

**UNIVERSIDADE FEDERAL DE ALAGOAS-UFAL
CAMPUS DE A.C. SIMÕES
CIÊNCIA DA COMPUTAÇÃO**

LUAN GUSTAVO ALVES VIANA

**UMA META-HEURÍSTICA HÍBRIDA PARA O PROBLEMA DE COBERTURA DE
DISCOS PONDERADOS**

**MACEIÓ
2022**

Luan Gustavo Alves Viana

Uma meta-heurística híbrida para o problema de cobertura de discos ponderados

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal de Alagoas - UFAL, Campus de A.C. Simões.

Orientador: Prof. Dr. Rian Gabriel Santos Pinheiro

Maceió
2022

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

V614m Viana, Luan Gustavo Alves.

Uma meta-heurística híbrida para o problema de cobertura de discos ponderados / Luan Gustavo Alves Viana. – 2022.
30 f. : il.

Orientador: Rian Gabriel Santos Pinheiro.

Monografia (Trabalho de conclusão de curso em Ciência da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2022.

Bibliografia: f. 29-30.

1. Meta-heurística. 2. Cobertura (Redes de computadores). 3. Discos ponderados. I. Título.

CDU: 004.023

Dedico este trabalho aos meus pais, meus maiores incentivadores, que sempre acreditaram em mim, que tenho orgulho de compartilhar a vida e os ensinamentos que adquiro.

AGRADECIMENTOS

Quero agradecer, em primeiro lugar, à Deus, pela força e coragem durante toda esta longa caminhada. Agradeço também a todos os professores do curso, em particular ao meu orientador prof. Rian Gabriel Santos Pinheiro, que foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia.

Por fim, a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Se você acredita de coração, sabe que é uma coisa importante pra você e que não vai machucar ninguém, vá em frente e faça.

Renato Russo

RESUMO

Neste trabalho é realizado o estudo do problema da cobertura de pontos por discos ponderados, onde dado um conjunto de n pontos $p \in P$, com $p = (x, y) \in \mathbb{R}^2$, e m discos $d \in D$, sendo $d = (r, c)$ com r e $c \in \mathbb{R}$ representando o raio de intensidade e o custo, respectivamente, busca-se encontrar um subconjunto $D' \subseteq D$ que cubra todos os pontos com custo mínimo.

Este problema relaciona-se com problemas demandados por cidades inteligentes, como posicionamento de estações rádio base, iluminação, entre outras onde o objetivo é otimizar os recursos.

O trabalho propõe um método exato usando programação quadrática e uma meta-heurística híbrida CMSA. O exato conseguiu provar otimalidade em instâncias com até 29 pontos, percebendo a influência do tamanho e distribuição dos pontos em cada instância, onde a heurística híbrida acha todos os ótimos encontrados pelo exato em um baixo tempo computacional.

Palavras-chave: CMSA.

ABSTRACT

In this work, the study of the problem of covering points by weighted discs is carried out, where given a set of n points $p \in P$, with $p = (x, y) \in \mathbb{R}^2$, and m disks $d \in D$, being $d = (r, c)$ with r and $c \in \mathbb{R}$ representing the intensity radius and the cost, respectively, with objective to find a subset $D' \subseteq D$ that covers all points with minimum cost.

This problem is related to problems demanded by smart cities, such as positioning of radio base stations, lighting, among others, where the objective is to optimize resources.

The work proposes an exact method using quadratic programming and a CMSA hybrid metaheuristic. Exact was able to prove optimality in instances with up to 29 points, realizing the influence of the size and distribution of points in each instance, where the hybrid heuristic finds all the optimal ones found by exact in a low computational time.

Keywords: CMSA.

LISTA DE FIGURAS

Figura 1 – Posicionamento de antenas	14
Figura 2 – Exemplo do problema abordado	15
Figura 3 – Particionamento recursivo de P	22
Figura 4 – Exemplo de construção de D'	23

LISTA DE TABELAS

Tabela 1 – Instâncias	25
Tabela 2 – Resultados Exato x CMSA	26

LISTA DE ABREVIATURAS E SIGLAS

CMSA Construct, Merge, Solve & Adapt

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Definição formal do problema	15
1.2	Objetivos	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Algoritmos aproximativos	17
2.2	Heurísticas e aplicações	17
3	ALGORITMOS PARA O PCPDP	19
3.1	Formulação Matemática	19
3.2	CMSA	20
3.2.1	Construct	21
3.2.2	Merge	23
3.2.3	Solve	23
3.2.4	Adapt	24
4	RESULTADOS	25
4.1	Instâncias	25
4.2	Experimentos	26
5	CONCLUSÃO	28
	REFERÊNCIAS	29

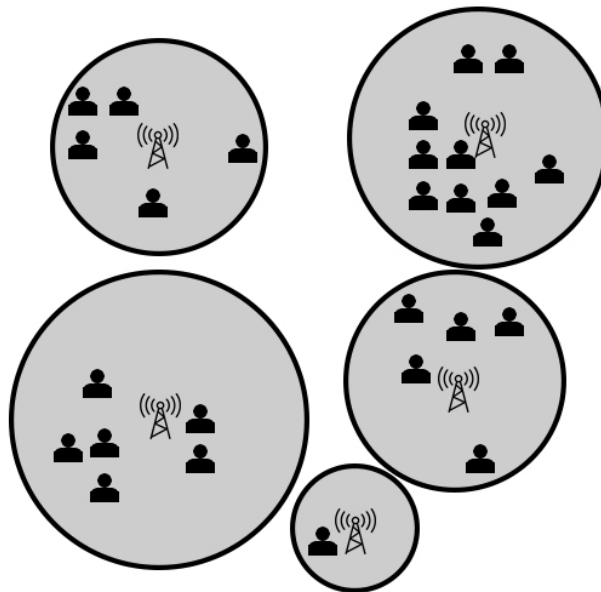
1 INTRODUÇÃO

O controle e planejamento dos recursos de uma cidade são importantes para o seu desenvolvimento. De acordo com Hall (2000), pode-se definir uma cidade inteligente como aquela que monitora e integra condições das suas infra-estruturas críticas, otimizando seus recursos e planejando ações preventivas enquanto maximiza os serviços para seus cidadãos.

Dentro desse ambiente, destaca-se o problema do posicionamento de ERBs (estações rádio-base), como mostra o trabalho de Amorim et al. (2020). Exigindo uma alta capilaridade nas redes, o problema se traduz na necessidade de um aumento de estações. Atrelado a isso, tem-se a chegada da rede 5G, com o objetivo de melhorar a eficiência no tráfego e comportando o crescente volume de informações, quando comparada com suas gerações anteriores. Além do posicionamento de ERBs, como citado acima, outros serviços públicos estão relacionados a problemas de cobertura, como por exemplo os serviços de iluminação pública. Todos esses problemas possuem como objetivo minimizar o custo ao cobrir uma certa região.

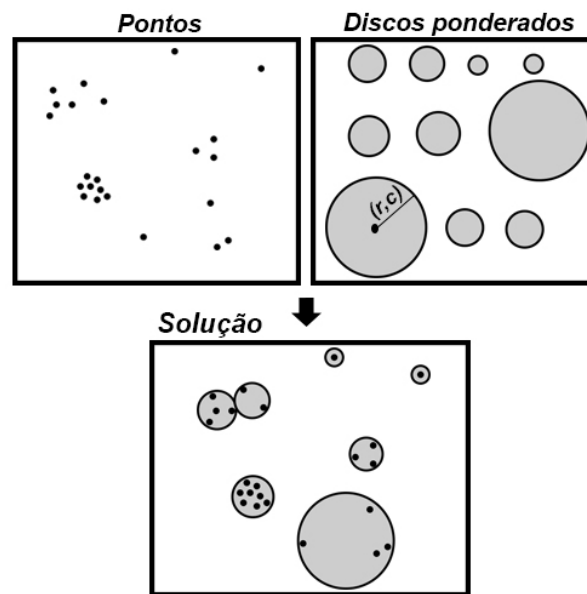
A partir da Figura 1, observa-se que o problema de posicionamento das estações rádio-base pode ser reduzido a um problema de cobrir pontos do \mathbb{R}^2 utilizando discos, onde os pontos representam as demandas e os discos a distribuição do sinal. Apesar das diferentes aplicações, todos esses problemas podem ser modelados como um problema da cobertura de pontos por discos ponderados, no espaço bidimensional.

Figura 1 – Posicionamento de antenas



A Figura 2 apresenta um exemplo do problema abordado neste trabalho, onde a partir de um conjunto de pontos e de discos ponderados, deseja-se realizar uma cobertura com o menor custo e suas posições. Assim, o conjunto solução, da melhor combinação, não necessariamente possuirá todos os discos da entrada.

Figura 2 – Exemplo do problema abordado



1.1 DEFINIÇÃO FORMAL DO PROBLEMA

Boa parte das aplicações de cobertura descritas anteriormente necessita da utilização de diferentes discos. Em que cada disco possui um raio de cobertura e um custo. O objetivo consiste em cobrir todos os pontos (demandas) como menor custo de implantação. Assim, o PROBLEMA DA COBERTURA DE PONTOS POR DISCOS PONDERADOS (PCPDP), abordado neste trabalho, e o PROBLEMA DA COBERTURA DE PONTOS POR DISCOS UNITÁRIOS, em Biniaz et al. (2017), são definidos com:

PROBLEMA DA COBERTURA DE PONTOS POR DISCOS PONDERADOS

Entrada: Dois conjuntos $P = (p_1, \dots, p_n)$ e $D = (d_1, \dots, d_m)$, sendo P o conjunto dos n pontos, onde $p_i = (x_i, y_i) \in \mathfrak{R}^2$, a serem cobertos e D o conjunto dos m discos, onde $d_j = (r_j, c_j)$ com r_j e $c_j \in \mathfrak{R}$ representando o raio de intensidade e o custo, respectivamente.

Objetivo: Encontrar um subconjunto $D' \subseteq D$ que cubra todos os pontos P com custo mínimo.

PROBLEMA DA COBERTURA DE PONTOS POR DISCOS UNITÁRIOS

Entrada: Conjunto P de n pontos no plano.

Objetivo: O problema solicita o número mínimo de discos de raio 1, unitário, que cobrem todos os pontos de P .

Assim, pode-se observar que o problema proposto nesse trabalho é uma generalização do problema abordado por Biniáz et al. (2017), que mostrou ser NP-completo. Dessa forma, é fácil observar que o PCPDP é NP-difícil. Para mais detalhes, sobre a NP-completude, recomenda-se o livro de Garey & Johnson (1979).

1.2 OBJETIVOS

O problema estudado nesse trabalho, possui grande relevância tanto no aspecto prático como teórico. Diante do fato do PCPDP ser NP-completo, justifica-se o desenvolvimento de algoritmos heurísticos voltados a esse contexto com o objetivo de gerar soluções em baixo custo computacional.

Este trabalho tem como objetivo principal o desenvolvimento de métodos algorítmicos para o problema da cobertura de pontos por discos ponderados. Para isso, será apresentada a modelagem do PCPDP por uma formulação matemática e o desenvolvimento de uma heurística híbrida.

De forma específica, serão utilizado dois métodos para obter a solução. O primeiro é a criação de uma solução exata, com o *CPLEX ILOG CPLEX Optimization Studio* (2020), a partir de uma formulação matemática. O segundo é a meta-heurística híbrida CMSA (*Construct, Merge, Solve & Adapt*) (Blum et al. 2016).

2 REVISÃO BIBLIOGRÁFICA

Nesta seção, será apresentada uma revisão da literatura com os principais trabalhos relacionados ao PCPDP. A revisão está dividida em algoritmos aproximativos e heurísticos.

2.1 ALGORITMOS APROXIMATIVOS

Gonzalez (1991) propõe um algoritmo com fator de aproximação $2(1 + 1/l)^{d-1}$, observando também, essa cobertura para outros formatos de conjunto, com uma variação sendo coberto por hiper retângulos ortogonais de tamanho fixo, além dos discos.

Carmi et al. (2007) apresentam um algoritmo 4-aproximativo e mostram a importância desse problema, como por exemplo, em projetos de redes sem fio.

No estudo feito por Biniáz et al. (2017), a respeito da cobertura de pontos por discos, é apresentado a prova da NP-completude considerando que os discos possuem características unitárias. Nele, é proposto um algoritmo 4-aproximativo que roda em tempo $O(n \log n)$ na norma Euclidiana L_2 , e apresentando como relacionar a norma L_2 com outras normas L_t , para $t \geq 1$.

No artigo de Ghosh et al. (2019), é realizado uma análise dos resultados obtidos a partir dos algoritmos aproximativos projetados nas últimas três décadas antes do seu ano de publicação. O seu objetivo é comparar, e investigar, o desempenho desses algoritmos em conjuntos de pontos massivos, como por exemplo a instância denominada monalisa que possui 100000 pontos. Além disso, apresenta-se um algoritmo simples 7-aproximativo para o PROBLEMA DA COBERTURA DE PONTOS POR DISCOS UNITÁRIOS.

2.2 HEURÍSTICAS E APLICAÇÕES

Com relação a heurísticas, ao conhecimento dos autores, não existe algoritmo heurístico para o problema de cobertura de pontos por discos. Dessa forma, será apresentada uma revisão de problemas similares aplicados a cidades inteligentes.

Relacionando este problema de cobertura com o posicionamento de estações rádio base, tem-se o trabalho de Amorim et al. (2020), que propõe uma implementação utilizando a meta-heurística BRKGA, um algoritmo genético baseado em chaves aleatórias, com o objetivo de alcançar esse balanço de forma ponderada. Com os resultados, foi possível provar a eficácia com 1,4%, em média, da cobertura ótima. Seu contexto é diante de áreas hexagonais, diferentemente de discos como o problema abordado neste trabalho.

Dentro do contexto de estações base, visando um algoritmo eficiente para cobrir uma região, o trabalho de Das et al. (2006) apresenta um algoritmo eficiente para este problema usando o diagrama de Voronoi. Levando como proposta de tema a comunicação móvel, o trabalho mostra como relacionar esse tipo de cobertura com a de pontos por discos, tendo como objetivo cobrir uma região por um determinado número de discos de raios iguais minimizando o custo.

O trabalho de Chen et al. (2022) aborda um contexto mais atual, o problema de planejamento da rede 5G. Utilizando o algoritmo *k-means*, minimizando o custo total de construção das novas estações base, o artigo considera as restrições como a distância mínima e tráfego de comunicação. Seu trabalho contribui para exemplificar a importância do problema de cobertura, considerando o custo, em aspectos práticos, levando em consideração a complexidade da comunicação nessa cobertura, diferentemente desse trabalho.

3 ALGORITMOS PARA O PCPDP

Nesta seção é apresentada uma formulação matemática para o PCPDP utilizando programação quadrática e uma heurística híbrida CMSA.

3.1 FORMULAÇÃO MATEMÁTICA

Dados P e D , os conjuntos de pontos e discos, em que cada disco $d_i \in D$ composto por (r_i, c_i) representando o raio e o custo de implementação do disco i , respectivamente, em que as variáveis de decisão do modelo serão:

- (x_i, y_i) — variáveis reais que denotam as posições do disco d_i no plano;
- z_i — variáveis binárias que valem 1 se o disco i está sendo usado, e 0 caso contrário.
- p_{ji} — variáveis binárias que valem 1 se o ponto j é atendido pelo disco i , e 0 caso contrário.

Além disso, M possui o valor 2147483647 definido de acordo máximo inteiro da linguagem de programação C++, (x_j, y_j) representa a posição do ponto p_j no plano e $dist(p_j, d_i) = ((x_j - x_i)^2 + (y_j - y_i)^2)^{1/2}$ a distância entre o ponto j e o disco i . A formulação proposta é descrita como:

$$\min \sum_{i=1}^{|D|} c_i z_i \quad (1)$$

$$dist(p_j, d_i)^2 \leq r_i^2 p_{ji} + M(1 - p_{ji}) \quad \forall j \in \{1, \dots, |P|\}, \forall i \in \{1, \dots, |D|\} \quad (2)$$

$$\sum_{i \in D} p_{ji} \geq 1 \quad \forall j \in \{1, \dots, |P|\}. \quad (3)$$

$$p_{ji} \leq z_i \quad \forall j \in \{1, \dots, |P|\}, \forall i \in \{1, \dots, |D|\} \quad (4)$$

$$x_i \in \mathfrak{R}, y_i \in \mathfrak{R}, z_i \in \{0, 1\} \quad \forall i \in \{1, \dots, |D|\} \quad (5)$$

$$p_{ji} \in \{0, 1\} \quad \forall j \in \{1, \dots, |P|\}, \forall i \in \{1, \dots, |D|\} \quad (6)$$

As Restrições (2) indicam que se o ponto j for coberto pelo disco i então a distância entre eles deve ser menor que r^2 . Nas Restrições (3), tem-se que todos os pontos devem ser cobertos por pelo menos um disco. As Restrições (4) apresentam a regra de que se o ponto está sendo coberto por um disco, esse disco precisa estar sendo utilizado na cobertura. Por fim, as Restrições (5) e (6) definem o domínio das variáveis.

3.2 CMSA

O CMSA é uma metaheurística híbrida voltada para problemas combinatórios desenvolvida por Blum et al. (2016), composto de 4 etapas: *Construct*, *Merge*, *Solve* e *Adapt*, que são executadas a cada iteração do algoritmo. Na primeira etapa, o *Construct*, é realizada a construção de diversas soluções para o problema. Em seguida, na etapa do *Merge*, as soluções construídas na etapa anterior são adicionadas, mediante um conjunto de regras, a um *pool* de soluções. Posteriormente, na etapa *Solve*, é utilizado um método exato sobre esse *pool* resultante, obtendo uma nova solução. Por fim, na etapa *Adapt*, algumas soluções pouco relevantes são removidas do *pool* de soluções.

Sendo P e D , os conjunto dos pontos e discos respectivamente, temos:

Algoritmo 1: CMSA

```

Entrada:  $P, D, age_{max}$ 
1  $S_{geral} = null; C' = \emptyset$ 
2 while  $time \leq limit$  do
3   for  $i = 1, \dots, n_a$  do
4     //Fase Construct (Seção 3.2.1)
5      $P' \leftarrow RandomSampleOfPoints(P)$ 
6      $D' \leftarrow SmallestCircle(P')$            // $D'$  é um subconjunto de discos de  $D$ 
7     //Fase Merge (Seção 3.2.2)
8     for  $all\ c \in D' \ \&\ c \notin C'$  do
9        $age[c] = 0$ 
10       $C' = C' \cup c$ 
11    end
12  end
13  //Fase Solve (Seção 3.2.3)
14   $S_{local} = ApplyCplex(C')$ 
15  if  $S_{local}$  is better than  $S_{geral}$  then
16     $S_{geral} = S_{local}$ 
17  end
18  //Fase Adapt (Seção 3.2.4)
19   $Adapt(C', S_{local}, age_{max})$ 
20 end
Saída:  $S_{geral}$ 

```

O algoritmo 1 descreve o CMSA, que recebe as entradas características do problema, sendo elas, o conjunto dos pontos e discos, e atribuindo um valor em age_{max} , que será utilizada como limiar para a categorização do conjunto resultante da fase *Merge*. Na linha 1, é determinada a variável S_{geral} , para armazenar o conjunto da melhor solução encontrada, e C' , para os discos resultantes da condição de cobertura.

Após esse primeiro momento, na linha 2, é dado o início a interação que realizará as 4 fases do CMSA, com a condição de parada determinada pelo tempo limite de execução do algoritmo. Construindo particionamentos do conjunto P , de forma aleatória, e executando a função de encontrar o menor disco que cobre todos os pontos desse particionamento, representado nas linhas 5 e 6. O Merge, realizado entre as linhas 8 e 11, relaciona o conjunto de discos D' provenientes do particionamento com D de entrada.

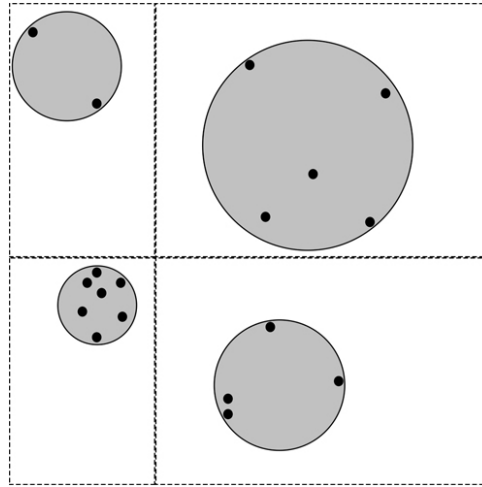
Dessa forma, tem-se uma solução exata recebendo o resultado dessa mesclagem, representados na linha 14, de característica combinatorial, com o objetivo de minimizar o custo. A melhor solução encontrada a cada interação do algoritmo, por meio da linha 16, é armazenada e substituída quando necessária. Por fim, temos a execução da fase Adapt, na linha 19, determinando quais elementos de C' continuará ou será descartado, criando um conjunto cada vez mais apto para a resolução do problema.

3.2.1 Construct

Nesta etapa, n_a soluções são geradas a cada iteração. O primeiro passo da construção é separar o conjunto de pontos P em subconjuntos P' . Para isso, é realizado o particionamento recursivo do plano originado de P em 4 quadrantes, para compor o conjunto P' . Com a região dividida em grids, cada subconjunto P' corresponde aos pontos na mesma células do grid.

Por fim, é aplicada a função `SmallestCircle` proposta em Welzl (1991), com uma metodologia recursiva onde retira aleatoriamente $p \in P$ até ficar com o conjunto vazio, adicionando o último ponto retirado como extremo do disco, e realizando a verificação de que se o próximo ponto está contido nesse disco, mantém a cobertura, ou o ponto torna-se o novo extremo do disco caso contrário. A aplicação possui o objetivo de encontrar a posição e o raio do menor disco que cobre todos os pontos em P' , resultando em uma tupla $R = (r, x, y)$, utilizando a ferramenta desenvolvida em *Project Nayuki* (2020).

A Figura 3 mostra um exemplo de execução em 4 células distintas da grade.

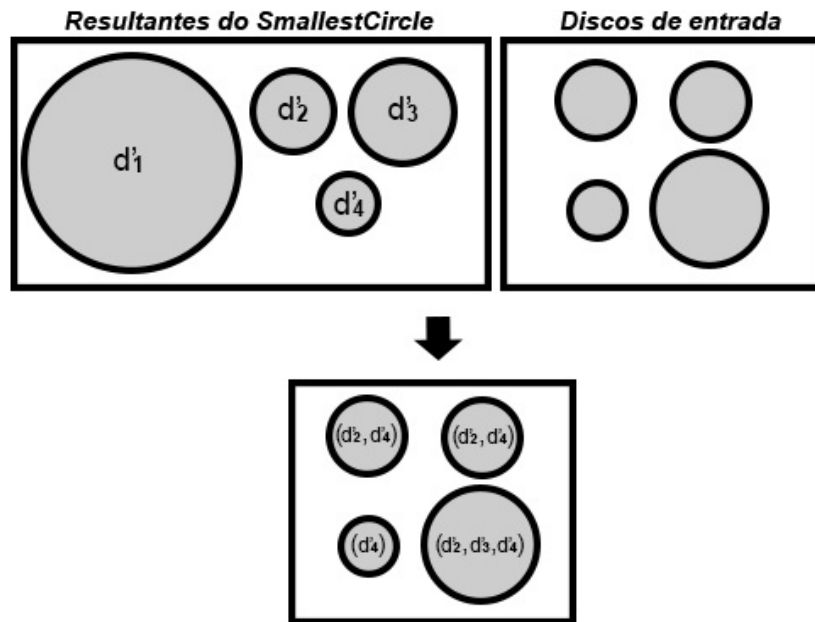
Figura 3 – Particionamento recursivo de P 

Em seguida, seja $D_r = \{D_r^1, \dots, D_r^\ell\}$ o conjunto de todos os discos de D com raio maior o igual a r . Seja c_r^i o custo do disco D_r^i . A fase *Construct*, ao final, deve retornar o conjunto D' definido como:

$$D' = \cup_{i=1}^{\ell} (D_r^i, c_r^i, x, y)$$

Em outras palavras, o conjunto D' é formado por todas as tuplas formadas pela combinação dos discos de raio maior que r e a posição (x, y) . A Figura 4 mostra a construção de D' para cada célula do exemplo anterior.

Para garantir uma diversidade de soluções, a etapa de particionamento recursivo (criação de grid) é feita de forma estocástica. Em outras palavras, as divisões produzem quatro quadrantes de tamanhos distintos. Com isso, cada iteração deve produzir soluções distintas.

Figura 4 – Exemplo de construção de D' 

3.2.2 Merge

Na etapa *Merge*, cada novo conjunto D' é inserido no *pool* de soluções C' . Nesta etapa, cada nova tupla é verificada se já pertence a C' . Cada tupla em D' possui um valor do vetor *ages* associado. Inicialmente, o valor de cada nova tupla é atribuído o valor zero. Este valor representa a quantidade de iterações que a tupla está no *pool* C' .

3.2.3 Solve

A fase *Solve* consiste em resolver um problema combinatório de encontrar um subconjunto de C' de custo mínimo que cubra todos os pontos e não tenha discos repetidos.

Seja $C' = \{C_1, \dots, C_k\}$ o *pool* de soluções. Cada solução C_i corresponde a uma tupla (d_i, c_i, x_i, y_i) , em que d_i é um disco de D , c_i o seu custo e (x_i, y_i) a posição em que o disco foi alocado. Seja o conjunto $C_p \subseteq C'$ o conjunto formado por todas as tuplas que cobrem o ponto $p \in P$. Seja $G_d \subseteq C'$ o conjunto de tuplas que contém o disco $d \in D$.

O modelo ILP utilizado no *Solve* é:

$$\min \sum_{i=1}^k c_i x_i \quad (7)$$

$$\sum_{j \in G_d} x_j \leq 1 \quad \forall d \in D. \quad (8)$$

$$\sum_{j \in C_p} x_j \geq 1 \quad \forall p \in P. \quad (9)$$

$$x_i \in \{0, 1\} \quad \forall i \in 1, \dots, k \quad (10)$$

em que, x_i vale um se e somente se a tupla C_i for utilizada na solução ótima. Neste caso, o disco d_i é alocado na posição (x_i, y_i) . As Restrições (8) proíbem que dois discos idêntico sejam utilizados. E as Restrições (9) obrigam que todo ponto $p \in P$ seja coberto por pelo menos um disco. Por fim, as Restrições (10) definem os domínios das variáveis.

3.2.4 Adapt

A categorização de $c \in C'$, é importante para determinar os elementos que mais fazem parte do conjunto de soluções S_{local} , encontradas pelo algoritmo exato, isso é avaliado nessa fase mediante os valores de *age*, relacionados a cada c , tendo como limiar o age_{max} configurado anteriormente.

Essa é a última fase dentro da iteração, resultando em uma filtragem e adaptação das soluções, visando a busca dos elementos em S_{local} mais aptos, gerando uma tendência aos melhores resultados encontrados.

4 RESULTADOS

Esta seção apresenta os resultados computacionais das soluções algorítmicas propostas para o PCPDP. Os algoritmos foram desenvolvidos em C++, utilizando o Visual Studio 2019 e executados numa máquina AMD FX(tm)-8300 Eight-Core Processor 3.30 GHz e 8Gb de memória RAM, no sistema operacional Windows 10 64 bits.

4.1 INSTÂNCIAS

Como não é conhecido um conjunto de instâncias teste para o problemas, as instâncias utilizadas nos testes consistem em pontos e discos provenientes da plataforma CVRPLIB (*Capacitated Vehicle Routing Problem Library*), onde os pontos são representados por (x, y) , e os discos por (r, c) . Nessas instâncias os discos foram criados de forma aleatória com os pesos seguindo uma distribuição proporcional ao raio. A Tabela 1 apresenta as configurações das instâncias, com n-pontos e n-discos representando a quantidade de pontos e discos, respectivamente.

Tabela 1 – Instâncias.

nome	n-pontos	n-discos
P-n16-k8	16	14
P-n19-k2	18	18
P-n20-k2	19	19
P-n22-k2	21	21
E-n23-k3	22	22
B-n31-k5	29	30
A-n32-k5	31	31
B-n35-k5	33	33
A-n37-k5	36	36
A-n39-k5	38	38
P-n40-k5	39	39
A-n45-k6	44	44
B-n78-k10	76	77
A-n80-k10	79	79
M-n101-k10	101	68

4.2 EXPERIMENTOS

Os experimentos realizados neste trabalho, com o objetivo de propor soluções para o Problema da Cobertura de Pontos por Discos Ponderados, foram divididos em duas categorias, um relacionado ao algoritmo Exato e outro ao CMSA. De acordo com a capacidade computacional da máquina utilizada nos experimentos, foi determinado limites de execução, onde para o Exato, desenvolvido em CPLEX, foi executado com um tempo limite determinado em 1 hora, e para o algoritmo híbrido CMSA com um limite de $100 + |P|$ iterações, com 10 repetições quando $|P| > 100$, e 20 caso contrário.

Os resultados, representados na Tabela 2, apresentam a solução e o bound encontrado pelo algoritmo Exato, e o tempo em segundos. Para os resultados do CMSA, os valores *best* da melhor solução encontrada, a média das soluções, o *ttb** (*time-to-best-corrigido*) representando o tempo médio das melhores soluções encontradas de cada repetição em segundos, e o *gap*(%) do *best* comparado ao ótimo do exato.

Tabela 2 – Resultados Exato x CMSA.

instância	Exato		CMSA			
	solução (bound)	tempo (s)	best	média	ttb* (s)	gap (%)
P-n16-k8	280*	1.95	280	280	0.23	0
P-n19-k2	260*	7.11	260	260	0.08	0
P-n20-k2	260*	9.83	260	260	0.09	0
P-n22-k2	260*	10.2	260	260	0.13	0
E-n23-k3	750*	0.45	750	750	0.05	0
B-n31-k5	320*	28.13	320	320	1.62	0
A-n32-k5	760 (320)	3599.39	740	744	10.95	0
B-n35-k5	450 (95)	3608.5	420	420	1.93	0
A-n37-k5	- (56.78)	3407.31	960	971.5	5.23	-
A-n39-k5	- (90.66)	3620.39	1070	1093	6.39	-
P-n40-k5	41*	7.5	41	44	14.12	0
A-n45-k6	- (115)	3649.69	1090	1130	10.47	-
B-n78-k10	- (100)	3613.81	640	688	76.86	-
A-n80-k10	- (90)	3611.73	1020	1062	81.88	-
M-n101-k10	600 (425.8)	3651.5	600	760	226.52	0

Dentre as 15 instâncias, o exato encontrou a otimalidade em 7, percebendo um tamanho de até 29 pontos, com a exceção da instância P-n40-k5 que possui 39, onde a distribuição dos pontos no plano influencia na busca das combinações que compõe a solução. Encontrou-se também, solução viável em 3 instâncias, utilizando todo o tempo limite disponível, e para 5 não retornou solução.

O CMSA encontrou todos os ótimos conhecidos em um baixo custo computacional, observando o gap de 0%. Foram geradas soluções para todas as instâncias selecionadas no trabalho, com um tempo de execução viável.

5 CONCLUSÃO

Analisando os resultados apresentados, pode-se observar que a aplicação do CMSA apresentou boas soluções para todas as instâncias selecionadas em um baixo tempo computacional para o problema de Cobertura de Pontos por Discos Ponderados. Além disso, em todas as instâncias com ótimo conhecido o algoritmo encontrou a solução ótima. O método exato, diante da complexidade do problema, apresentou resultados satisfatórios para instâncias com pequena quantidade de pontos.

Diante das instâncias selecionadas no trabalho, observa-se que as características de distribuição dos pontos no espaço, além do seu tamanho, influenciam de forma direta no desempenho dos algoritmos apresentados, uma vez que, afetam as possibilidades de combinação dentro do conjunto de soluções.

De acordo com a pesquisa desse trabalho, pode-se concluir que é possível aplicar CMSA para o problema, com uma eficiência satisfatória, e possibilitando o estudo dessa aplicação em problemas da sociedade, como nos exemplos citados de cidades inteligentes, trazendo alternativas de resolução, e contribuindo com a modelagem da meta-heurística híbrida.

Entende-se que há espaço para melhoria em trabalhos futuros, como um estudo dos melhores parâmetros utilizados no CMSA, para reduzir as dificuldades quando o algoritmo é exposto a diversas configurações e tamanhos de instâncias.

REFERÊNCIAS

- Amorim, L., Vassalo, D., Pereira, G. & Pinheiro, R. (2020), Posicionamento de estações-base em uma cidade simulada usando um algoritmo genético de chaves aleatórias enviesadas, *in* ‘Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional’, SBC, Porto Alegre, RS, Brasil, pp. 984–995.
URL: <https://sol.sbc.org.br/index.php/eniac/article/view/9351>
- Biniáz, A., Liu, P., Maheshwari, A. & Smid, M. (2017), ‘Approximation algorithms for the unit disk cover problem in 2d and 3d’, *Computational Geometry* 60, 8–18. The Twenty-Seventh Canadian Conference on Computational Geometry August 2015.
URL: <http://www.sciencedirect.com/science/article/pii/S092577211630030X>
- Blum, C., Pinacho, P., López-Ibáñez, M. & Lozano, J. A. (2016), ‘Construct, merge, solve & adapt a new general algorithm for combinatorial optimization’, *Computers & Operations Research* 68, 75–88.
URL: <http://www.sciencedirect.com/science/article/pii/S0305054815002452>
- Carmi, P., Katz, M. J. & Lev-Tov, N. (2007), Covering points by unit disks of fixed location, *in* T. Tokuyama, ed., ‘ALGORITHMS AND COMPUTATION’, Vol. 4835 of *Lecture Notes in Computer Science*, Tohoku Univ, Grad Sch Info Sci; Tohoku Univ, Res Inst Elect Comm; New Horizons Comp, Sci Res Priority Areas; Telecommun Advancement Fdn; Kayamori Fdn Informat Sci Advancement; Support Ctr Adv Telecommun Technol Res Fdn; Sendai Tourism & Convent Bur, pp. 644+. 18th International Symposium on Algorithms and Computations (ISAAC 2007), Sendai, JAPAN, DEC 17-19, 2007.
- Chen, J., Tian, J., Jiang, S., Zhou, Y., Li, H. & Xu, J. (2022), ‘The allocation of base stations with region clustering and single-objective nonlinear optimization’, *MATHEMATICS* 10(13).
- Das, G. K., Das, S., Nandy, S. C. & Sinha, B. P. (2006), ‘Efficient algorithm for placing a given number of base stations to cover a convex region’, *JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING* 66(11), 1353–1358.
- Garey, M. R. & Johnson, D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*, first edition edn, W. H. Freeman.

URL: <http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455>

Ghosh, A., Hicks, B. & Shevchenko, R. (2019), Unit disk cover for massive point sets, in I. Kotsireas, P. Pardalos, K. E. Parsopoulos, D. Souravlias & A. Tsokas, eds, ‘Analysis of Experimental Algorithms’, Springer International Publishing, Cham, pp. 142–157.

Gonzalez, T. F. (1991), ‘Covering a set of points in multidimensional space’, *Information Processing Letters* 40(4), 181–188.

URL: <https://www.sciencedirect.com/science/article/pii/002001909190075S>

Hall, P. (2000), ‘Creative cities and economic development’, *Urban Studies* 37(4), 639–649.

URL: <https://doi.org/10.1080/00420980050003946>

ILOG CPLEX Optimization Studio (2020), <https://www.ibm.com/products/ilog-cplex-optimization-studio>. Acessado: 31/01/2020.

Project Nayuki (2020), <https://www.nayuki.io/page/smallest-enclosing-circle>. Acessado: 24/07/2022.

Welzl, E. (1991), Smallest enclosing disks (balls and ellipsoids), in H. Maurer, ed., ‘New Results and New Trends in Computer Science’, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 359–370.