UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Cobertura Maximal de Pontos por Elipses

Danilo Françoso Tedeschi

São Carlos – SP

Cobertura Maximal de Pontos por Elipses

Danilo Françoso Tedeschi

Orientadora: **Profa. Dra. Marina Andretta**

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Bacharel em Computação.
Área de Concentração: Otimização

USP – São Carlos
Novembro de 2017

Tedeschi, Danilo Françoso
Cobertura Maximal de Pontos por Elipses / Danilo Françoso Tedeschi. - São Carlos - SP, 2017.
45 p.; 29,7 cm.

Orientadora: Marina Andretta.
Monografia (Graduação) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP, 2017.

1. Otimização. 2. Cobertura maximal de pontos por elipses. I. Andretta, Marina. II. Instituto de Ciências Matemáticas e de Computação (ICMC/USP). III. Título.

Este trabalho é dedicado à minha família, em especial à minha mãe que sempre me apoiou em tudo e me ajudou sempre que precisei.

AGRADECIMENTOS

Agradeço à minha orientadora por ter sido paciente comigo e ter me deixado mudar de tema no meio do semestre, além de claro ter me orientado no desenvolvimento deste trabalho. Agradeço também aos professores que tive durante a graduação e aos amigos que fiz ao longo do meu tempo aqui. Por último, agradeço à minha família pela ótima educação que tive e por proporcionar a oportunidade de me dedicar integralmente aos estudos por tanto tempo.

RESUMO

TEDESCHI, D. F.. **Cobertura Maximal de Pontos por Elipses**. 2017. 45 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Cobertura maximal de pontos por elipses é um problema de otimização em que deseja-se posicionar elipses no plano com o objetivo de cobrir pontos de demanda. Dentre todas as possíveis soluções, é de interesse encontrar aquela que maximize uma função que depende do valor dos pontos cobertos e do custo das elipses utilizadas. Este trabalho estudará a versão do problema em que as elipses são paralelas aos eixos cartesianos. Um algoritmo que resolve o problema de maneira determinística será proposto e resultados serão comparados com os de métodos desenvolvidos anteriormente.

Palavras-chave: Otimização, Cobertura maximal de pontos por elipses.

ABSTRACT

TEDESCHI, D. F.. **Cobertura Maximal de Pontos por Elipses**. 2017. 45 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Maximal covering of points using ellipses is an optimization problem which one wants to place ellipses on the plane to cover demand points, such that a function depending on the value of the covered points and on the cost of the ellipses that have been used is maximized. This work will present a study on the version of the problem where ellipses are parallel to the coordinate axes. An algorithm that solves the problem deterministically will be proposed and the results will be compared with previous works.

Key-words: Optimization, Maximal covering of points using ellipses.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

SUMÁRIO

Capítulo 1

INTRODUÇÃO

1.1 Motivação e Contextualização

O primeiro trabalho sobre o problema de cobertura maximal de pontos por elipses encontrado na literatura é feito por (CANBOLAT; MASSOW, 2009). A motivação apresentada neste primeiro estudo é a resolução do problema de se determinar as posições de instalação de antenas para que o maior número de cidades seja coberto pelo sinal transmitido. O caso em que uma empresa de internet *wireless* da Califórnia distribui sinal à algumas cidades de uma região utilizando três antenas com alcance elíptico é utilizado como exemplo prático.

Pela característica de não impor que todos os pontos sejam cobertos e, também, por ter como objetivo a maximização do lucro obtido pela cobertura ótima, o problema tratado neste trabalho é categorizado como um *Maximal Covering Location Problem* (MCLP) (CANBOLAT; MASSOW, 2009). Introduzido em (CHURCH; REVELLE, 1974), MCLPs são problemas em que deseja-se maximizar a cobertura de um conjunto de demanda por elementos de um conjunto de recursos. A localidade, de alguma forma, é utilizada para se estabelecer a cobertura.

A versão planar do problema, em que os elementos do conjunto de demanda e os recursos podem ser posicionados em qualquer ponto do plano, é pela primeira vez abordada em (MEHREZ; STULMAN, 1982). Inicialmente, MCLP foi definido como um problema de otimização em grafos finitos. Tanto os elementos do conjunto de demanda quanto os recursos podiam estar apenas localizados em vértices do grafo (YOUNIES; WESOLOWSKY, 2004). Cobertura maximal de pontos por elipses é um MCLP planar em que os recursos possuem cobertura em forma de elipse. Um caso particular é estudado em (MEHREZ; STULMAN, 1982; CHURCH, 1984), no qual os recursos cobrem pontos localizados a uma distância menor do que um valor definido, isto é, os recursos possuem cobertura em forma de circunferência.

1.2 Objetivos

O objetivo deste trabalho é realizar e apresentar os resultados das seguintes atividades:

- Revisão bibliográfica: é feita uma revisão dos trabalhos encontrados sobre cobertura maximal de pontos por elipses, também são revisados alguns conceitos de geometria que são utilizados tanto nas definições do problema quanto nos métodos estudados.

- Elaboração de algoritmo: um novo método de resolução do problema que obtém soluções ótimas é proposto. Instâncias elaboradas em outros trabalhos são utilizadas para que uma comparação de resultados obtidos em trabalhos anteriores seja feita.

1.3 Organização

O Capítulo 2 introduz conceitos básicos que serão utilizados no decorrer do texto. Lá também é definido o significado de coberturas equivalentes, bem como é apresentada uma propriedade importante utilizada na elaboração do algoritmo no capítulo seguinte. No Capítulo 3 são feitas a revisão bibliográfica, a proposta de um novo método de resolução e a comparação de resultados obtidos. O Capítulo 4 apresenta um resumo das atividades realizadas bem como possíveis trabalhos futuros. Uma análise do curso de graduação também é feita nesse capítulo.

Capítulo 2

MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS

2.1 Conceitos Básicos

Nesta seção serão apresentados alguns conceitos básicos que serão utilizados no restante do texto.

2.1.1 Elipse

Sejam f_1 e $f_2 \in \mathbb{R}^2$ pontos, $2c \in \mathbb{R}$ a distância euclidiana entre eles e $d \in \mathbb{R}$, uma constante, tal que $d > c$. Uma elipse é o lugar geométrico dos pontos $x \in \mathbb{R}^2$ que satisfazem: $dist(x, f_1) + dist(x, f_2) = 2d$, com $dist(x, y)$ a distância euclidiana entre os pontos x e y. Os pontos f_1 e f_2 são chamados de focos da elipse (BOULOS; CAMARGO, 2006).

No caso de elipses paralelas aos eixos cartesianos, uma equação chamada reduzida pode ser obtida (BOULOS; CAMARGO, 2006):

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-x_0)^2}{b^2} = 1. \tag{2.1}$$

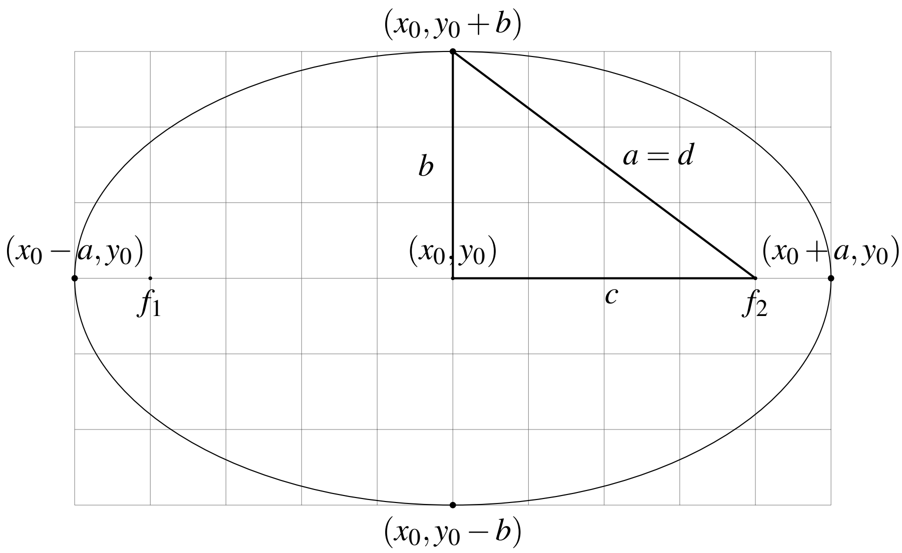
Note que os parâmetros utilizados em (2.1) (a, b, x_0, y_0) são unicamente determinados pelos focos, a constante d e o fato da elipse ser paralela aos eixos. Isto é melhor destacado pela Figura 1, onde pode ser visto que (a, b, c) formam um triângulo retângulo e (x_0, y_0) é o centro da elipse, que, por sua vez, é o ponto médio do segmento $f_1 f_2$.

2.1.2 Cobertura de um ponto por uma elipse

Um ponto é dito ser coberto por uma elipse se o mesmo pertence à região delimitada por essa elipse. Os pontos com soma das distâncias para os focos menor ou igual a $2d$ formam tal região. Em particular, todos os pontos (x, y) em que (2.2) é satisfeita, descrevem a região delimitada por uma elipse paralela aos eixos (CANBOLAT; MASSOW, 2009).

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} \leq 1. \tag{2.2}$$

Figura 1 – Gráfico de uma elipse paralela aos eixos



Fonte: Elaborada pelo autor.

2.1.3 Posição de uma elipse que contém dois pontos

Suponha que sejam conhecidos todos os parâmetros, exceto o centro, de uma elipse paralela aos eixos. Dados dois pontos $p = (p_x, p_y)$ e $q = (q_x, q_y) \in \mathbb{R}^2$, deseja-se determinar todos os centros que essa elipse pode ser posicionada, de tal maneira que ela contenha os pontos p e q. Em outras palavras, sejam $a, b \in \mathbb{R}$, $p, q \in \mathbb{R}^2$, deseja-se encontrar todos os pontos $(x, y) \in \mathbb{R}^2$, tal que, o sistema de equações (2.3) seja satisfeito. Será denotado como $\Phi(a, b, p, q)$ esse conjunto solução.

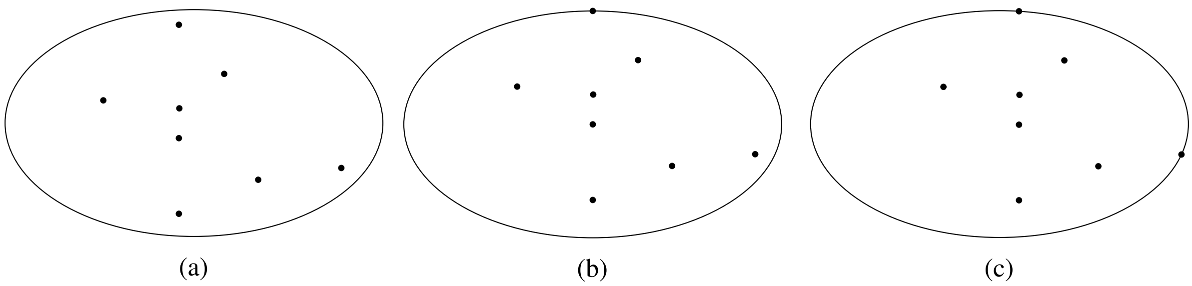
$$\begin{cases} \frac{(p_x - x)^2}{a^2} + \frac{(p_y - y)^2}{b^2} = 1 \\ \frac{(q_x - x)^2}{a^2} + \frac{(q_y - y)^2}{b^2} = 1 \end{cases} \tag{2.3}$$

Detalhes de como determinar $\Phi(a, b, p, q)$ são dados no Apêndice A. Lá também é mostrado que (2.3) possui no máximo 2 soluções.

2.2 Coberturas equivalentes

Sejam $E_{a,b}$ uma elipse paralela aos eixos e $X = \{x_1, ..., x_m\}$, $m > 1$, um conjunto de pontos. Defini-se Λ como o conjunto de todos os pontos $y \in \mathbb{R}^2$, tais que, $E_{a,b}$ com centro em y cobre todos os pontos de X. Isto é, Λ contém todas as posições em que $E_{a,b}$, quando posicionada, cobre todos os pontos do conjunto X. Se $z_1, z_2 \in \Lambda$, diz-se que tais pontos definem coberturas equivalentes de X.

Figura 2 – Três coberturas equivalentes



Fonte: Elaborada pelo autor.

2.2.1 Cobertura com dois pontos na elipse

Se $|\Lambda| \geq 1$, então existe uma cobertura de X que contém pelos menos dois elementos de X em $E_{a,b}$. Em outras palavras, todo conjunto Λ não-vazio possui um elemento que define uma cobertura com pelo menos dois pontos na elipse.

A Figura 2 ilustra, de maneira intuitiva, os argumentos que podem ser utilizados para demonstrar a proposição acima.

Em Figura 2a é exibida uma cobertura na qual nenhum dos pontos pertence à elipse. A Figura 2b, com um ponto na elipse, é obtida movendo o centro de $E_{a,b}$ para baixo. Para se obter dois pontos na elipse, ilustrado pela Figura 2c, é necessário fixar o primeiro ponto p, que já pertence à elipse, e analisar todos os centros y, em que $E_{a,b}$, centrada em y, contém p. Isso pode ser feito movimentando a elipse de maneira contínua[1]. Como inicialmente todos os pontos estão cobertos, em algum momento um segundo ponto atingirá a fronteira da região.

Com a ressalva de que o estado em que dois pontos estão na elipse pode ser alcançado antes do último passo, o processo descrito acima pode ser generalizado para qualquer elipse e conjunto de pontos.

[1] Os centros da elipse que contém o primeiro ponto formam uma outra elipse.

Capítulo 3

DESENVOLVIMENTO

3.1 O Problema

Sejam n pontos de demanda no plano e m elipses dados, em que cada ponto possui um ganho associado e cada elipse possui um custo. Deseja-se posicionar k elipses, $k \leq m$, de maneira a maximizar o lucro total, sendo que o lucro total é definido como a soma dos ganhos dos pontos cobertos menos o custo das k elipses utilizadas.

Os n pontos de demanda são dados por $p_1, \ldots, p_n \in \mathbb{R}^2$, com $p_i = (p_i^x, p_i^y)$, o ganho associado do i-ésimo ponto é w_i, com $w_i \geq 0$. As m elipses são paralelas aos eixos cartesianos e identificadas pelos parâmetros a_i, b_i, $i \in \{1, \ldots, m\}$. O custo da i-ésima elipse é dado por $\tilde{w}_i$, com $\tilde{w}_i \geq 0$.

O posicionamento de uma elipse selecionada é dado pela posição de seu centro $c_j = (c_j^x, c_j^y)$, $j \in \{1, \ldots, m\}$. A j-ésima elipse é selecionada se j pertence ao conjunto $J \subset \{1, \ldots, m\}$, tal que, $|J| = k$. Com isso, definindo o conjunto X que representa os pontos cobertos:

$$X = \left\{ i \in \{1, \ldots, n\} : \exists j \in J, \text{ tal que, } \frac{(p_i^x - c_j^x)^2}{a^2} + \frac{(p_i^y - c_j^y)^2}{b^2} \leq 1 \right\},$$

uma solução do problema pode ser descrita pela tripla (J, X, c) que satisfaz as condições acima. Note que, dado J e c, X pode ser obtido trivialmente; dado J e X, determinar c pode ser modelado como um problema de otimização (ANDRETTA; BIRGIN, 2013). Encontrar a solução ótima é, portanto, equivalente a encontrar (J^*, c^*) que maximize:

$$f(J, c) = \sum_{i \in X} w_i - \sum_{i \in J} \tilde{w}_i.$$

3.2 Revisão Bibliográfica

Cobertura maximal de pontos por elipses é modelado em (CANBOLAT; MASSOW, 2009) como um problema de Programação Não-Linear Inteiro Misto. O modelo apresentado tem como variáveis a determinar: o conjunto de elipses utilizadas bem como seus centros e o conjunto de pontos cobertos. Os métodos empregados se mostraram ineficientes, sendo que, para algumas instâncias nenhum resultado foi obtido em uma hora de tempo de execução. *Simulated Annealing* é então utilizado no desenvolvimento de um método heurístico que, em média, nos resultados apresentados, aproximou-se da solução ótima.

Em (ANDRETTA; BIRGIN, 2013) o problema é resolvido de maneira determinística e resultados são obtidos para todas instâncias elaboradas em (CANBOLAT; MASSOW, 2009) com tempo de execução viável. Por mais que possam existir infinitas combinações de posições de centros das elipses que produzam uma solução ótima, o número de combinações de conjunto de pontos cobertos e elipses utilizadas é finito[1]. É a partir desta observação que o método é desenvolvido.

Sendo assim, para cada elipse, todos os possíveis conjuntos de pontos cobertos são encontrados. Fixado um conjunto X, um simples problema de otimização é resolvido para descobrir se existe um centro tal que a elipse nele posicionada cobre o conjunto X. Para não ser necessário enumerar todos estes conjuntos, é argumentado que se X é coberto, não é mais necessário avaliar nenhum de seus subconjuntos.

Então, são construídas, para cada elipse, uma estrutura de dados eficiente que armazena esses possíveis conjuntos de pontos cobertos. Então, todos os subconjuntos de elipses de cardinalidade k são avaliados como sendo o conjunto J. A resposta é obtida para cada um deles e o que obtém melhor valor é reportado como a solução ótima.

Em (ANDRETTA; BIRGIN, 2013) também é apresentada a versão do problema onde as elipses podem ser livremente rotacionadas. Para esta versão, o algoritmo determinístico elaborado não mostrou ser eficiente e um método heurístico foi utilizado.

3.3 Método proposto

Primeiramente, um algoritmo será proposto para a versão do problema onde $k = 1$, ou seja, apenas uma elipse pode ser utilizada. A partir dessa primeira versão, o método para qualquer $k \leq m$ será desenvolvido.

3.3.1 Caso em que $k = 1$

Além de produzir uma resposta para o problema com a restrição $k = 1$, o algoritmo proposto nesta seção também produzirá todas as coberturas que possam ser utilizadas na solução ótima quando $k > 1$.

Suponha que a elipse utilizada na solução ótima seja a de índice l, denotada por E_{a_l,b_l}, e que o conjunto de pontos cobertos nessa solução seja U_l. Suponha também que $|U_l| > 1$.

Como visto no Capítulo 2, existe uma cobertura de U_l com pelo menos dois pontos na borda de E_{a_l,b_l}. O Algoritmo 1 utiliza-se desse fato e verifica para cada par de pontos (p_i, p_j), todas as coberturas definidas por $\Phi(a_l, b_l, p_i, p_j)$. Note que, como todas coberturas com dois pontos na borda da elipse são verificadas, alguma delas será uma cobertura de U_l.

[1] O número de pares (J, X) viáveis é menor ou igual à $2^n \times \binom{m}{k}$.

A função $GeraCoberturas(P,a,b)$, apresentada no Algoritmo 1, é responsável por essa construção de coberturas. Ela recebe um conjunto de pontos P e os parâmetros de uma elipse $E_{a,b}$. O seu retorno pode ser separado em alguns casos: primeiro, são consideradas todos os subconjuntos de P de tamanho um como possíveis coberturas, isso é feito para atender o caso em que a solução ótima cobre apenas um ponto ($|U_l| = 1$). Depois, todos os subconjuntos de P, possíveis soluções ótimas, em que pelo menos dois pontos pertencem à $E_{a,b}$. Por último, se P for um conjunto vazio, a função retorna um conjunto contendo o conjunto vazio, isso significa que a única cobertura possível é a vazia. Isso é necessário para que tanto o Algoritmo 2 quanto o Algoritmo 3 consigam construir uma solução em casos, nos quais não há pontos a serem cobertos.

Seja Υ_n o número de coberturas retornadas pela função $GeraCoberturas(P,a,b)$ para uma elipse. Como são consideradas n possíveis coberturas de apenas um ponto e, como mostrado no Apêndice A, $|\Phi(a_l,b_l,p_i,p_j)| \leq 2$, é possível obter um limitante superior para Υ_n:

$$\Upsilon_n = n + \sum_{1\leq i<j\leq n} |\Phi(a_l,b_l,p_i,p_j)| \leq n + \sum_{1\leq i<j\leq n} 2 = n + n(n-1) = n^2.$$

Logo, $\Upsilon_n = O(n^2)$. Determinar o conjunto X de pontos cobertos para cada uma das Υ_n coberturas é feito em $O(n)$ no Algoritmo 1. Portanto, para cada elipse, como cada uma das Υ_n coberturas são verificadas como possíveis soluções ótimas, a complexidade de tempo do algoritmo para encontrar a cobertura ótima é $O(n^3)$. Assim, a complexidade de tempo total do Algoritmo 2 é $O(mn^3)$.

3.3.2 Caso geral

Para um conjunto $J = \{j_1,\ldots,j_k\}$ de elipses fixado, a solução ótima pode ser encontrada avaliando-se todas as possibilidades de conjunto de pontos que cada uma das elipses consegue cobrir. Utilizando a função $GeraCoberturas$, isso é feito recursivamente: dado que as elipses de índices $j_1,\ldots,j_{h-1}$ foram posicionadas e até o momento X seja o conjunto de pontos cobertos, avaliando-se todos os possíveis conjuntos retornados por $GeraCoberturas(P\setminus X, a_{j_h}, b_{j_h})$, a resposta ótima será encontrada para h elipses, dado a configuração inicial das $h-1$ elipses.

O Algoritmo 3 implementa essa ideia na função $solve(P,E,W,\tilde{W},J,X,l)$, no qual l indica que as elipses de índices $1,\ldots,l-1$ estão fixadas, cada uma podendo estar no conjunto J como utilizada ou não. Em $solve(P,E,W,\tilde{W},J,X,l)$ é, então, fixada a l-ésima elipse. Se ela é utilizada e posicionada de maneira a cobrir o conjunto X' de pontos, $solve(P,E,W,\tilde{W},J\cup\{l\},X\cup X',l+1)$ é chamada (os possíveis conjuntos X' são retornados por $GeraCoberturas(P\setminus X,a_l,b_l)$). Também é verificada a possibilidade da l-ésima elipse não ser utilizada, neste caso $solve(P,E,W,\tilde{W},J,X,l+1)$ é chamada. No caso base ($l = m+1$), é verificado se há exatamente k elipses utilizadas na solução construída. Se sim, o seu valor é retornado, caso contrário um valor menor do que

Algoritmo 1: Função $GeraCoberturas(P,a,b)$

Input: Conjunto de pontos $P = \{p_1, ..., p_n\}$, uma elipse de parâmetros a, b

Output: Um conjunto A composto de subconjuntos de P, que são possíveis conjuntos cobertos pela elipse $E_{a,b}$ numa solução ótima.

```
Function GeraCoberturas(P, a, b) is
    if $P = \emptyset$ then                      // A única cobertura possível é a vazia
        return $\{\emptyset\}$
    end
    $A \leftarrow \emptyset$
    for $i \leftarrow 1$ to $n-1$ do
        $A \leftarrow A \cup \{\{i\}\}$   // Caso onde a solução ótima cobre apenas um ponto
        for $j \leftarrow i+1$ to $n$ do
            foreach $(c_x, c_y) \in \Phi(a,b,p_i,p_j)$ do        // No máximo duas iterações
                $X \leftarrow \emptyset$
                for $h \leftarrow 1$ to $n$ do
                    if $\frac{(p_h^x - c_x)^2}{a^2} + \frac{(p_h^y - c_y)^2}{b^2} \leq 1$ then
                        $X \leftarrow X \cup \{h\}$
                    end
                end
                $A \leftarrow A \cup \{X\}$
            end
        end
    end
    return $A$
end
```

qualquer outro ($-\infty$ [2]) é retornado. Para se obter o valor da solução ótima atraves do Algoritmo 3, basta que $solve(P,E,W,\tilde{W},\emptyset,\emptyset,1)$ seja chamada.

Tanto o Algoritmo 2 quanto o Algoritmo 3 obtêm apenas o valor da solução ótima, pequenas adaptações podem ser feitas para que J^* e c^* sejam também incluídos na resposta. Algumas melhorias foram feitas na implementação utilizada para apresentar os resultados na próxima seção:

- Um pré-processamento foi feito para que não fosse necessário que a função *GeraCoberturas* fosse chamada a todo momento em Algoritmo 3. Isso reduz a complexidade de tempo de encontrar a solução ótima para um conjunto J fixado de $O(n^{3k})$ para $O(n^{2k})$.

- Coberturas equivalentes (que cobrem o mesmo conjunto de pontos) são removidas. Isto é, se uma cobertura cobre X e outra cobre Y, com $Y \subset X$, apenas a cobertura do conjunto X é considerada. Como $w_i \geq 0$, o lucro da solução que cobre X é, com certeza, maior ou igual ao da solução que cobre Y.

[2] $-\infty$ é um valor, tal que, para qualquer x, $\max\{x, -\infty\} = x$

Algoritmo 2: Algoritmo que resolve o problema em que $k = 1$

Input: Conjunto de pontos de demanda $P = \{p_1, ..., p_n\}$, conjunto de m elipses $E = \{(a_1, b_1), ..., (a_m, b_m)\}$, conjunto $W = \{w_1, ..., w_n\}$, em que w_i é o ganho do i-ésimo ponto, conjunto $\tilde{W} = \{\tilde{w}_1, ..., \tilde{w}_m\}$, em que $\tilde{w}_i$ é o custo da i-ésima elipse.

Output: Lucro Z da solução ótima

$Z \leftarrow -\infty$ `// Armazena a melhor resposta até o momento`
foreach $(a,b) \in E$ **do**
  $A_l = GeraCoberturas(P, a, b)$
  foreach $X \in A_l$ **do**
    $Z \leftarrow \max\{Z, \sum_{i \in X} w_i - \tilde{w}_l\}$
  end
end
return Z

Algoritmo 3: Algoritmo que resolve o problema no caso geral

Input: Conjunto de pontos de demanda $P = \{p_1, ..., p_n\}$, conjunto de m elipses $E = \{(a_1, b_1), ..., (a_m, b_m)\}$, conjunto $W = \{w_1, ..., w_n\}$, em que w_i é o ganho do i-ésimo ponto, conjunto $\tilde{W} = \{\tilde{w}_1, ..., \tilde{w}_m\}$, em que $\tilde{w}_i$ é o custo da i-ésima elipse.

Output: Lucro Z da solução ótima

Function *solve(P, E, W, $\tilde{W}$, J, X, l)* **is** `// Resolve o problema recursivamente`
  if $l = m + 1$ **then** `// Caso base`
    if $|J| = k$ **then**
      return $\sum_{i \in X} w_i - \sum_{i \in J} \tilde{w}_i$
    else
      return $-\infty$
    end
  end
  $Z \leftarrow solve(P, E, W, \tilde{W}, J, X, l + 1)$ `// l-ésima elipse não é selecionada`
  if $|J| < k$ **then** `// elipse é utilizada`
    $A_l \leftarrow GeraCoberturas(P \setminus X, a_l, b_l)$ `// todos as coberturas para o conjunto de pontos atual`
    foreach $X' \in A_l$ **do**
      $Z \leftarrow \max\{Z, solve(P, E, W, \tilde{W}, J \cup \{l\}, X \cup X', l + 1)\}$
    end
  end
  return Z
end

3.4 Resultados

A linguagem de programação escolhida para a implementação do método proposto foi C++ (g++ 6.2.0). O sistema em que o programa foi executado possui as seguintes características:

- Processador Intel Core i5 2.2Ghz

- 4Gb de Memória RAM
- Sistema operacional Linux 64bit (Lubuntu 14.04)

Foi adotada a *flag* de compilação -O3 que permite que o compilador otimize o código. Isso, de certo modo, faz com que a performance do método dependa menos em detalhes de implementação.

As instâncias utilizadas são as mesmas apresentadas em (ANDRETTA; BIRGIN, 2013) e foram obtidas em <https://www.ime.usp.br/~egbirgin/sources/ab-ellipses/>. Os nomes dados a elas serão mantidos, nas tabelas que apresentam os resultados também será feita referência ao conjunto de pontos que cada instância utiliza.

Como o método proposto é determinístico e sempre encontra a solução ótima, apenas a performance será comparada com (ANDRETTA; BIRGIN, 2013). Para facilitar a apresentação dos resultados nas tabelas, o tempo de execução obtido pelo método em (ANDRETTA; BIRGIN, 2013) será chamado de "Tempo de CPU em segs. (b)."

Como os experimentos foram executados em diferentes ambientes, nada será concluído para performances consideradas próximas. Por exemplo, observando os resultados em Tabela 1 é possível afirmar que o método proposto neste trabalho é superior em termos de tempo de execução para as instâncias CM7-CM9. No entanto, para as instâncias CM1-CM3, tem-se que a diferença de performance apresentada pode ser considerada inconclusiva.

Na Tabela 2 são apresentados alguns resultados em que a diferença entre o tempo de execução dos dois métodos foi significativa. É possível dizer que o método proposto por este trabalho se torna ineficiente a medida que k e m crescem. O método em (ANDRETTA; BIRGIN, 2013) é, nessa questão, mais estável.

Tabela 1 – Resultados obtidos para as instâncias CM1-CM9

Problema					Solução		Performance	
Nome	Pontos	n	m	k	Elipses utilizadas	Lucro	Tempo de CPU em segs.	Tempo de CPU em segs. (b)
CM1	P_1	25	3	1	3	2.0	0.00	0.00
CM2	P_1	25	3	2	1,2	3.8	0.00	0.11
CM3	P_1	25	3	3	1,2,3	3.0	0.01	0.11
CM4	P_2	50	3	1	3	4.2	0.00	0.10
CM5	P_2	50	3	2	1,3	8.2	0.00	4.59
CM6	P_2	50	3	3	1,2,3	10.0	0.07	4.60
CM7	P_3	100	3	1	3	12.0	0.05	71.61
CM8	P_3	100	3	2	2,3	20.0	0.10	2,772.33
CM9	P_3	100	3	3	1,2,3	27.0	5.1	2,786.42

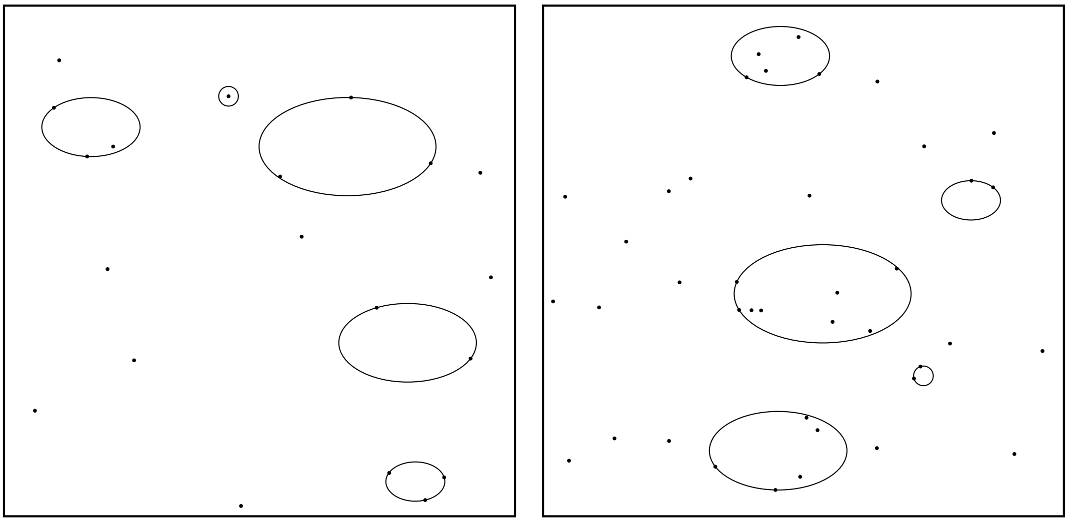
Fonte: Dados da pesquisa.

Tabela 2 – Alguns resultados de AB001-AB120 em que uma diferença significativa de performance entre os dois métodos foi observada

Problema					Solução		Performance	
Nome	Pontos	n	m	k	Elipses utilizadas	Lucro	Tempo de CPU em segs.	Tempo de CPU em segs. (b)
AB092	Q_8	80	5	1	2	6.2	0.01	0.59
AB093	Q_8	80	5	2	2,3	10.7	0.03	17.20
AB096	Q_8	80	5	5	1,2,3,4,5	19.5	1247.42	17.42
AB104	Q_9	90	5	1	2	8.2	0.01	1.80
AB105	Q_9	90	5	2	2,3	12.7	0.04	56.37
AB108	Q_9	90	5	5	1,2,3,4,5	21.5	2422.92	58.1
AB116	Q_{10}	100	5	1	1	8.5	0.02	13.16
AB117	Q_{10}	100	5	2	1,3	16	0.07	293.94
AB118	Q_{10}	100	5	3	1,2,3	22.2	5.70	298.43
AB120	Q_{10}	100	5	5	1,2,3,4,5	27.5	6678.83	293.73

Fonte: Dados da pesquisa.

Figura 3 – Solução ótima encontrada para AB024 e AB048



(a) AB024 (b) AB048

Fonte: Elaborada pelo autor.

No Apêndice B os resultados para todas as instâncias são exibidos. A Figura 3 ilustra a solução ótima encontrada para as instâncias AB024 e AB048. Em ambos os casos o ganho de cada ponto de demanda é unitário e $m = k = 5$, ou seja, deseja-se cobrir o maior número de pontos utilizando todas as elipses.

3.5 Dificuldades e Limitações

No Capítulo 2 foi descrita uma proposição (2.2.1) que serve como base para o algoritmo desenvolvido neste capítulo. Nela é apresentado um processo capaz de, a partir de uma cobertura de um conjunto de pontos, obter uma outra cobertura, tal que dois pontos desse conjunto estejam na borda da elipse utilizada. Quando utilizada no algoritmo, é fácil observar que, neste processo, novos pontos podem acabar entrando na cobertura da elipse. Isto não é um problema se os seus ganhos forem não-negativos. Por outro lado, se esta condição não é imposta, a solução ótima pode não ser encontrada. Esse caso acontece, por exemplo, se toda cobertura equivalente da solução ótima, com dois pontos na borda da elipse, acaba cobrindo um ponto com custo negativo que não pertence à solução ótima.

Capítulo 4

CONCLUSÃO

O algoritmo proposto neste trabalho mostrou-se eficiente para algumas instâncias, no entanto para $k \approx m$ foram obtidos tempos de execução muito piores que os apresentados em (ANDRETTA; BIRGIN, 2013). Isso se deve principalmente ao *backtracking* feito em Algoritmo 3 que verifica todas as possibilidades de posicionamento para cada elipse. Quando $m = k$, a complexidade de tempo para realizar essa tarefa é de $O(n^{2m}) + O(mn^3) = O(n^{2m})$, em que $O(mn^3)$ é a complexidade de pré-processamento da função *GeraCoberturas*. Como trabalhos futuros, algumas melhorias podem ser feitas:

- A função *GeraCoberturas* pode ser implementada em $O(n^2 \lg n)$ fixando cada ponto e ordenando os outros pelo ângulo polar com relação ao fixado e mantendo o conjunto de pontos cobertos em uma estrutura de dados que permita inserção e remoção de elementos em $O(\lg n)$. Veja que esta mudança alteraria apenas a complexidade do pré-processamento de $O(mn^3)$ para $O(mn^2 \lg n)$.

- Todo par de pontos (p_i, p_j) que determina uma cobertura satisfaz: $|p_i^x - p_j^x| \leq 2a$ e $|p_i^x - p_j^x| \leq 2b$. Portanto, para cada ponto p_i, apenas os pontos que pertencem ao retângulo $[p_i^x - a, p_i^x + a] \times [p_i^y - b, p_i^y + b]$ precisam ser avaliados. Esta mudança também só afeta a fase de pré-processamento da função *GeraCoberturas*.

- Se um conjunto X de pontos é coberto, nenhum dos pares (i, j), tal que, $i, j \in X$ precisam ser avaliados como pares de pontos que determinam coberturas de uma possível solução ótima. Por isso, suspeita-se que seja necessário avaliar apenas $O(n)$ conjunto de pontos para se encontrar a solução ótima.

- O *backtracking* feito no Algoritmo 3 pode ser modelado como um problema de programação inteira. Um estudo podo ser feito para comparar se há melhoria empregando esse tipo de modelagem.

A versão do problema em que as elipses podem ser livremente rotacionadas também pode ser estudada a partir deste trabalho. Uma investigação pode ser feita para dizer se há um número finito de centros em que uma elipse pode ser posicionada de tal maneira a conter três pontos de demanda em sua borda.

4.1 Análise do Curso

Primeiramente, vou apresentar algumas críticas construtivas ao curso e tentar propor soluções que, provavelmente, não são ideias, mas podem ser um começo para a resolução de questões que talvez estejam impedindo de termos um curso melhor.

- Falta de um padrão em disciplinas ministradas por diferentes docentes:

  Várias disciplinas que cursei durante a graduação tiveram mais de uma turma em que cada uma era dada por um professor diferente. A maneira diferente de se abordar a ementa, algumas vezes, foi impressionante, mas em meu ponto de vista, o grande problema é a diferença do nível exigido por cada professor.

  Eu acho que este é o pior problema dos cursos do ICMC, talvez até da USP. A dificuldade e quantidade de trabalho que você terá para conseguir passar em uma matéria está diretamente relacionada com qual docente será responsável por ela. Durante a graduação presenciei diversas vezes a mesma disciplina, no mesmo semestre, tendo uma turma com quase cem por cento de aprovação e a outra com menos de vinte por cento. Acho que pequenas medidas seriam suficientes para resolver ou diminuir este problema. Minhas sugestões são: fazer com que um professor elabore e corrija a prova da turma que não é dele, colocar um terceiro professor para ser responsável apenas pela avaliação.

- Falta de flexibilidade:

  Acho que a nossa grade curricular passou por algumas mudanças positivas como a retirada das disciplinas obrigatórias: Laboratório de Bases de Dados, Prática em Organização de Computadores e Sistemas Operacionais 2. No entanto, ainda temos uma grade muito pouco flexível se comparada com universidades de referência fora do país. Gostaria muito mais de ter feito um curso em que, nos dois primeiros anos cursaria disciplinas básicas de matemática e computação e no restante teria um número menor de disciplinas obrigatórias para cursar por semestre, o que me permitiria escolher matérias de minha preferência. Acho que a grade atual nos força a aprender muita coisa superficialmente que um cientista da computação não precisa saber. Seria melhor pra todos que os alunos fazendo essas disciplinas estivessem lá por escolha. Eu, por exemplo, gosto mais da parte teórica da computação e matemática. Achei muito ruim ter que cursar, no sétimo período, a disciplina chamada Sistemas de Informação. Não aprendi praticamente nada, fiz apenas o suficiente para passar e senti que perdi meu tempo com algo que não tenho interesse. Claro que haviam alunos que se interessavam pelo assunto e participavam ativamente das aulas e atividades, para eles e para o docente seria muito mais proveitoso se alunos desinteressados como eu, não tivessem sido obrigados a estarem lá.

- Álgebra Linear e Equações Diferenciais

> “Na verdade não sei muito bem o porquê de ser ensinado EDO junto com álgebra linear”. Não posso citar propriamente, mas quem disse isso foi a minha professora na primeira aula de Álgebra Linear e Equações Diferenciais. Realmente, terminei a graduação, usei álgebra linear, usei equações diferencias em algum momento, porém ainda não entendi a ligação entre as duas que justifique serem dadas na mesma matéria. Posso dizer, porém, é que um curso inteiro dedicado à álgebra linear acabou me fazendo falta, até mesmo em disciplinas da graduação como calculo numérico e processos estocásticos. Por isso, acho que Álgebra Linear e Equações Diferenciais deveriam ser separadas e, talvez, equações diferencias deveria se tornar uma disciplina optativa.

Apesar das críticas, ainda acho que o ICMC é um dos melhores, se não o melhor, lugar para se cursar Ciências de Computação no Brasil. Ótima estrutura, professores que gostam de ensinar e, na maioria dos casos, são especialistas naquilo que estão ensinando e a cidade que nos permite morar perto do campus e viver uma experiência universitária bem legal. Para finalizar vou citar alguns professores que me marcaram durante a graduação por aulas muito boas e desafiadoras que contribuíram para minha evolução durante esse período:

Eugênio Massa. Foi meu professor de cálculo I, foi em suas aulas que comecei a perceber que a matemática na graduação era um negócio bem mais sério do que eu estava acostumado. Eu acho que a aula dele foi a mais desafiadora que encontrei durante os meus anos aqui, estudei muito, no começo por medo e depois por aceitar o desafio de entender as coisas que ele passava.

Sandra Maria Aluisio. Foi minha professora de Teoria da Computação e Linguagens Formais, minha matéria de favorita de computação durante a graduação. Sandra é muito dedicada e claramente adora ensinar essa disciplina. Gostei muito do assunto, acho que a Sandra tem um pouco de culpa nisso.

Francisco A. Rodrigues. Foi meu professor de Processos Estocásticos, matéria que despertou grande interesse em mim por teoria de probabilidade (acabei até cursando Probabilidade II com a estatística e agora estou cursando matérias da pós-graduação de probabilidade). O Francisco foi, sem dúvidas, o professor mais carismático e didático que tive na graduação, talvez ele nem perceba isso, me parece que é uma habilidade natural dele, a de ensinar coisas complicadas de uma maneira simples e interessante.

João E. S. Batista Neto. Durante a graduação participei do grupo de estudos para maratona de programação, quando estava no segundo ano, o João assumiu o posto de docente responsável pelo grupo. Para nossa surpresa, o professor logo de cara começou a nos ajudar e realmente a participar. Ele ia nas reuniões, viagens, se enturmou com todo mundo e fazia propaganda do grupo nas aulas. Isso não mudou com o tempo, desde aquela época, ele continua nos ajudando e já conseguimos classificar para dois mundiais com ele presente, comemorando junto com todo mundo.

REFERÊNCIAS

ANDRETTA, M.; BIRGIN, E. G. Deterministic and stochastic global optimization techniques for planar covering with elipses problems. **European Journal of Operational Research**, v. 22, p. 23–40, 2013. Citado 4 vezes nas páginas 25, 26, 30 e 33.

BOULOS, P.; CAMARGO, I. de. **Cônicas**. São Paulo: Pearson Education do Brasil, 2006. 285-350 p. Citado na página 21.

CANBOLAT, M. S.; MASSOW, M. von. Planar maximal covering with ellipses. **Computers and Industrial Engineering**, v. 57, p. 201–208, 2009. Citado 4 vezes nas páginas 19, 21, 25 e 26.

CHURCH, R. The planar maximal covering location problem. **Journal of Regional Science**, v. 24, p. 185–201, 1984. Citado na página 19.

CHURCH, R.; REVELLE, C. The maximal covering location problem. **Papers of the Regional Science Association**, v. 6, p. 101–118, 1974. Citado na página 19.

MEHREZ, A.; STULMAN, A. The maximal covering location problem with facility placement on the entire plane. **Journal of Regional Science**, Blackwell Publishing Ltd, v. 22, n. 3, p. 361–365, 1982. ISSN 1467-9787. Disponível em: <http://dx.doi.org/10.1111/j.1467-9787.1982.tb00759.x>. Citado na página 19.

YOUNIES, H.; WESOLOWSKY, G. O. A mixed integer formulation for maximal covering by inclined parallelograms. **European Journal of Operational Research**, v. 159, n. 1, p. 83 – 94, 2004. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221703003898>. Citado na página 19.

APÊNDICE A

DETERMINANDO $\Phi(a,b,p,q)$

Nesta seção será apresentada uma forma de se determinar $\Phi(a,b,p,q)$, que denota o conjunto de centros onde $E_{a,b}$ (elipse de parâmetros a e b) pode ser posicionada, de tal maneira a conter os pontos p e $q \in \mathbb{R}^2$.

A.1 Intersecção elipse-reta

A intersecção de uma elipse na origem com uma reta será utilizada para facilitar os cálculos mais adiante. Ela pode ser obtida substituindo a equação $y = mx + c$, na equação $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

$$\begin{aligned} x_{1,2} &= \frac{-a^2mc \pm ab\sqrt{a^2m^2+b^2-c^2}}{a^2m^2+b^2} \\ y_{1,2} &= \frac{-b^2c \pm abm\sqrt{a^2m^2+b^2-c^2}}{a^2m^2+b^2} \end{aligned} \tag{A.1}$$

A.1.1 Distância entre pontos de uma intersecção elipse-reta

A distância ao quadrado entre dois pontos de intersecção entre uma reta e elipse é, utilizando (A.1):

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = \frac{(a^2m^2+b^2-c^2)(4a^2b^2+4a^2b^2m^2)}{(a^2m^2+b^2)^2} \tag{A.2}$$

A.2 Caso $p_x = q_x$

Suponha que o centro da elipse seja $c = (c_x, c_y)$, tem-se:

$$\frac{(p_x-c_x)^2}{a^2} + \frac{(p_y-c_y)^2}{b^2} = \frac{(q_x-c_x)^2}{a^2} + \frac{(q_y-c_y)^2}{b^2} = \frac{(p_x-c_x)^2}{a^2} + \frac{(q_y-c_y)^2}{b^2}$$

$$\implies \frac{(p_y-c_y)^2}{b^2} - \frac{(q_y-c_y)^2}{b^2} = 0 \implies c_y = \frac{p_y^2-q_y^2}{2p_y-2q_y} = \frac{p_y+q_y}{2}$$

Como c_x pode ser escrito como:

$$c_x = p_x \pm \frac{a}{b}\sqrt{b^2 - (p_y - c_y)^2},$$

tem-se que:

$$\Phi(a,b,p,q) = \begin{cases} \{(c_x^+, \frac{p_y+q_y}{2}), (c_x^-, \frac{p_y+q_y}{2})\}, & |p_y - q_y| \leq 2b \\ \emptyset & c.c. \end{cases}$$

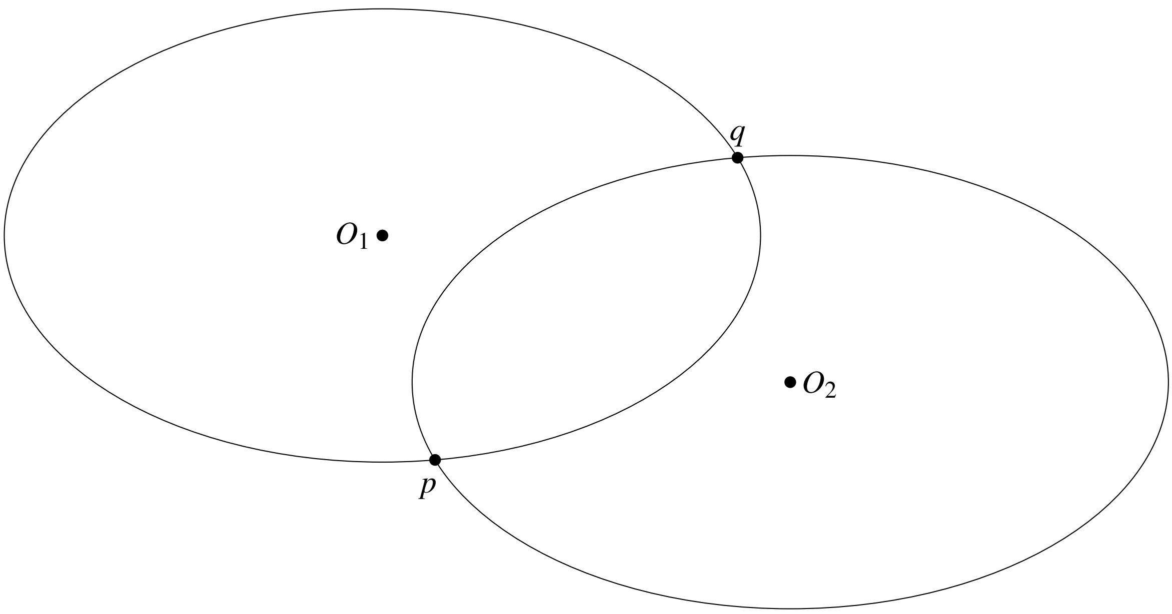
em que c_x^+ e c_x^- são os dois possíveis valores que c_x pode assumir.

A.3 Caso $p_x \neq q_x$

Suponha que o sistema de coordenadas esteja transladado para o centro da elipse e que p' e q' sejam os pontos p e q nesse sistema de coordenadas, isto é $p' = p + \Delta$, $q' = q + \Delta$. O problema equivalente que será resolvido é o de determinar os possíveis pares (p', q') que pertencem à $E_{a,b}$ com centro na origem.

Veja que p' e q' definem uma reta de equação $y = mx + c$, cujos pontos de intersecção com a elipse têm distância igual a $dist(p,q)$. Ou seja, $dist(p,q)^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$, em que $(x_1, y_1), (x_2, y_2)$ são os pontos de intersecção da reta, que passa por p' e q', com $E_{a,b}$ de centro na origem. Como o parâmetro m desta reta é conhecido ($m = \frac{q_y - p_y}{q_x - p_x}$), determinar todos os possíveis valores que o parâmetro c pode assumir equivale a encontrar todos os pares (p', q').

Figura 4 – Gráfico de uma elipse posicionada nos dois centros onde ela contém os pontos p e q



Fonte: Elaborada pelo autor.

Veja que a Figura 4 ilustra as posições onde uma elipse nelas localizadas contém os pontos p e q. Note que, se O_1 for a origem do sistema de coordenadas, a reta definida pelo segmento pq terá equação $y = \frac{q_y - p_y}{q_x - p_x} x + c_1$; se a origem for O_2 a reta terá equação $y = \frac{q_y - p_y}{q_x - p_x} x + c_2$. Como a equação dessa reta é conhecida no sistema de coordenadas original, obter c_1 e c_2 significa obter a translação feita para que O_1 e O_2 sejam as novas origens respectivamente.

Para facilitar a notação, são definidos $D := dist(p,q)^2$, $A := 4a^2b^2 + 4a^2b^2m^2$ e $B := a^2m^2 + b^2$, substituindo em (A.2):

$$D = \frac{(B - c^2)A}{B^2}$$
$$DB^2 = -c^2A + BA$$
$$c^2 = \frac{BA - DB^2}{A} \iff c = \pm\sqrt{\frac{BA - DB^2}{A}}$$

Substituindo c em (A.1) é possível encontrar p' e q' e consequentemente, encontrar a translação feita no sistema de coordenadas para levar a elipse à origem. Note que, como há no máximo 2 valores possíveis para c, concluí-se que $|\Phi(a,b,p,q)| \leq 2$.

APÊNDICE B

TABELA DE RESULTADOS

Tabela 3 – Resultados obtidos para instancias AB001-AB031

Problema					Solução		Performance	
Nome	Pontos	n	m	k	Elipses utilizadas	Lucro	Tempo de CPU em segs.	Tempo de CPU em segs. (b)
AB001	Q_1	10	3	1	2	1.4	0.00	0.00
AB002	Q_1	10	3	2	2,3	2.3	0.00	0.00
AB003	Q_1	10	3	3	1,2,3	2.8	0.00	0.00
AB004	Q_1	10	4	1	4	0.9	0.00	0.00
AB005	Q_1	10	4	2	2,4	1.4	0.00	0.00
AB006	Q_1	10	4	3	2,3,4	1.8	0.00	0.00
AB007	Q_1	10	4	4	1,2,3,4	1	0.00	0.00
AB008	Q_1	10	5	1	5	0.9	0.00	0.00
AB009	Q_1	10	5	2	3,5	1.4	0.00	0.00
AB010	Q_1	10	5	3	3,4,5	1.8	0.00	0.00
AB011	Q_1	10	5	4	2,3,4,5	1	0.00	0.00
AB012	Q_1	10	5	5	1,2,3,4,5	-1.5	0.00	0.00
AB013	Q_2	20	3	1	2	1.4	0.00	0.00
AB014	Q_2	20	3	2	2,3	2.3	0.00	0.00
AB015	Q_2	20	3	3	1,2,3	2.8	0.00	0.00
AB016	Q_2	20	4	1	2	1.5	0.00	0.00
AB017	Q_2	20	4	2	2,3	2.9	0.00	0.00
AB018	Q_2	20	4	3	2,3,4	3.8	0.00	0.00
AB019	Q_2	20	4	4	1,2,3,4	4	0.00	0.00
AB020	Q_2	20	5	1	4	2.4	0.00	0.00
AB021	Q_2	20	5	2	3,4	3.9	0.00	0.02
AB022	Q_2	20	5	3	3,4,5	4.8	0.00	0.01
AB023	Q_2	20	5	4	2,3,4,5	4	0.02	0.01
AB024	Q_2	20	5	5	1,2,3,4,5	2.5	0.06	0.01
AB025	Q_3	30	3	1	1	2.5	0.00	0.00
AB026	Q_3	30	3	2	1,2	4.9	0.00	0.00
AB027	Q_3	30	3	3	1,2,3	6.8	0.00	0.01
AB028	Q_3	30	4	1	2	2.5	0.00	0.00
AB029	Q_3	30	4	2	2,3	4.9	0.00	0.03
AB030	Q_3	30	4	3	1,2,3	6.1	0.00	0.04
AB031	Q_3	30	4	4	1,2,3,4	7	0.03	0.03

Fonte: Dados da pesquisa.

Tabela 4 – Resultados obtidos para instancias AB032-AB075

Problema					Solução		Performance	
Nome	Pontos	n	m	k	Elipses utilizadas	Lucro	Tempo de CPU em segs.	Tempo de CPU em segs. (b)
AB032	Q_3	30	5	1	3	2.5	0.00	0.00
AB033	Q_3	30	5	2	3,4	4.9	0.00	0.14
AB034	Q_3	30	5	3	2,3,4	7.1	0.01	0.15
AB035	Q_3	30	5	4	2,3,4,5	9	0.18	0.15
AB036	Q_3	30	5	5	1,2,3,4,5	9.5	0.79	0.15
AB037	Q_4	40	3	1	1	2.5	0.00	0.00
AB038	Q_4	40	3	2	1,2	4.9	0.00	0.02
AB039	Q_4	40	3	3	1,2,3	6.8	0.00	0.02
AB040	Q_4	40	4	1	1	5.2	0.00	0.00
AB041	Q_4	40	4	2	1,4	7.1	0.00	0.16
AB042	Q_4	40	4	3	1,2,4	8.6	0.01	0.16
AB043	Q_4	40	4	4	1,2,3,4	10	0.12	0.16
AB044	Q_4	40	5	1	3	3.5	0.00	0.01
AB045	Q_4	40	5	2	1,3	7	0.00	0.17
AB046	Q_4	40	5	3	1,2,3	9.2	0.05	0.16
AB047	Q_4	40	5	4	1,2,3,5	11.1	0.78	0.16
AB048	Q_4	40	5	5	1,2,3,4,5	12.5	4.80	0.16
AB049	Q_5	50	3	1	1	5.5	0.00	0.01
AB050	Q_5	50	3	2	1,2	7.9	0.00	0.18
AB051	Q_5	50	3	3	1,2,3	9.8	0.00	0.17
AB052	Q_5	50	4	1	1	5.2	0.00	0.02
AB053	Q_5	50	4	2	1,2	8.7	0.00	0.80
AB054	Q_5	50	4	3	1,2,3	11.1	0.03	0.80
AB055	Q_5	50	4	4	1,2,3,4	13	0.30	0.80
AB056	Q_5	50	5	1	1	3.5	0.00	0.02
AB057	Q_5	50	5	2	1,4	6.9	0.00	0.89
AB058	Q_5	50	5	3	1,3,4	9.4	0.14	0.90
AB059	Q_5	50	5	4	1,2,3,4	11.6	3.10	0.91
AB060	Q_5	50	5	5	1,2,3,4,5	13.5	25.50	0.93
AB061	Q_6	60	3	1	1	3.5	0.00	0.00
AB062	Q_6	60	3	2	1,2	5.9	0.00	0.07
AB063	Q_6	60	3	3	1,2,3	7.8	0.01	0.06
AB064	Q_6	60	4	1	1	5.2	0.00	0.05
AB065	Q_6	60	4	2	1,2	8.7	0.00	0.87
AB066	Q_6	60	4	3	1,2,3	12.1	0.09	0.88
AB067	Q_6	60	4	4	1,2,3,4	14	1.11	0.87
AB068	Q_6	60	5	1	3	4.5	0.00	0.16
AB069	Q_6	60	5	2	1,3	9	0.00	3.90
AB070	Q_6	60	5	3	1,3,4	12.4	0.44	3.88
AB071	Q_6	60	5	4	1,2,3,4	14.6	12.41	3.90
AB072	Q_6	60	5	5	1,2,3,4,5	16.5	135.76	4.12
AB073	Q_7	70	3	1	1	4.5	0.00	0.02
AB074	Q_7	70	3	2	1,2	7.9	0.00	0.20
AB075	Q_7	70	3	3	1,2,3	9.8	0.03	0.21

Fonte: Dados da pesquisa.

Tabela 5 – Resultados obtidos para instancias AB076-AB120

Problema					Solução		Performance	
Nome	Pontos	n	m	k	Elipses utilizadas	Lucro	Tempo de CPU em segs.	Tempo de CPU em segs. (b)
AB076	Q_7	70	4	1	1	5.2	0.00	0.03
AB077	Q_7	70	4	2	1,2	9.7	0.00	0.80
AB078	Q_7	70	4	3	1,2,3	13.1	0.20	0.73
AB079	Q_7	70	4	4	1,2,3,4	16	2.96	0.79
AB080	Q_7	70	5	1	1	5.5	0.00	0.30
AB081	Q_7	70	5	2	1,3	10	0.01	6.75
AB082	Q_7	70	5	3	1,2,3	14.2	0.79	6.76
AB083	Q_7	70	5	4	1,2,3,4	17.6	26.33	7.12
AB084	Q_7	70	5	5	1,2,3,4,5	19.5	332.29	7.07
AB085	Q_8	80	3	1	1	4.5	0.00	0.02
AB086	Q_8	80	3	2	1,2	7.9	0.00	0.14
AB087	Q_8	80	3	3	1,2,3	10.8	0.04	0.17
AB088	Q_8	80	4	1	1	7.2	0.00	0.20
AB089	Q_8	80	4	2	1,2	12.7	0.01	5.11
AB090	Q_8	80	4	3	1,2,3	16.1	0.45	5.25
AB091	Q_8	80	4	4	1,2,3,4	18	7.87	5.10
AB092	Q_8	80	5	1	2	6.2	0.01	0.59
AB093	Q_8	80	5	2	2,3	10.7	0.03	17.20
AB094	Q_8	80	5	3	1,2,3	15.2	1.90	17.49
AB095	Q_8	80	5	4	1,2,3,4	18.6	78.98	18.64
AB096	Q_8	80	5	5	1,2,3,4,5	19.5	1247.42	17.42
AB097	Q_9	90	3	1	1	5.5	0.00	0.03
AB098	Q_9	90	3	2	1,2	9.9	0.00	0.61
AB099	Q_9	90	3	3	1,2,3	11.8	0.10	0.50
AB100	Q_9	90	4	1	1	6.2	0.00	0.24
AB101	Q_9	90	4	2	1,2	10.7	0.01	4.30
AB102	Q_9	90	4	3	1,2,3	14.1	0.67	4.26
AB103	Q_9	90	4	4	1,2,3,4	17	14.62	4.32
AB104	Q_9	90	5	1	2	8.2	0.01	1.80
AB105	Q_9	90	5	2	2,3	12.7	0.04	56.37
AB106	Q_9	90	5	3	1,2,3	16.2	2.98	56.71
AB107	Q_9	90	5	4	1,2,3,4	19.6	137.92	58.52
AB108	Q_9	90	5	5	1,2,3,4,5	21.5	2422.92	58.13
AB109	Q_{10}	100	3	1	1	5.5	0.00	0.08
AB110	Q_{10}	100	3	2	1,2	10.9	0.00	1.05
AB111	Q_{10}	100	3	3	1,2,3	13.8	0.18	0.96
AB112	Q_{10}	100	4	1	1	7.2	0.01	0.76
AB113	Q_{10}	100	4	2	1,2	12.7	0.02	17.87
AB114	Q_{10}	100	4	3	1,2,3	17.1	1.36	17.84
AB115	Q_{10}	100	4	4	1,2,3,4	20	32.49	17.58
AB116	Q_{10}	100	5	1	1	8.5	0.02	13.16
AB117	Q_{10}	100	5	2	1,3	16	0.07	293.94
AB118	Q_{10}	100	5	3	1,2,3	22.2	5.70	298.43
AB119	Q_{10}	100	5	4	1,2,3,4	25.6	318.56	291.37
AB120	Q_{10}	100	5	5	1,2,3,4,5	27.5	6678.83	293.73

Fonte: Dados da pesquisa.