (GONZÁLEZ et al., 2014). Por isso, é bastante importante otimizar a forma com que os cabos são conectados para coleta de energia.

Após uma breve análise dos resultados, é possível perceber que a solução terá uma forma de árvore, enraizada na subestação, com os filhos sendo as turbinas. A figura 1 apresenta uma visualização da árvore como solução. Note que não existe uma conexão entre os vértices 42 e 43 nem entre 20 e 38.

Para reduzir os custos, cada turbina pode ser conectada em série a outra turbina, que pode estar conectada a outra, e assim sucessivamente até a subestação. A figura 2 mostra a topologia de um parque real, Barrow, construído no mar da Irlanda. Note que a

Figura 1 – Visualização de uma solução como uma árvore.



Fonte: adaptado de Fischetti e Pisinger (2017).

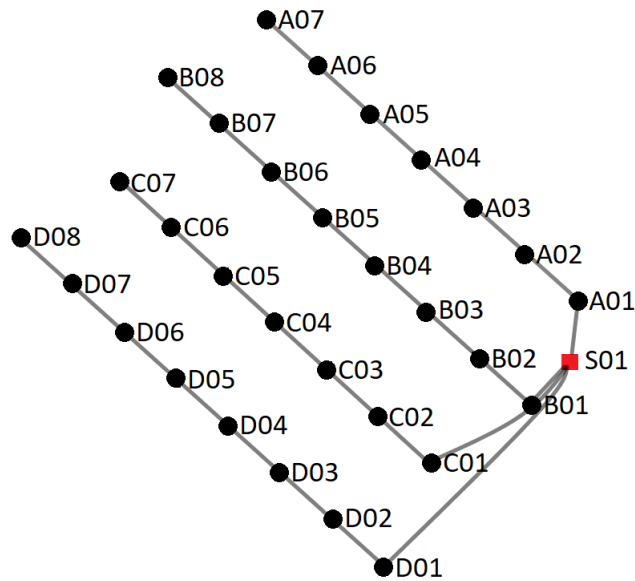
conexão $S01-B01$ é paralela às conexões $S01-C01$ e $S01-D01$, ambas passando ao lado da turbina $B01$ sem se conectar.

Um requisito das empresas de energia é que não haja dois cabos cruzados, evitando uma possível remoção de um cabo para manutenção em outro (FISCHETTI; PISINGER, 2017). Além disso, se dois cabos se cruzam, é necessário isolá-los, já que aquecem bastante devido a alta voltagem nos cabos (BAUER; LYSGAARD, 2015).

Os parques eólicos podem ser conectados utilizando vários tipos de cabos ou apenas um tipo em comum para toda o parque. A figura 3 mostra qual seria a topologia ótima utilizando um único tipo de cabo, os mesmos do projeto original.

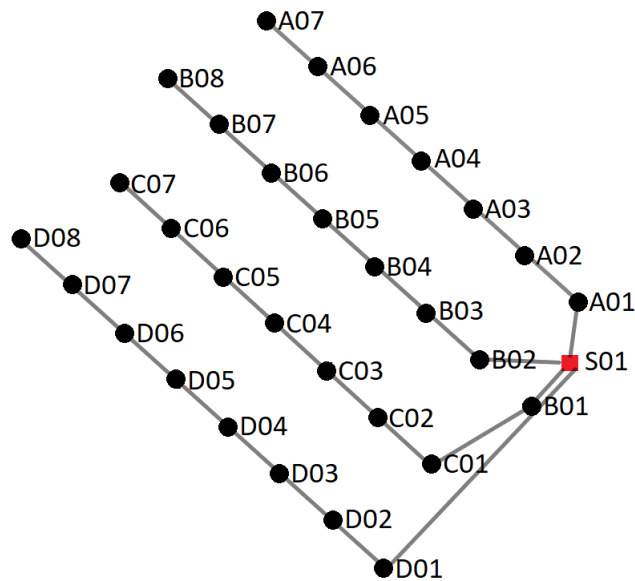
Existem diversos tipos de cabos disponíveis com custos diferentes. Cada um possui suas vantagens, com resistência e capacidade elétrica diferentes. Isso gera ainda mais opções de conexões, dando opção de escolher o cabo a ser utilizado em cada conexão. Levando em consideração a resistência elétrica do cabo utilizado, o problema modela melhor a realidade e é possível planejar perdas de lucro a longo prazo. A figura 4 mostra qual seria a conexão ótima para o parque Barrow se fossem utilizados dois tipos diferentes de cabos.

Figura 2 – Parque eólico Barrow no mar da Irlanda. Um círculo representa uma turbina, enquanto um quadrado representa uma subestação.



Fonte: adaptado de Bauer e Lysgaard (2015).

Figura 3 – Conexão ótima para o parque eólico Barrow utilizando o mesmo tipo de cabo do projeto já construído.

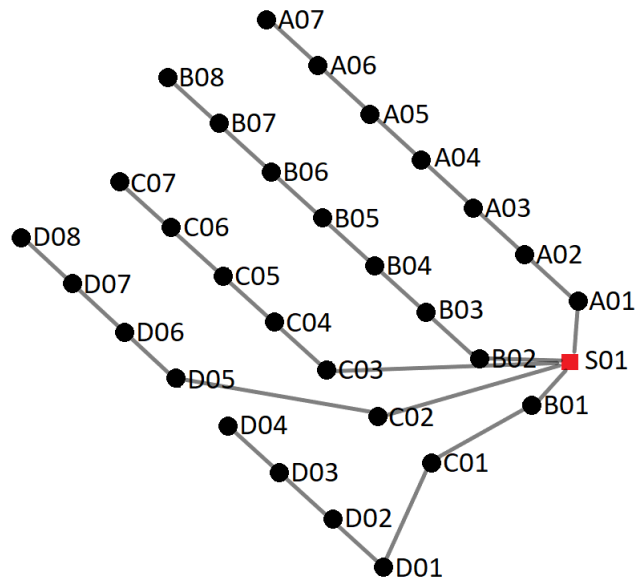


Fonte: adaptado de Bauer e Lysgaard (2015).

1.1 Objetivos

O objetivo deste trabalho de conclusão de curso é expor uma abordagem utilizando meta-heurística para solução do problema de roteamento de cabos elétricos. Será abordado apenas o problema de cabeamento, supondo que a definição da localização das turbinas e da subestação já foram definidos. Para reduzir o problema, o parque eólico será limitada

Figura 4 – Conexão ótima para o parque eólico Barrow utilizando dois tipos de cabos. Note que as turbinas são limitadas a duas conexões.



Fonte: adaptado de Bauer e Lysgaard (2015).

a uma subestação, que é o que ocorre na maioria dos casos (CAZZARO; FISCHETTI; FISCHETTI, 2019). A meta-heurística deve: dado um conjunto de cabos, as posições de várias turbinas e de uma subestação, encontrar uma forma econômica de colocar os cabos entre as turbinas e a subestação de forma que a energia das turbinas flua para a subestação sem exceder os limites dos cabos.

2 CONCEITOS BÁSICOS

Este capítulo apresenta os conceitos necessários para o entendimento do restante do trabalho. A seção 2.1 introduz o conceito de otimização combinatória, que será utilizada para definição formal do problema. A seção 2.2 define o que é programação linear e suas variantes, além de apresentar alguns resolvedores para elas. Por fim, a seção 2.3 apresenta o conceito de heurística e meta-heurística, juntamente com os algoritmos utilizados neste trabalho.

2.1 Otimização Combinatória

Problemas de otimização combinatória são aqueles que necessitam de técnicas matemáticas dentro de um conjunto de soluções finito, visando encontrar a solução ótima (VELDKAMP, 2015). Esse conjunto de soluções é definido por um conjunto de restrições e, geralmente, possui uma quantidade enorme de soluções distintas.

Segundo Carvalho et al. (2001), um problema de otimização combinatória possui um conjunto de instâncias, um conjunto de soluções viáveis $S(I)$ para cada instância I , e uma função $f(s)$, o valor da solução $s \in S(I)$. O conjunto $S(I)$ é definido a partir das restrições do problema. Se $S(I)$ é vazio, então a instância I não possui solução.

Problemas de minimização e maximização buscam uma solução de valor mínimo e máximo, respectivamente. Uma solução $s^* \in S(I)$ é ótima para uma instância I se a instância possuir solução e $f(s^*) \leq f(s), \forall s \in S(I)$ para um problema de minimização ou $f(s^*) \geq f(s), \forall s \in S(I)$ para um problema de maximização.

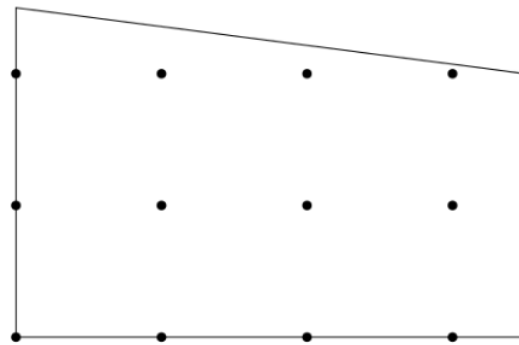
2.2 Programação Linear

Programação Linear (LP) é um problema de otimização combinatória que possui como objetivo minimizar uma função linear sujeito a equações e inequações lineares (BERTSIMAS; TSITSIKLIS, 1997). Na LP, as variáveis assumem valores reais, enquanto na Programação Linear Inteira (ILP), todas as variáveis devem ser inteiras. Note que variáveis binárias, isto é, que assumem valores 0 ou 1, também podem ser classificadas como variáveis inteiras. A figura 5 mostra um exemplo de problema em LP e em ILP.

O conjunto intermediário entre a LP e a ILP é a Programação Linear Inteira Mista (MILP), que permite tanto variáveis com valores inteiros ou quanto com valores decimais.

A LP pode ser resolvida em tempo polinomial (KARMARKAR, 1984), mas a ILP e a MILP estão na classe NP-Completo (KARP, 1972). Isto significa que a LP pode ser resolvida em um curto espaço de tempo enquanto a ILP e a MILP podem levar milhares

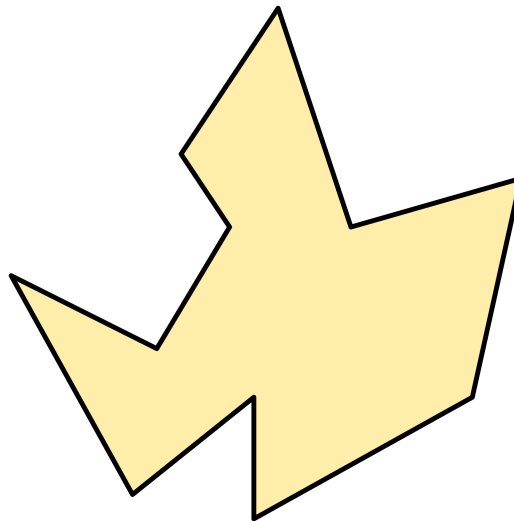
Figura 5 – Exemplo de problema que pode ser resolvido com LP. As retas representam as restrições. Uma solução em LP deve estar dentro da área enquanto em ILP deve ser um dos pontos destacados.



Fonte: Adaptado de Mixed-Integer... (2020)

de anos. A figura 6 mostra uma possível área de soluções válidas para um problema 2D em MILP. Com a figura é possível imaginar o porquê da dificuldade de resolver um problema em MILP.

Figura 6 – Exemplo de uma possível área de soluções de um problema em ILP ou MILP.



Fonte: Wikimedia (2017)

2.2.1 Resolvedor de Programação Linear

Para encontrar soluções de um modelo MILP, resolvedores foram desenvolvidos reunindo-se avanços e técnicas da área. Alguns deles são: o GNU Linear Programming Kit (GLPK, 2018), o SCIP (ACHTERBERG, 2007) e o Gurobi (Gurobi Optimization, LLC, 2020).

Todos os resolvedores conseguem atingir a solução ótima do problema, mas não conseguem garantir quando essa solução será encontrada, já que, normalmente, utilizam

o método enumerativo *branch-and-bound* (BOCKMAYR; WEISPFENNING; MAHER, 2001). O método faz “podas” sempre que, a partir da solução atual, na melhor das hipóteses, não será gerada uma solução melhor que a melhor já encontrada, reduzindo o número de soluções observadas e descartando soluções não promissoras.

2.3 Heurísticas e Meta-heurísticas

Para alguns problemas específicos foram desenvolvidas as heurísticas, que não garantem a solução ótima mas podem atingir bons resultados em um curto espaço de tempo (GLOVER, 1986). Já as meta-heurísticas são independentes do problema a ser resolvido, talvez necessitando alguma adaptação do problema para o funcionamento correto do algoritmo (LEHTIHET, 2013).

Cada meta-heurística pode ser classificada como de solução única ou de população de soluções interagindo entre si, de construção de solução ou de melhorias das anteriores (GENDREAU; POTVIN, 2005). Além disso, é necessário definir os parâmetros da meta-heurística, como número de iterações, tamanho da solução e probabilidade de trocar de solução. Todos esses fatores impactam no tempo e na qualidade do algoritmo.

2.3.1 Simulated Annealing

O Simulated Annealing (SA) é inspirado pela metalúrgica, quando é necessário o refinamento de determinado material. O material é aquecido a uma temperatura muito alta e resfriado lentamente, permitindo que os átomos se organizem e reduzam os defeitos do material (KIRKPATRICK; GELATT; VECCHI, 1983).

O algoritmo mantém uma temperatura T e uma solução x . A cada iteração, uma nova solução y é gerada a partir de x . A solução y é atribuída a x se o seu resultado for melhor ou se

$$\exp\left(\frac{f(y) - f(x)}{T}\right) < r$$

sendo f a função custo da solução e r um número aleatório no intervalo $[0, 1]$. Ao final da iteração, a temperatura T é resfriada a um fator α . O processo é repetido até atingir o critério de parada, retornando a melhor solução encontrada.

3 O PROBLEMA DE ROTEAMENTO DE CABOS

O primeiro passo no desenvolvimento de um parque eólico é a definição das localizações das turbinas e da subestação. Neste trabalho, é assumido que as localizações já foram definidas. Também é assumido que haverá apenas uma subestação por parque. Durante todo o trabalho, a energia será dada na unidade de medida *megawatt* (MW), exceto quando a medida estiver presente de maneira explícita.

O problema foi modelado baseado na formulação de Fischetti e Pisinger (2017), que assume os seguintes critérios:

- Qualquer cabo deve ter uma capacidade maior ou igual ao total de energia em qualquer turbina;
- Cabos diferentes estão disponíveis;
- Um cabo deve suportar o fluxo de energia que passa por ele;
- É dado o número máximo C de cabos podem ser conectados a uma subestação;
- Cabos cruzados devem ser evitados.

Para simplificar a formulação, o grafo não direcionado foi transformado em grafo direcionado, permitindo tratar as arestas (i, j) e (j, i) de maneira independente.

3.1 Definição formal

Considere o grafo completo e 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 os mesmos não se cruzam e não criará problemas, mas não é permitido passar dois cabos entre um mesmo par de nós, visto que a energia é indivisível.

Também é definida a variável

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

representando o conjunto de pares de cabos que se cruzam.

Considere ainda que $P_v > 0$ é o total de energia produzida pelo nó $v \in V_T$ e o fluxo de energia $f_{i,j} \geq 0$ é a energia total conduzida a partir do nó $i \in V$ para o nó $j \in V$. Considere também um conjunto de cabos T , cada cabo $t \in T$ com 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

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

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

Formalmente, o problema de roteamento de cabos é: dado um grafo G , um conjunto de cabos T e o número de máximo de conexões à subestação C , encontrar o custo mínimo para conectar as turbinas em V_T à subestação em V_0 , direta ou indiretamente, de modo que não hajam cabos cruzados $[(i, j), (h, k)] \in \zeta$ e a energia $f_{i,j}$ não exceda a capacidade do cabo que passa pela aresta $(i, j) \in A$, permitindo a coleta da energia $\sum_{v \in V_T} P_v$.

3.2 Formulação matemática

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

A definição matemática do modelo será:

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

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

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

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

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

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

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

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

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

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

A equação (1) representa o objetivo do problema, que é minimizar o custo para conectar as turbinas. A equação (2) define o número de conexões entre dois nós. 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 exatamente um cabo sai de cada turbina e (6) que nenhum sai da subestação. A equação (7) limita o

número máximo de cabos que chegam na subestação. A equação (8) impede que hajam cabos cruzados. A equação (9) garante que exista exatamente uma ou nenhuma conexão entre dois nós, enquanto a equação (10) garante que um único tipo de cabo é utilizado na conexão.

Note que a variável y define uma componente conexa na subestação. Isto é, o resultado será uma árvore, com a raiz na subestação e os demais nós serão as turbinas, com a energia fluindo desde as folhas para a subestação.

4 ANÁLISE COMPUTACIONAL

4.1 Instâncias

Para teste dos algoritmos, foram utilizados instâncias reais de parques eólicos localizados no Reino Unido e na Dinamarca. Todos os testes foram disponibilizados por Fischetti e Pisinger (2017). A Tabela 1 apresenta os parques reais utilizados no trabalho.

Tabela 1 – Parques eólicos utilizados

Código	Nome do parque	Nº de turbinas	C
wf01	Horns Rev 1	80	10
wf02	Kentish Flats	30	∞
wf03	Ormonde	30	4
wf04	Dan Tysk	80	10
wf05	Thanet	100	10

Fonte: Adaptada de Fischetti e Pisinger (2017)

Além das localizações das turbinas, é necessário definir quais cabos serão utilizados. Nos testes, cinco conjuntos de cabos foram utilizados.

Tabela 2 – Conjuntos de cabos e número de turbinas suportadas por cada cabo.

Código	Tipo	Nº de turbinas suportadas				
		wf01	wf02	wf03	wf04	wf05
cb01	1	7	5	3	4	5
	2	11	8	4	6	8
	3	13	9	6	8	9
cb02	1	7	5	3	5	5
	2	12	8	5	7	8
cb03	1	12	8	5	7	8
	2	23	16	5	14	16
cb04	1	9	7	4	6	7
	2	21	15	9	13	15
cb05	1	10	7	4	6	7
	2	14	10	6	8	10

Fonte: Fischetti e Pisinger (2017)

4.2 Testes

Inicialmente, foi desenvolvido o algoritmo ótimo para o problema utilizando força bruta. O algoritmo possui complexidade $O(t \cdot n^2 \cdot 3^n)$, onde t é o número de cabos e n o número de vértices, sendo impossível executar inclusive nas menores instâncias, com 30 vértices. Com base nisso, foi desenvolvido uma solução utilizando o Gurobi, que sempre encontra o valor ótimo, mas sem garantia de tempo. Para evitar que muito tempo fosse utilizado, a execução em cada instância foi limitada a duas horas. A tabela 3 apresenta os resultados obtidos, juntamente com os resultados de outros trabalhos.

Tabela 3 – Resultados obtidos. O símbolo — indica que nenhuma solução foi encontrada.

Número	Parque	Conjunto de Cabos	Gurobi	Simulated Annealing	Fischetti e Pisinger (2017)
01	wf01	cb01	19726498.65	64615874.32	19436700.18
02		cb02	22774240.65	75999218.47	22611988.67
03		cb05	24185686.53	71856516.85	23482483.25
04	wf02	cb01	8555171.40	16295486.89	8555171.40
05		cb02	10056670.31	19877912.64	10056670.31
06		cb04	8604208.93	17270682.16	8604208.93
07		cb05	10173931.59	19892932.33	10173931.59
08	wf03	cb03	8054844.90	18816377.02	8054844.90
09		cb04	8357195.76	17315242.50	8357195.91
10	wf04	cb01	—	165123223.49	38977593.84
11		cb05	—	192619312.43	50379247.34
12	wf05	cb04	22337934.77	59909852.85	22337935.84
13		cb05	—	69006113.23	26637602.25

Fonte: Autor

Foram utilizados números pseudo-aleatórios para uma primeira visualização do funcionamento do algoritmo e averiguação de possíveis bugs, os parâmetros não foram calibrados e o algoritmo executa em menos de 5% do tempo utilizado pelo Gurobi. Ainda nos testes 10 e 11 o algoritmo chega em uma solução em que é impossível fazer qualquer movimento, ficando preso sem poder otimizá-la.

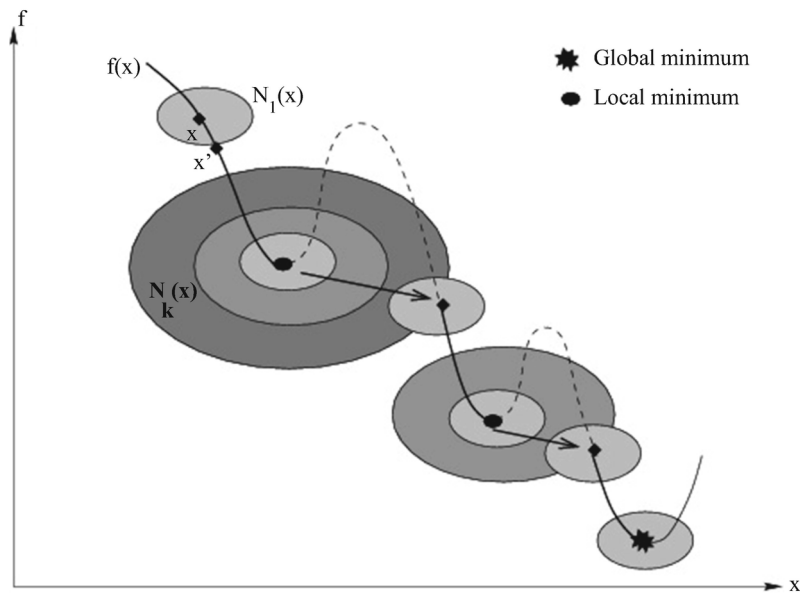
5 RESULTADOS PRELIMINARES

Neste trabalho foi desenvolvido a meta-heurística SA para resolver o problema de roteamento de cabos. Também foi desenvolvido a solução ótima utilizando força bruta e o resolvidor Gurobi.

No trabalho de Fischetti e Pisinger (2017) o valor ótimo é encontrado em menos de 30min para todas as instâncias exceto 03 e 13, utilizando math-heurísticas no resolvidor CPLEX. Os resultados do SA foram cerca de 100% piores quando comparados com o resolvidor Gurobi. Vale ressaltar que o SA é extremamente simples e se concentra em uma única solução, que pode ser trocada por outra pior durante a execução do algoritmo.

Além do SA, pretende-se implementar o algoritmo de Busca por Vizinhaça Variável (VNS). O algoritmo explora grupos de soluções vizinhas ao invés de soluções isoladas, aplicando uma busca local em cada vizinhança para tentar encontrar o ótimo da região. Acredita-se que a exploração por vizinhanças vai reduzir o número total de soluções pesquisadas, permitindo que o algoritmo não perca tempo em soluções extremamente ruins. A figura 7 dá uma ideia do funcionamento do algoritmo.

Figura 7 – Exemplo de Busca por Vizinhaça Variável.



Fonte: Hansen e Mladenović (2018)

Como a busca local deve ser implementada para uso do VNS, ela também pode ser adicionada ao SA para impedi-lo de aceitar soluções que provavelmente não levam à ótima. Espera-se que essa adição melhore bastante os resultados.

REFERÊNCIAS

- ACHTERBERG, T. **Constraint Integer Programming**. Tese (Doutorado), 01 2007.
- BAUER, J.; LYSGAARD, J. The offshore wind farm array cable layout problem: a planar open vehicle routing problem. **Journal of the Operational Research Society**, v. 66, p. 360–368, 2015.
- BERTSIMAS, D.; TSITSIKLIS, J. N. **Introduction to Linear Optimization**. [S.l.]: Athena Scientific, 1997.
- BOCKMAYR, A.; WEISPFENNING, V.; MAHER, M. Chapter 12 - solving numerical constraints. In: ROBINSON, A.; VORONKOV, A. (Ed.). **Handbook of Automated Reasoning**. Amsterdam: North-Holland, 2001, (Handbook of Automated Reasoning). p. 751 – 842. ISBN 978-0-444-50813-3. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B978044450813350014X>>.
- CARVALHO, M. H. de; CERIOLI, M. R.; DAHAB, R.; FEOFILOFF, P.; FERNANDES, C. G.; FERREIRA, C. E.; GUIMARÃES, K. S.; MIYAZAWA, F. K.; JR., J. C. de P.; SOARES, J. A. R.; WAKABAYASHI, Y. **Uma Introdução Sucinta a Algoritmos de Aproximação**. 2001.
- CAZZARO, D.; FISCHETTI, M.; FISCHETTI, M. **Heuristic Algorithms for the Wind Farm Cable Routing Problem**. 2019.
- EIA. **Cost and Performance Characteristics of New Generating Technologies**. [S.l.], 2020.
- FANTIDIS, J.; MANTZARI, V.; KALKANI, E.; BANDEKAS, D.; VORDOS, N. A hybrid wind and hydroelectric power production system in plaka, alexandroupolis, greece. **International Journal of Computational Engineering Science**, v. 2, n. 4, p. 376–385, 11 2012.
- FISCHETTI, M.; PISINGER, D. Optimizing wind farm cable routing considering power losses. **European Journal of Operational Research**, v. 270, n. 3, p. 917–930, 2017.
- FOLK, E. **Hydroelectric vs. wind power: What offshore energy should we invest in?** 2018. Disponível em: <conservationfolks.com/hydroelectric-vs-wind-power-what-offshore-energy-should-we-invest-in>.
- GENDREAU, M.; POTVIN, J.-Y. Metaheuristics in combinatorial optimization. **Annals of Operations Research**, v. 140, p. 189–213, 2005.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers and Operations Research**, v. 13, n. 5, p. 533–549, 1986.
- GLPK. **GNU Linear Programming Kit**. 2018. Disponível em: <www.gnu.org/software/glpk>.
- GONZÁLEZ, J. S.; PAYÁN, M. B.; SANTOS, J. M. R.; LONGATT, F. G. A review and recent developments in the optimal wind-turbine micro-siting problem. **Renewable and Sustainable Energy Reviews**, v. 30, p. 133–144, 2014.

GREENPEACE. **Nuclear Energy**. 2017. Disponível em: <www.greenpeace.org/usa/global-warming/issues/nuclear>.

Gurobi Optimization, LLC. **Gurobi Optimizer Reference Manual**. 2020. Disponível em: <www.gurobi.com>.

HANSEN, P.; MLADENOVIC, N. Variable neighborhood search. In: MARTÍ, R.; PARDALOS, P.; RESENDE, M. (Ed.). **Handbook of Heuristics**. Cham: Springer Nature, 2018. cap. 15, p. 759–787.

IEA. **Electricity Information 2019**. 2019. Disponível em: <www.iea.org/reports/electricity-information-2019>.

IEA. **Global electricity demand by region in the Stated Policies Scenario, 2000-2040**. 2019. Disponível em: <www.iea.org/data-and-statistics/charts/global-electricity-demand-by-region-in-the-stated-policies-scenario-2000-2040>.

KARMARKAR, N. A new polynomial-time algorithm for linear programming. **Combinatorica**, v. 4, p. 373–395, 12 1984.

KARP, R. M. Reducibility among combinatorial problems. In: MILLER, R. E.; THATCHER, J. W. (Ed.). **Complexity of Computer Computations**. [S.l.]: Plenum Press, New York, 1972. (The IBM Research Symposia Series), p. 85–103.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 05 1983.

KOONTZ, R. **What's Good and What's Bad about Hydropower?** 2015. Disponível em: <www.kidsdiscover.com/teacherresources/whats-good-and-whats-bad-about-hydropower>.

LEHTIHET, H. **What are the differences between heuristics and metaheuristics?** 2013. Disponível em: <www.researchgate.net/post/What_are_the_differences_between_heuristics_and_metaheuristics/51f1d695d039b1ed71ad3e5d/citation/download>.

MIXED-INTEGER Programming. 2020. Disponível em: <developers.google.com/optimization/mip/integer-opt>.

NIRS. **Nuclear Energy is Dirty Energy**. [S.l.], 2011.

Robin Office of Nuclear Energy. **Nuclear Power is the Most Reliable Energy Source and It's Not Even Close**. 2018. Disponível em: <www.energy.gov/ne/articles/nuclear-power-most-reliable-energy-source-and-its-not-even-close>.

SADEKINA, S.; ZAMANA, S.; MAHFUZA, M.; SARKAR, R. Nuclear power as foundation of a clean energy future: A review. **Energy Procedia**, v. 160, p. 513–518, 2 2019.

SAIDUR, R.; RAHIM, N. A.; ISLAM, M. R.; SOLANGI, K. H. Environmental impact of wind energy. **Renewable and Sustainable Energy Reviews**, v. 15, p. 2423–2430, 2011.

VELDKAMP, B. P. Computerized test construction. **International Encyclopedia of the Social and Behavioral Sciences**, v. 4, p. 2477–2480, 2015.

WIKIMEDIA. **Simple concave polygon**. 2017. Disponível em: <commons.wikimedia.org/wiki/File:Simple_concave_polygon.svg>.