

O problema do carteiro chinês

Carlos Eduardo Ferreira
Gabriel Fernandes de Oliveira

1 As sete pontes de Königsberg

O problema das sete pontes de Königsberg foi descrito e solucionado pelo matemático Leonhard Euler em 1736, o problema consistia em decidir se seria possível traçar no mapa de Königsberg um trajeto que percorresse cada uma de suas 7 pontes uma única vez, sem repetições.

Euler resolveu esse problema do seguinte modo: Primeiramente, ele identificou cada uma das massas de terra do mapa com as letras A, B, C e D.

Em seguida, ele definiu que um trajeto nesse mapa seria descrito por uma sequência dessas letras: por exemplo, "ACD" indicaria o trajeto que se inicia na massa de terra A, move-se para a massa de terra C, usando uma das pontes, e termina na massa D.

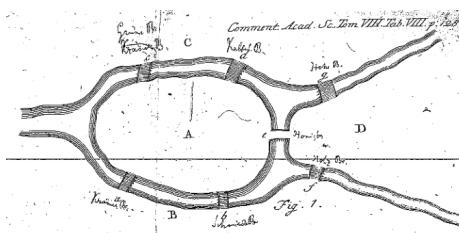


Figura 2: Representação de Euler

entanto possuem 3 pontes, portanto, suas letras correspondentes deverão aparecer apenas 2 vezes na sequência.

Chegamos assim a um absurdo, pois inicialmente provamos que a sequência

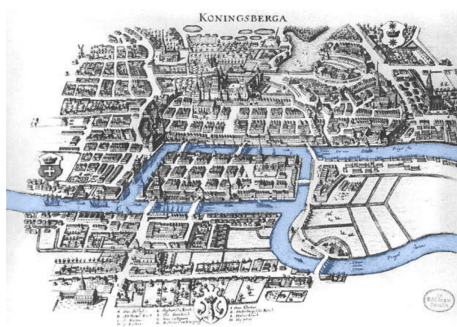


Figura 1: Representação das sete pontes de Königsberg

Euler então começou a definir algumas restrições, assumindo que o problema possuiria alguma solução:

Como o trajeto final deverá passar por todas as 7 pontes exatamente uma vez, isso implica que a sequência de letras que o representa deverá ter tamanho 8.

Além disso, como a massa de terra A possui 5 pontes, necessariamente a letra A aparecerá exatamente 3 vezes na sequência. A massa B, C e D, no

de letras que representa uma solução deveria ter tamanho 8 e depois chegamos que a mesma deveria ter tamanho 9. Provando assim, por absurdo, que não existe um trajeto como o pedido no enunciado do problema.

Essa observação que, aos olhos de hoje, parece muito simples, teve um impacto profundo nas áreas da Matemática e Computação. A modelagem de Euler, tratando as massas de terra e pontes de forma abstrata foi absolutamente inovadora e é o embrião da área da teoria dos grafos.

Vamos definir como **passeio** em um grafo uma sequência finita não vazia $P = \{v_0, v_1, \dots, v_k\}$, cujos termos são vértices v_i tal que, para todo i , $0 \leq i < k$, os vértices v_i e v_{i+1} são ligados por uma aresta. Os vértices v_0 e v_k são a origem e o término de P , respectivamente; e os vértices v_1, v_2, \dots, v_{k-1} são chamados vértices internos de P .

Uma **trilha** é um passeio sem arestas repetidas. Um **caminho** é um passeio sem vértices repetidos. Definimos o comprimento de um passeio, denotado por $\|P\|$, como o número de arestas de P .

Um passeio é considerado **fechado** se sua origem e término são iguais.

Uma trilha fechada é um **circuito**.

Devido às contribuições de Euler ao problema descrito, chama-se **trilha euleriana** como uma trilha que passa por todas arestas de um grafo e **grafo euleriano** como um grafo que possui um circuito euleriana fechada.

Euler modelou essa área da cidade como um grafo, representado na figura 3, tratando as pontes como arestas e as massas de terra como vértices.

A partir de tal modelagem e das definições feitas, o problema de Königsberg consiste, em definir se o grafo que representa a cidade possui ou não uma trilha euleriana.

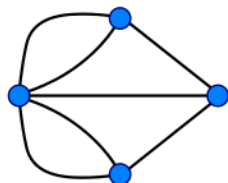


Figura 3:
Modelagem
de Königsberg
como um grafo

Apesar de ter recebido grande parte do crédito histórico pelas suas implicações, Euler não provou que qualquer grafo conexo com vértices de grau par era euleriano, essa prova só foi publicada mais de cem anos depois, em 1873, por Carl Hierholzer [2], no que se tornou o conhecido Teorema de Euler.

Teorema 1.1 (Teorema de Euler ou de Euler-Hierholzer). *Um grafo é euleriano se, e somente se, é conexo e todos seus vértices possuem grau par.*

Antes de mostrar a prova de tal teorema, primeiramente apresentaremos o seguinte lema:

Seja $\delta(G)$ o grau mínimo de um vértice pertencente a G .

Lema 1. *Se G é um grafo tal que $\delta(G) \geq 2$, então G possui um circuito.*

Demonstração. Vamos assumir que um grafo qualquer $G = \{V, A\}$ não possui um caminho fechado. Seja P um caminho de comprimento maximal pertencente a G , denominamos v um dos extremos de P . Como P é um caminho de comprimento maximal, é impossível, por definição, que v possua uma aresta vu que o ligue a um vértice u não pertencente a P .

Como, da premissa, todos vértices possuem grau maior ou igual a 2, isso implica que v possuirá ao menos duas arestas que o ligam a vértices pertencentes a P .

Porém, como v é um vértice extremo de P , apenas uma dessas arestas pode pertencer ao caminho P . Isso implica que a outra aresta, digamos vw , não pertencente a P , implica na existência de um caminho fechado.

Basta tomar o subcaminho entre os vértices v e w pertencente à P juntamente com a aresta vw que teremos um caminho fechado. Chegando assim em uma contradição.

Demonstra-se assim que G deverá possuir ao menos um caminho fechado dadas as condições do lema. \square

Provado tal lema, podemos agora provar o teorema 1.1:

Demonstração. Seja $G = \{V, E\}$ o grafo em questão.

(\Rightarrow) Começamos provando que se um grafo é euleriano então todos seus vértices possuem grau par.

Seja T um circuito euleriano, cuja existência é garantida já que G é euleriano. Analisaremos o grau de um vértice qualquer v de G .

Se v não for um vértice extremo de T , então sempre que o mesmo aparecer em T ele deverá ser precedido e sucedido de arestas. Indicando assim que v deverá possuir um grau par.

Do contrário, se v for o vértice extremo de T ele necessariamente possui duas arestas, uma ligando-o ao segundo vértice do circuito, e outra o ligando ao penúltimo vértice de T . Além disso, cada aparição de v como vértice interno de T contabiliza mais duas arestas ao vértice em questão, de modo que v possuirá um grau par ao final.

Sendo assim, se G é euleriano, então todos seus vértices (extremos ou não) possuem grau par, como queríamos demonstrar.

(\Leftarrow) Agora, provaremos por indução no número de arestas de G que se o grafo for conexo e se todos seus nós possuem grau par, então ele é euleriano.

O caso base da indução é quando não há arestas em G . O único grafo conexo que respeita tal condição é o grafo que possui apenas um vértice v . Neste exemplo, $\{v\}$ é o circuito euleriano do grafo.

A hipótese de indução é que todo grafo simples, conexo, que possui até $k - 1$ arestas e cujos vértices têm grau par é euleriano. Seja G um grafo conexo, de vértices de grau par e que possua k arestas, provaremos que ele também deverá ser euleriano.

Como G é conexo, vale que $\delta(G) \geq 1$ e como todos nós de G têm grau par, podemos afirmar que $\delta(G) \geq 2$. Sendo assim, pelo lema 1, G deverá possuir um circuito C .

Se C possui todas arestas de G , então C é um circuito euleriano do grafo, finalizando a prova.

Do contrário, aplicamos o seguinte procedimento para construir um circuito euleriano \mathcal{C} de G :

Retira-se de G as arestas pertencentes a C , resultando assim em um grafo G' . Possivelmente G' será desconexo, por isso definimos que G' será a união de k componentes conexas G'_1, G'_2, \dots, G'_k disjuntas entre si.

O grau dos vértices dessas componentes G'_i deverá ser par, já que, ao retirar todas as arestas de um circuito do grafo G , diminuimos o grau de um vértice qualquer v em duas vezes o número de aparições do mesmo no circuito, mantendo assim a paridade dos graus.

Além disso, cada componente conexa de G' possuirá uma quantidade de arestas menor do que k . Portanto, pela hipótese da indução, cada uma dessas componentes deverá possuir um circuito euleriano próprio. Chamaremos de C_i o circuito euleriano da componente G'_i .

Ao longo do algoritmo, os circuitos C_i serão adicionados, um a um, ao circuito resultante \mathcal{C} . Definimos \mathcal{T} , como o conjunto dos circuitos eulerianos C_i que ainda não fazem parte de \mathcal{C} . Inicialmente, \mathcal{T} é igual ao conjunto $\{C_1, C_2, \dots, C_k\}$.

$$C = \{v, v_2, \dots, v_n, v\}$$

Para cada vértice u de C devemos realizar as seguintes verificações:

Começamos adicionando u ao circuito euleriano \mathcal{C} que estamos construindo.

Enquanto houver um circuito euleriano C_i pertencente a \mathcal{T} do qual u faz parte, fazemos o seguinte:

1. Representamos C_i como:

$$C_i = \{u, u_2, \dots, u_l, u\}$$

2. Adicionamos ao final de \mathcal{C} o circuito C_i como representado, exceto pelo primeiro vértice u .
3. Removemos de \mathcal{T} o circuito C_i , indicando que a mesma já foi adicionada a \mathcal{C} .

Repetem-se então as mesmas verificações para os próximos vértices de C .

Ao final desse procedimento, o conjunto \mathcal{T} deverá ser vazio, já que toda trilha C_i possui pelo menos um vértice em C . Além disso, toda trilha C_i deverá ter sido adicionada à \mathcal{C} uma única vez, já que logo após adicionar uma trilha à \mathcal{C} já a removíamos de \mathcal{T} , impedindo que ela fosse adicionada outra vez na trilha euleriana final.

Comprovado o passo da indução, finalizamos a prova do Teorema de Euler por indução.

□

Corolário 1. *Um grafo possui uma trilha euleriana se, e somente se, é conexo e possui apenas zero ou dois vértices de grau ímpar.*

Demonstração. Seja G um grafo conexo qualquer. Realizaremos a demonstração para os seguintes casos:

1. G não possui vértices de grau ímpar. Neste caso, G possui, segundo o teorema 1.1, um circuito euleriano, e portanto, uma trilha euleriana.
2. G possui apenas um vértice de grau ímpar. Este caso é impossível de se acontecer, já que a soma do grau de todos vértices deve ser par, impossibilitando assim que apenas um vértice tenha grau ímpar.
3. G possui dois vértices de grau ímpar.

Sejam u e v os únicos vértices de G que possuem grau ímpar. Adiciona-se uma aresta fictícia ao grafo G , a aresta uv , fazendo com que tanto u quanto v possuam graus pares. Chamaremos o grafo G acrescido da aresta uv de G' . Como u e v eram os únicos vértices de grau ímpar de G , vale que todos vértices de G' possuirão grau par. Além disso, vale que G' é conexo, pois faz parte da premissa que o grafo original G era conexo.

Sendo assim, podemos aplicar o teorema 1.1, provando a existência de um circuito euleriano G' , que chamaremos de C . Por ser euleriano, C deverá percorrer a aresta uv inserida, portanto podemos representar C com u e v lado a lado, do seguinte modo:

$$C = \{u, v, w_1, w_2, \dots, w_k, u\}$$

Tome, agora, T igual ao circuito C sem seu vértice inicial:

$$T = \{v, w_1, w_2, \dots, w_k, u\}$$

Tal procedimento retira do circuito C a aresta artificial uv , transformando-o em uma trilha T , que percorre todas arestas de G' exceto uv , sendo portanto uma trilha euleriana do grafo G .

4. G possui três ou mais vértices de grau ímpar.

Assuma que existe uma trilha euleriana T para G . Neste caso, como pelo menos 3 vértices possuem grau ímpar, necessariamente existirá um vértice v que não é nem o primeiro nem o último vértice de T . Isso implica que todas aparições de v em T são internas ao caminho, ou seja, toda aparição de v será precedida e sucedida de arestas ligadas a v . Como estamos tratando de uma trilha euleriana, sabemos que todas arestas adjacentes a v estão presentes em T uma única vez. Mas como todas arestas de v devem aparecer em pares (precedendo e sucedendo v), isso implica que o grau de v deverá ser par. Contradizendo a premissa.

Por essa contradição provamos que G não possuirá trilha euleriana se tiver três ou mais vértices de grau ímpar.

□

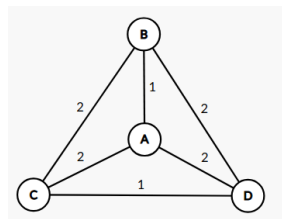
2 O problema do Carteiro Chinês

Com o passar dos anos, a área de teoria dos grafos se desenvolveu muito, tratando dos mais variados tipos de problemas.

Em 1962, mais de 200 anos após Euler descrever sua solução para o problema de Königsberg, o matemático chinês Meigu Guan publicou um estudo que generalizava ainda mais o problema dos grafos eulerianos. Esse problema foi denominado Problema da Inspeção de Rotas, ou, como também é conhecido hoje: Problema do carteiro chinês. A ideia desse problema é encontrar um passeio fechado que visite toda aresta de um grafo conexo pelo menos uma vez. A grande diferença aqui é que as arestas podem ser repetidas, ou seja, usadas mais de uma vez no trajeto final.

O nome do problema está relacionado a um problema que carteiros encontram no planejamento de suas rotas: dada uma cidade com várias ruas de diferentes comprimentos e um posto de carteiros, encontrar a menor rota que um carteiro deve percorrer de modo a poder entregar cartas em todas as ruas da cidade e voltar ao posto de carteiros no fim de sua rota.

Por exemplo, para a figura 4, um passeio fechado de custo ótimo seria:



$$\{A, B, D, C, A, D, C, B, A\}$$

, sendo este um caminho fechado de custo 12. Neste exemplo foi necessário que as arestas AB (ou BA) e DC (ou CD) fossem percorridas duas vezes no passeio, porém nem sempre é necessária esta repetição.

Figura 4: Exemplo de grafo

No caso em que o grafo tratado é euleriano, a resposta para o problema do carteiro chinês é justamente o circuito euleriano do grafo. Nos outros casos, sendo o grafo conexo, o procedimento que seguiremos será similar a realizar a cópia de algumas arestas do grafo de modo a torná-lo euleriano, ou seja, adicionar duplicatas de arestas até que todos os vértices possuam grau par.

Discutiremos nas seções a seguir a solução para o problema em questão com base nas especificidades do grafo do problema:

2.1 Grafos não direcionados

Analisaremos o caso em que o problema é modelado a partir de um grafo $G(V, E)$ simples, conexo e com arestas não direcionadas.

Um circuito Euleriano é um circuito que percorre todas arestas de G exatamente uma vez. Já uma solução qualquer que resolva o problema do carteiro chinês deverá percorrer cada aresta de G pelo menos uma vez, sendo assim uma trilha fechada. Seja $1 + x_e$ o número de vezes que uma aresta $e \in E$ é percorrida em uma solução T do problema do carteiro chinês (PCC).

Se G é euleriano, então a solução para o PCC é o próprio circuito euleriano do grafo.

Do contrário, definimos G' como o grafo formado por G adicionado de x_e cópias de cada aresta $e \in E$. Isto é, cada aresta e de G aparecerá em G' $1 + x_e$ vezes. Deste modo, o grafo G' será euleriano, e portanto a trilha euleriana de G' corresponderá à solução do PCC para o grafo G .

Sendo assim, podemos separar a solução do PCC em duas partes: Encontrar o valor ótimo de x_e para o problema descrito e, em seguida, encontrar um circuito euleriano no grafo G' construído.

Resolveremos a primeira parte do problema com um algoritmo de emparelhamento.

Lema 2. *Para todo grafo simples e conexo, existirá uma solução ótima do PCC em que cada aresta é copiada no máximo 1 vez, ou seja que x_e valerá apenas 0 ou 1.*

Demonstração. Seja T uma solução ótima do PCC para um grafo G e x um vetor indicando a quantidade de cópias necessárias de cada aresta de G para a solução T . Sabemos que x deverá possuir valores inteiros não negativos, além disso, sabemos que o grafo G' , induzido por x , deverá ser euleriano e que o valor de $\sum_e c_e x_e$ será mínimo.

Seja x^* um vetor definido do seguinte modo:

$$x_e^* = x_e \bmod 2$$

Como o grafo G' induzido por x era euleriano, o grafo G^* induzido por x^* também deverá ser euleriano, pois a paridade dos vértices em ambos grafos é a mesma, e também pois tanto G' quanto G^* possuem G , que é conexo, como um subgrafo, e portanto também são conexos.

Pela definição, G^* usa um número menor ou igual de arestas duplicadas que G' , fazendo com que o seguinte se mantenha: $\sum_e c_e x_e^* \leq \sum_e c_e x_e$.

No entanto, como x foi derivado da solução ótima T , sabemos que $\sum_e c_e x_e$ deverá ser mínimo, e portanto, $\sum_e c_e x_e^* = \sum_e c_e x_e$.

Sendo assim, os valores de x^* serão apenas 0 ou 1, por definição, e T^* consistirá em uma solução ótima, assim como T , provando o lema. \square

Lema 3. *Para todo grafo G simples e conexo, haverá uma solução ótima do PCC cujo conjunto de arestas duplicadas consistirá na união de caminhos aresta-disjuntos entre vértices de grau ímpar.*

Demonstração. Inicialmente trataremos o caso em que G não possui nenhum vértice de grau ímpar: Se todos vértices de G tem grau par, então o mesmo deve possuir um circuito euleriano. Por isso, o vetor $x' = \{0, 0, \dots, 0\}$ induz uma solução válida para o PCC, e possui custo zero, sendo assim esta uma solução ótima. Como o conjunto de arestas duplicadas nesta solução é vazio, vale o lema para este grafo.

Provado este caso, seguiremos a demonstração assumindo que G possui ao menos um vértice de grau ímpar.

Seja T uma solução ótima do PCC para um grafo $G(V, E)$ em que cada aresta é copiada no máximo 1 vez, ou seja que x_e valerá 0 ou 1. A existência de tal solução é garantida pelo lema 2. Realizaremos uma prova por indução no número de arestas duplicadas de G , ou seja, em $\sum_e x_e$, para toda aresta $e \in E$.

O caso base dessa indução é quando $\sum_e x_e = 0$. Neste caso, o conjunto de arestas duplicadas será vazio, valendo então a propriedade do lema.

A hipótese de indução será que a propriedade do lema vale para uma soma $\sum_e x_e < k$, com $k > 0$.

Analizaremos agora a situação em que $\sum_e x_e = k$, com $k > 0$.

Seja v um vértice de G de grau ímpar, C um caminho de tamanho maximal que começa no vértice v e percorre apenas arestas duplicadas e seja G' o grafo, possivelmente desconexo, G decrescido das arestas pertencentes a C .

Todos os vértices em G e G' terão a mesma paridade de grau, exceto pelos vértices extremos de C . Como estes eram vértices de grau ímpar em G e apenas uma aresta foi retirada dos mesmos, ambos terão grau par em G' .

Sendo assim, podemos definir um vetor x' que induz uma solução ótima para G' do seguinte modo:

$$x'_e = \begin{cases} 0 & e \in C \\ x_e & e \notin C \end{cases}$$

Vale também o seguinte:

$$\begin{aligned} \sum_e x'_e &= \sum_e x_e - |C| \\ \sum_e x'_e &< \sum_e x_e \\ \sum_e x'_e &< k \end{aligned}$$

Sendo assim, vale a hipótese de indução para o grafo G' , ou seja, existe um conjunto \mathcal{C} de caminhos aresta-disjuntos entre vértices de grau ímpar composto por arestas duplicadas (ou seja, para as quais $x'_e = 1$).

Como o grafo G' não possui as arestas do caminho C , temos que todo caminho de \mathcal{C} deverá ser aresta-disjunto com C . Deste modo, o conjunto de caminhos aresta-disjuntos entre vértices de grau ímpar $\mathcal{C} \cup C$, que induz o grafo T , consistirá de uma solução ótima para o PCC em G . Provando assim o passo da indução.

Finalizando então a prova por indução do lema.

□

Lema 4. *Para todo grafo G simples e conexo, haverá uma solução ótima do PCC cujo conjunto de arestas duplicadas consistirá na união de caminhos de custo mínimo entre vértices de grau ímpar.*

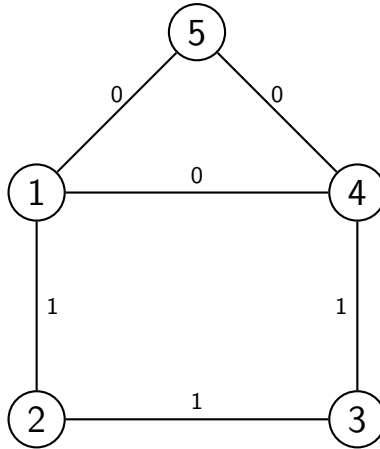


Figura 5: Possível estrutura em grafo com vértices de grau ímpar

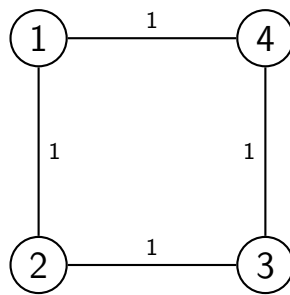


Figura 6: Caso em que o grafo tratado não possui vértices de grau ímpar

Demonstração. Seja S o conjunto de arestas duplicadas em uma solução ótima do PCC para o grafo G , que consiste de uma união de caminhos aresta-disjuntos entre vértices de grau ímpar.

Imagine que um caminho C pertencente a S , que liga os vértices quaisquer u e v , não é o caminho mínimo entre os vértices que liga. Isto é, existe um caminho C' de custo menor que C que liga u e v .

Neste caso, poderíamos retirar de S as arestas do caminho C e adicionar ao conjunto as arestas de C' . Deste modo, o conjunto S possuiria um custo menor que o original, nos permitindo derivar de S uma solução para o PCC de custo menor que a solução original.

Porém, fazia parte da hipótese que a solução original era ótima, nos levando assim a uma contradição.

Provamos assim o lema, garantindo a existência de uma solução para o PCC que consiste de uma união de caminhos mínimos entre vértices de grau ímpar. \square

Um algoritmo que soluciona o problema se baseia em criar um novo grafo $G'(V', E')$. V' é definido como o subconjunto de vértices de V que possuem um grau ímpar em G , já E' é definido como o conjunto de arestas entre todo par de vértices de V' , sendo que uma aresta entre dois vértices u e v quaisquer terá o custo igual ao custo de um caminho de menor custo entre os vértices u e v no grafo original G .

Pelo lema 4, uma solução ótima do PCC em G possui um conjunto de arestas duplicadas que pode ser representado como uma união de caminhos de custo mínimo entre vértices de grau ímpar. Como V' é o conjunto dos vértices de grau ímpar em G e cada aresta de E' representa um caminho de custo mínimo entre dois vértices de V' , podemos reduzir o problema original à um problema de emparelhamento perfeito de custo mínimo no grafo G' .

Dado um emparelhamento perfeito M em G' de custo mínimo, é possível derivar uma solução ótima do PCC. Esta solução será dada por um circuito euleriano no grafo G^* que construímos abaixo:

Considere $G^* = (V, E(G))$. Agora, para cada aresta $e = uv \in M$ (emparelhamento de G'), tome C um caminho de custo mínimo de u a v em G e faça $G^* \leftarrow G^* \cup C$, ou seja, duplique as arestas de C em G^* .

A duplicação das arestas de um caminho modifica apenas a paridade dos vértices extremos. Como, no procedimento acima, estamos duplicando as arestas de todos caminhos derivados do emparelhamento perfeito M , isso implica que modificamos a paridade de todos os vértices de G^* , ou seja, modificamos a paridade dos vértices de grau ímpar de G .

Sendo assim, o grafo G^* resultante é euleriano, e a solução para o PCC do grafo G é o circuito euleriano de G^* .

Analisaremos a complexidade do algoritmo descrito em quatro partes:

Solução:

1. Criar um grafo completo ligando todos vértices u, v que possuírem grau ímpar com arestas de custo igual ao menor caminho entre u e v .

2. Rodar algoritmo de emparelhamento perfeito de custo mínimo (Húngaro com otimização do Edmonds e Karp, $\mathcal{O}(n^3)$).

Observações:

- Pelo “Lema do aperto de mão” é garantido que haverá um número par de vértices de grau ímpar.
- Essa solução não se aproveita do grafo ser esparso, há outra formulação do Edmonds e Johnson que leva isso em consideração.
- Um problema similar é o de cobrir todas arestas com ciclos simples, de modo que o comprimento total dos ciclos é minimizado. Para grafos planares esses problemas são equivalentes.

3 Problemas resolvidos

Esta seção se dedica a discutir aplicações do Problema do Carteiro Chinês em problemas de competições de programação.

3.1 Sereja and the Arrangement of Numbers[3]

Problema

No enunciado do problema define-se uma sequência s como bonita quando existe um inteiro i para cada par de valores distintos u, v pertencentes a s , tal que $s_i = u, s_{i+1} = v$ ou $s_i = v, s_{i+1} = u$.

Em outras palavras, se uma sequência é bonita todo par de valores distintos pertencentes à essa sequência aparecem na mesma lado a lado ao menos uma vez.

Define-se também um sistema de recompensas: Para cada valor há uma recompensa positiva associada. A recompensa total de uma sequência é igual a soma das recompensas de seus valores distintos, ou seja, a recompensa não depende do número de vezes que cada valor aparece na sequência.

O problema consiste então desenvolver um algoritmo que calcule a maior recompensa atingível para uma sequência bonita de tamanho n que seja constituída apenas por valores definidos pela entrada.

A entrada do programa, portanto será dada pelos valores n, m e por m pares q_i, w_i , indicando que q_i é um valor que pode ser utilizado e que possui recompensa w_i , garante-se que todos valores de q_i são distintos e que w_i é positivo.

Um exemplo de entrada para este problema é:

5 4	Neste exemplo $n = 5$ e $m = 4$
1 4	O primeiro valor é 1, e sua recompensa é 4
2 3	Segue o valor 2, com recompensa 3
4 2	Segue o valor 4, com recompensa 2
3 10	Finalmente temos o valor 3, de recompensa 10

Para este exemplo, uma solução ótima é a sequência 1, 2, 3, 1, 1, que é bonita e possui recompensa total igual a 17. Não existe uma sequência bonita que possua os quatro valores do exemplo.

Solução

Como todo valor possui uma recompensa positiva, a solução final deverá usar a maior quantidade de valores distintos que for possível.

Começaremos discutindo um problema mais simples: Qual o tamanho da menor sequência bonita que utiliza x valores distintos?

Para resolver esse problema faremos uma modelagem do problema usando um grafo completo K_x , onde cada vértice representa um valor distinto e cada aresta possui custo unitário.

Todo passeio do grafo K_x pode ser representado como uma sequência dos valores dos vértices do grafo, as sequências definidas como bonitas são aquelas que representam um passeio que percorre todas arestas do grafo K_x .

Para facilitar a análise deste problema, assumiremos que pode existir em um passeio dois vértices de mesmo valor em sequência, mesmo sem haver um loop neste vértice, além disso, vamos assumir que os valores distintos são $1, 2, \dots, x$.

Sendo assim, podemos interpretar uma sequência s que possua apenas valores entre 1 e x como um passeio em K_x , os valores da sequência representam os vértices do grafo K_x , e dois valores adjacentes u e v representam que a aresta uv está presente no passeio derivado de s .

Para que a sequência s seja bonita, todo par de valores distintos u, v deverá aparecer ao menos uma vez lado a lado, o que implica que o passeio derivado de s deverá percorrer a aresta uv ao menos uma vez.

Concluimos então que um passeio derivado de uma sequência que seja bonita, deverá percorrer todas arestas do grafo K_x , sendo assim uma solução do problema do carteiro chinês para o mesmo.

Voltamos assim à pergunta inicial, a menor sequência bonita que utiliza x valores distintos, será aquela que representa uma solução ótima do problema do carteiro chinês para o grafo K_x , já que a solução que minimiza o custo do PCC também minimizará o número de arestas percorridas, e portanto, o tamanho da sequência bonita.

Vejamos agora alguns exemplos:

Para $x = 5$, o grafo induzido é o K_5 , como representado na figura 7. Como todos os vértices do K_5 possuem grau par, a solução ótima do Problema do

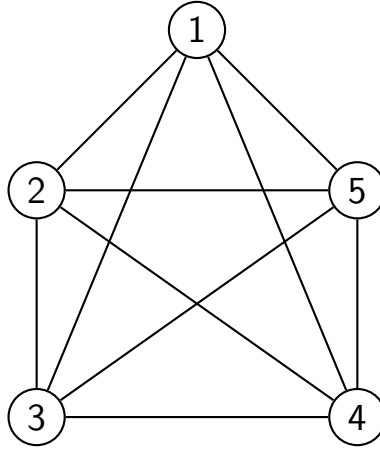


Figura 7: Exemplo com $x = 5$

Carteiro Chinês para este exemplo é também um circuito euleriano, como o seguinte:

$$P = \{1, 2, 3, 4, 5, 1, 3, 5, 2, 4, 1\}$$

Como discutido anteriormente, P , além de solução do PCC é também uma sequência bonita que utiliza os x valores definidos pelos vértices do grafo.

Sendo assim, para utilizar 5 valores distintos, é necessário ter uma sequência de tamanho no mínimo 11, valor este que é igual a $1 + |E(K_5)| = 1 + 10 = 11$.

Analisaremos agora outro exemplo, em que $x = 4$, do qual se deriva o grafo K_4 representado na figura 8.

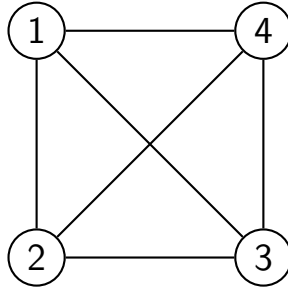


Figura 8: Exemplo com $x = 4$

Neste exemplo, todos os vértices de K_4 possuem grau ímpar, por isso, a primeira parte da solução do problema do carteiro chinês se baseia em duplicar algumas arestas do grafo, tornando-o um grafo euleriano.

Como estamos tratando de um grafo completo, com arestas de custo unitário, não é necessário criar uma condensação do grafo K_x , e achar um emparelha-

mento perfeito de custo mínimo. Basta apenas criar uma duplicação de aresta entre todo par de vértices de índices i e $i + 1$ para todo índice i ímpar, como representado na figura 9 para o exemplo em questão.

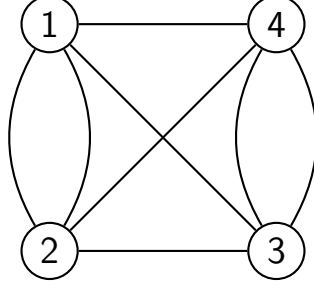


Figura 9: K_4 com arestas duplicadas

A partir do grafo euleriano criado, é possível encontrar o caminho euleriano P , que também é solução do PCC para o grafo K_4 , e, finalmente, sequência bonita de tamanho mínimo que usa 4 valores distintos.

$$P = \{1, 2, 3, 4, 1, 3, 4, 2, 1\}$$

Sendo assim, o tamanho mínimo de uma sequência bonita que possui 4 valores distintos é 9, valor este composto da seguinte forma: $1 + |E(K_4)| = 1 + 6 = 7$ somado ao número de arestas duplicadas $\frac{4}{2} = 2$.

Podemos assim abstrair uma fórmula geral para o tamanho mínimo de uma sequência bonita P que utilize k valores distintos em sua composição:

$$|P_k| = 1 + \frac{k(k-1)}{2} + \begin{cases} 0 & k \text{ ímpar} \\ \frac{k}{2} & k \text{ par} \end{cases}$$

Todo grafo completo com uma quantidade ímpar de vértices é euleriano, já que todos seus vértices possuem grau par, enquanto os grafos completos com quantidade par de vértices não são eulerianos. Por isso a diferença na fórmula de $|P_k|$.

A partir dessa fórmula fechada, é possível descobrir com uma busca binária, em complexidade $\mathcal{O}(\lg n)$, qual o maior valor de k tal que $|P_k| \leq n$.

Encontrado tal valor k , o número máximo de valores distintos que a solução pode possuir para ser bonita, basta descobrir quais valores da entrada escolher para maximizar a recompensa.

Se possuírmos na entrada uma quantidade de valores, m , tal que $m \leq k$, então a solução ótima será a soma das recompensas de todos valores, já que é possível encontrar uma sequência bonita que irá conter todos valores disponíveis.

Do contrário, a solução consistirá em escolher os k valores de maior recompensa disponíveis. Para encontrar tal valor, basta manter uma árvore de busca binária balanceada com as k maiores recompensas dos valores disponíveis, o que custa tempo $\mathcal{O}(m \lg k) = \mathcal{O}(m \lg \sqrt{n})$.

Uma solução para este problema pode ser encontrada, em C++, neste link.

3.2 Jogging Trails[4]

Neste problema devemos desenvolver um programa que calcula o custo de uma solução ótima para o problema do carteiro chinês em um grafo não direcionado de até 15 vértices com pesos nas arestas.

Para resolver esse problema podemos usar o algoritmo padrão descrito aqui para resolver o problema do carteiro chinês.

4 Directed CPP

O grafo deverá ser fortemente conexo.

Solução:

1. Criar um grafo P, Q -bipartido completo. P deve conter todos os vértices do grafo original com excesso de grau de entrada, e Q deve conter todos os vértices com excesso de grau de saída. que possuem valores diferentes de grau de entrada e saída. O custo das arestas entre P e Q deverá ser igual ao custo do menor caminho entre os dois vértices que a mesma liga.
2. Modelar uma rede de fluxo: vértices de P recebem um excesso igual a diferença do grau de entrada pelo grau de saída, vértices de Q recebem uma demanda igual a diferença do grau de saída pelo grau de entrada.
3. Rodar um algoritmo de fluxo de custo mínimo.

Observações:

- Modelagem de fluxo com arestas de capacidade unitária, aumentando a velocidade do algoritmo.

5 Mixed CPP

NP-hard, mesmo se G for planar e se todos c_{ij} forem iguais.

6 Windy Postman Problem

NP-hard

Se todo ciclo do grafo tem o mesmo custo em ambos sentidos, transforma a aresta entre i e j de custos c_{ij} e c_{ji} em uma aresta não direcionada com custo $\frac{c_{ij}+c_{ji}}{2}$. Essa transformação reduz o problema de a um CPP não direcionado, que tem solução polinomial, porém para isso é necessário checar se todo ciclo tem o mesmo custo nas duas direções.

7 Hierarchical Postman Problem

NP-hard

Seja $P = \{A_1, A_2, \dots, A_k\}$ uma partição do conjunto de arestas A . Determinar um caminho em G de custo mínimo que sai de um vértice s e atinge um vértice t , respeitando uma hierarquia das arestas. O caminho só poderá possuir uma aresta pertencente a A_j se o caminho passar anteriormente por todas arestas pertencentes a A_i , se $i < j$.

8 Anotações

- Todo mixed CPP pode ser transformado em um WPP.

Referências

- [1] Euler, Leonhard *Solution problematis ad geometriam situs pertinentis*. Comment. Acad. Sci. U. Petrop 8, 128–40, 1736.
- [2] Hierholzer, Carl “Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren”, Mathematische Annalen, 6 (1): 30–32, doi:10.1007/BF01442866, 1873.
- [3] Problema retirado do Codeforces, problema C do Round #215 (Div1) codeforces.com/problemset/problem/367/C
- [4] Problema retirado do UVa, problema 10296
Link