

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.

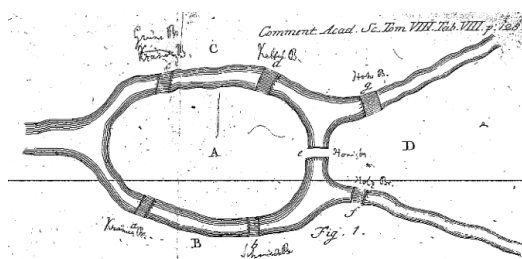


Figura 2: Representação de Euler

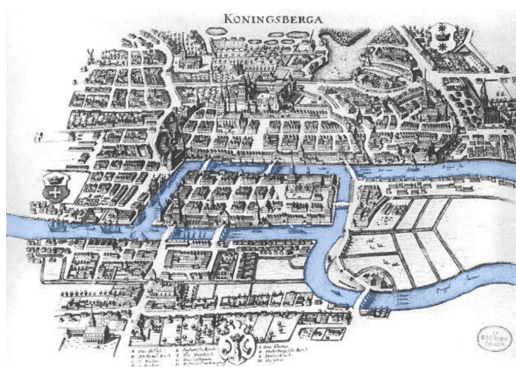


Figura 1: Representação das sete pontes de Königsberg

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, necessári-

amente a letra A aparecerá exatamente 3 vezes na sequência. A massa B, C e D, no 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 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 .

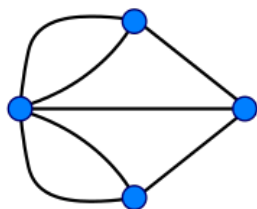
Um passeio é considerado **fechado** se sua origem e término são iguais.

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 8, 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.



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.*

Figura 3: Modelagem de Königsberg como um grafo

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 constuir 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 componetes 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 o mesmo já foi adicionado a \mathcal{C} .

Repetem-se então as mesmas verificações para os próximos vértices de \mathcal{C} .

Ao final desse procedimento, o conjunto \mathcal{T} deverá ser vazio, já que toda trilha C_i possui pelo menos um vértice em \mathcal{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.

□

Até então tratamos apenas de grafos não direcionados, mas podemos expandir esses mesmos conceitos para grafos direcionados, os **digrafos**.

Um grafo é direcionado quando suas arestas são orientadas, chamaremos de **arcos** tais arestas.

Numa analogia com trânsito, uma aresta é como uma estrada de mão dupla, pode-se percorrê-la nos dois sentidos, enquanto isso, um arco é como uma rua de mão única, em que apenas um sentido é permitido.

Quanto tratamos de grafos direcionados (ou digrafos), é necessário refinar a noção de grau de um vértice: Por isso, denominamos **grau de saída** de um vértice v , $\delta^-(v)$, como o número de arcos que saem de v , e **grau de entrada**, $\delta^+(v)$, como o número de arcos que entram em v .

Definimos como **fortemente conexo** um digrafo que possui um caminho entre todo par de vértices, e como **fracamente conexo** um digrafo que possui para todo par de vértices u e v um caminho de u a v ou de v a u .

Dadas as devidas definições, apresentamos agora o teorema de Euler para o caso de digrafos:

Teorema 1.2 (Teorema de Euler para digrafos). *Seja G um digrafo fortemente conexo. G é euleriano se, e somente se, todos seus vértices têm valores de grau de entrada e saída iguais, ou seja, se para todo vértice v de G vale que $\delta^+(v) = \delta^-(v)$.*

Demonstração. (\Rightarrow) Seja G um digrafo euleriano e C o circuito euleriano do mesmo.

Vamos assumir, contrariando o teorema, que G possui um vértice v que tem valores diferentes de grau de entrada e saída.

Assumimos, sem perda de generalidade, que $\delta^-(v) > \delta^+(v)$. Para que cada arco que sai de v seja percorrido uma única vez, é necessário que o circuito C passe por v exatamente δ^+ vezes. Porém, como v tem grau de entrada maior que o grau de saída, C deverá percorrer algum arco que entra em v mais de uma vez. Contradizendo assim a condição inicial de que C é um circuito euleriano.

Por absurdo, provamos que todos vértices de um digrafo euleriano devem possuir o mesmo valor de grau de entrada e saída.

(\Leftarrow) Provaremos a volta por indução no número de arcos de G . A prova a seguir é similar à utilizada na prova do teorema 1.1 caso de grafos não direcionados.

O caso base desta indução é o digrafo G conexo sem arcos. Neste caso G consistirá de um único vértice v , sendo assim $\{v\}$ será um circuito euleriano válido.

A hipótese de indução é que todo digrafo fortemente conexo, que possui vértices com grau de entrada e saída iguais e que possui até $k - 1$ arcos é euleriano.

Seja G , portanto, um digrafo fortemente conexo, em que todos vértices têm igual grau de entrada e saída mas que possui $k > 0$ arcos.

Inicialmente provaremos que G deve possuir um caminho fechado C :

Provando a existência de um caminho fechado em um digrafo G que possui vértices com grau de entrada e saída iguais

Tome um caminho maximal P de G , sendo v o último vértice de tal caminho.

Como o grau de entrada e saída dos vértices de G é igual, e que o vértice v tem ao menos um arco entrando nele, que é percorrido no caminho P , podemos concluir que v possui também um arco saindo dele, vw , não percorrido por P . Além disso, como P é maximal, podemos concluir w já deve pertencer a P .

Com isso prova-se a existência de um caminho fechado no grafo G , formado pelo arco vw e o subcaminho de w a v de P .

Provada a existência de C temos dois casos a analisar:

1. C **percorre todos arcos de G** . Neste caso, C é o circuito euleriano, finalizando a prova.
2. C **não percorre todos arcos de G** . Definimos então G' como o grafo G sem os arcos pertencentes a C .

Como estamos retirando do grafo G um caminho fechado, todo vértice em C perderá apenas um arco de entrada e de saída, mantendo em G' a característica de igualdade do grau de entrada e saída dos vértices.

Porém, não é garantido que G' é fortemente conexo.

Seja G então a união de k componentes fortemente conexas G'_1, G'_2, \dots, G'_k disjuntas entre si.

Sabemos que cada componente G'_i possuirá uma quantidade de arcos menor que k . Portanto, pela hipótese de indução definida, cada componente de G' é euleriana. Chamaremos de C_i o circuito euleriano da componente G'_i .

Mostraremos agora um algoritmo que construirá um circuito euleriano \mathcal{C} de G a partir de C e dos circuitos eulerianos C'_i :

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 $\{C_1, C_2, \dots, C_k\}$.

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:

- (a) Representamos C_i como:

$$C_i = \{u, w, \dots, v, u\}$$

- (b) Adicionamos ao final de \mathcal{C} o circuito C_i como representado, exceto pelo primeiro vértice u .
- (c) Removemos de \mathcal{T} o circuito C_i , indicando que o mesmo já foi adicionado 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 todo circuito C_i possui pelo menos um vértice em C . Além disso, todo circuito C_i deverá ter sido adicionado a \mathcal{C} uma única vez, já que logo após adicionar um circuito a \mathcal{C} já o removíamos de \mathcal{T} , impedindo que ele fosse adicionado outra vez no circuito euleriano final.

Como é possível realizar a construção de um circuito euleriano \mathcal{C} provamos que qualquer digrafo fortemente conexo com vértices possuindo valores iguais de grau de entrada e saída é euleriano.

□

Corolário 2. *Um digrafo G possui uma trilha euleriana se, e somente se, é fracamente conexo, possui no máximo um par de vértices u, v tal que $\delta^-(v) - \delta^+(v) = 1$, $\delta^+(u) - \delta^-(u) = 1$ e todos outros vértices w de $V(G) \setminus \{u, v\}$ possuem $\delta^+(w) = \delta^-(w)$.*

Demonstração. (\Rightarrow) Começaremos provando que G deverá ser fracamente conexo.

Se um digrafo G possui uma trilha euleriana T , podemos derivar um caminho entre qualquer par de vértices u, v a partir de T , como mostra-se a seguir:

Como T é euleriana, ela deve conter os vértices u e v , portanto podemos representar T de um dos seguintes modos:

$$T = \{w_1, \dots, w_i = u, w_{i+1}, \dots, w_{j-1}, w_j = v, \dots\}$$

$$T = \{w_1, \dots, w_i = v, w_{i+1}, \dots, w_{j-1}, w_j = u, \dots\}$$

Para ambos casos, a subtrilha de $w_i \dots w_j$ representa uma trilha entre os vértices u e v . A partir de uma trilha entre quaisquer dois vértices, podemos derivar um caminho, seguindo o seguinte método:

Método para derivar um caminho de um passeio qualquer

Seja P um passeio que liga dois vértices quaisquer, mostraremos como construir um caminho ligando esses mesmos vértices a partir de P .

Como P é um passeio, possivelmente ele percorre um mesmo vértice mais que uma vez. Do contrário, já podemos considerar P um caminho, finalizando o método.

Enquanto houver um vértice v percorrido mais que uma vez por P executa-se o seguinte passo:

Retiramos de P todos os vértices entre a primeira e a última aparição do vértice v , mantendo apenas tal vértice:

Portanto um passeio P qualquer com repetição do vértice v , como o seguinte:

$$P = \{u_1, \dots, u_i, v, \dots, v, u_j, \dots, u_n\}$$

Passa a ser:

$$P = \{u_1, \dots, u_i, v, u_j, \dots, u_n\}$$

P continua sendo um passeio, já que é garantido que u_i liga-se a v e v liga-se a u_j , porém agora P possui apenas uma aparição do vértice v . Repete-se tal passo até que P não percorra um mesmo vértice duas vezes, se tornando assim, por definição, um caminho.

Esse procedimento pode ser utilizado para encontrar um caminho entre quaisquer dois vértices de T , mas como T é euleriano, possuindo todos vértices de G , prova-se assim que G é fracamente conexo.

Basta, portanto, provar que G possui todos os vértices w tal que $\delta^+(w) = \delta^-(w)$ ou que existe um par de vértices u, v tal que $\delta^+(u) - \delta^-(u) = 1$, $\delta^-(v) - \delta^+(v) = 1$ e $\delta^+(w) = \delta^-(w)$ para todo $w \in V(G) \setminus \{u, v\}$.

Analisaremos dois casos em relação à trilha euleriana T de G :

- Se T é um circuito:

$$T = \{v, w_1, w_2, \dots, w_n, v\}$$

O vértice v é o vértice inicial e final de T , mas também pode estar presente internamente em T . Considere os vértices w_i como vértices diferentes de v , internos a T .

Todo vértice w_i possui, em cada uma de suas aparições em T , um vértice que o precede e um vértice que o sucede, conseqüentemente em cada aparição de w_i em T , dois arcos ligados a w_i podem ser contabilizados: um entrando em w_i e outro saindo do mesmo. Isso implica que os arcos que entram e saem de um vértice interno a T são sempre contabilizados em pares, mantendo assim a igualdade $\delta^+(w_i) = \delta^-(w_i)$.

Analisaremos o vértice v a parte:

No início de T , há uma aparição de v em que contabilizamos uma única aresta saindo de v , o que adiciona uma unidade ao grau de saída do mesmo $\delta^-(v)$.

As aparições de v como vértice interno de T contribuem igualmente para o grau de entrada e saída do mesmo, similarmente ao que ocorre com os vértices w_i .

No fim de T , há outra aparição de v , que contabiliza uma única aresta entrando em v , adicionando uma unidade ao grau de entrada do vértice $\delta^+(v)$.

Finalmente, nota-se que apesar de v ser um vértice extremo, seus graus de saída e entrada também se igualam, já que v é tanto o vértice inicial quanto o vértice final de T .

Provando assim que todos vértices de G possuem grau de saída e entrada iguais, satisfazendo a implicação do corolário.

- Devemos analisar agora o caso em que T não é um circuito.

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

A diferença neste caso é que os vértices extremos são diferentes. O vértice u , do início de T , possuirá exatamente uma unidade a mais de grau de saída em relação ao seu grau de entrada $\delta^-(u) - \delta^+(u) = 1$, enquanto que v , o vértice final de T , possuirá uma unidade a mais de grau de entrada em relação ao grau de saída, $\delta^+(v) - \delta^-(v) = 1$. Todos vértices w_i diferentes de u e v possuirão um valor igual de grau de entrada e saída, já que são apenas vértices internos de T .

Este caso também é válido, condizendo com a implicação do corolário.

(\Leftarrow) Seja G um digrafo fracamente conexo, com vértices u, v tal que $\delta^+(u) - \delta^-(u) = 1$, $\delta^-(v) - \delta^+(v) = 1$ e $\delta^+(w) = \delta^-(w)$ para todo $w \in V(G) \setminus \{u, v\}$, provaremos que deverá existir uma trilha euleriana em G .

Seja G' uma cópia do grafo G em que adiciona-se um arco de u a v . A adição de tal arco aumenta o grau de saída de u e o grau de entrada de v em uma unidade, fazendo com que $\delta^+(u) = \delta^-(u)$ e $\delta^+(v) = \delta^-(v)$.

Deste modo, todos vértices de G' possuem o mesmo grau de entrada e saída e G' continua sendo fracamente conexo.

Provaremos inicialmente que G' deverá ser fortemente conexo. Sejam w_i, w_j dois vértices quaisquer de G' , como este grafo é fracamente conexo podemos assumir, sem perda de generalidade, que existe ao menos um caminho de w_i a w_j . Provaremos que também deverá existir um caminho no sentido inverso, de w_j a w_i .

Seja T uma trilha maximal que percorre um dos caminhos de w_i a w_j :

$$T = \{w_1, w_2, \dots, w_i, \dots, w_j, \dots, w_n\}$$

Como T é maximal, podemos afirmar que todos os arcos que saem do vértice w_n são percorridos por T , já que do contrário T não seria maximal.

Além disso, como todos os vértices possuem grau de saída e entrada iguais em G' , deve valer que $w_1 = w_n$, já que do contrário $\delta^-(w_1) - \delta^+(w_1) = 1$ e $\delta^+(w_n) - \delta^-(w_n) = 1$. Consequentemente podemos reescrever a trilha T de outro modo, que chamaremos de T' , evidenciando a existência de uma trilha de w_j a w_i :

$$T' = \{w_j, w_{j+1}, \dots, w_n = w_1, w_2, \dots, w_i\}$$

A partir da trilha T' é possível derivar um caminho de w_j a w_i , como realizado anteriormente na demonstração deste corolário.

Prova-se assim que todo par de vértices w_i, w_j possui um caminho em G' de w_i a w_j e de w_j a w_i . G' , pela definição, é um grafo fortemente conexo.

Além disso, pelo teorema 1.2, G' é euleriano, possuindo assim um circuito euleriano C :

$$C = \{w_1, w_2, \dots, w_k = u, w_{k+1} = v, \dots, w_n, w_1\}$$

Podemos reescrever C do seguinte modo, tornando o arco artificial, de u a v , o último arco percorrido em C :

$$C = \{w_{k+1} = v, \dots, w_n, w_1, w_2, \dots, w_{k-1}, w_k = u, w_{k+1} = v\}$$

A partir de C podemos derivar uma trilha T , removendo apenas o arco artificial de C :

$$T = \{w_{k+1} = v, \dots, w_n, w_1, w_2, \dots, w_{k-1}, w_k = u\}$$

Ao remover de C o único arco que não pertencia ao grafo original G , criamos uma trilha euleriana T em relação a G , provando assim a volta do corolário.

□

2 O problema do Carteiro Chinês (PCC)

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:

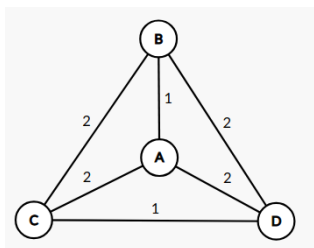


Figura 4: Exemplo de grafo

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

, 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.

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. Desconsideraremos nas análises a seguir grafos que possuam vértices isolados, isto é, vértices que não possuam arestas, já que tais vértices não afetam a solução 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 $k > 0$.

Analisaremos 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:

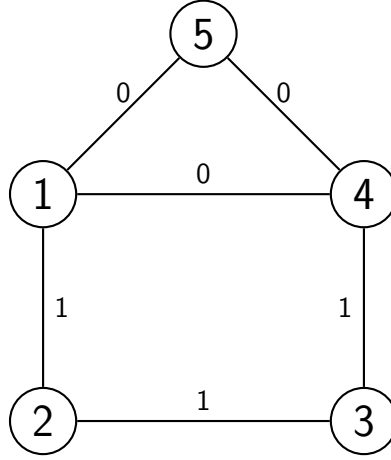


Figura 5: Possível estrutura em grafo com vértices de grau ímpar

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

Vale também o seguinte:

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

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.*

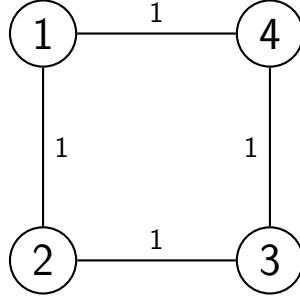


Figura 6: Caso em que o grafo tratado não possui vértices de grau ímpar

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.

□

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^* .

2.2 Grafos direcionados

Analisaremos agora o problema do carteiro chinês aplicado a digrafos.

Lema 5. *Um digrafo G possui solução para o problema do carteiro chinês se, e somente se, é fortemente conexo.*

Demonstração. (\Rightarrow) Começamos provando que para que um digrafo G possua solução para o problema do carteiro chinês é necessário que o mesmo seja fortemente conexo.

Vamos assumir, por absurdo, que G possui uma solução para o PCC mas não é fortemente conexo.

Como G possui uma solução para o PCC, então existe um passeio fechado T que passa por todos arcos de G .

Além disso, como T é um passeio fechado, para cada par de vértices u, v de G , sabemos que T passa por u e v . Assim, $T = \{v_0, v_1, \dots, v_i = u, \dots, v_j = v, \dots\}$.

A partir de um passeio entre dois vértices sempre é possível derivar um caminho entre os mesmos seguindo o seguinte método:

Método para derivar um caminho de um passeio qualquer

Seja P um passeio que liga dois vértices quaisquer, mostraremos como construir um caminho ligando esses mesmos vértices a partir de P .

Como P é um passeio, possivelmente ele percorre um mesmo vértice mais que uma vez. Do contrário, já podemos considerar P um caminho, finalizando o método.

Enquanto houver um vértice v percorrido mais que uma vez por P executa-se o seguinte passo:

Retiramos de P todos os vértices entre a primeira e a última aparição do vértice v , mantendo apenas tal vértice:

Portanto um passeio P qualquer com repetição do vértice v , como o seguinte:

$$P = \{u_1, \dots, u_i, v, \dots, v, u_j, \dots, u_n\}$$

Passa a ser:

$$P = \{u_1, \dots, u_i, v, u_j, \dots, u_n\}$$

Repete-se tal passo até que P não percorra um mesmo vértice duas vezes, se tornando assim, por definição, um caminho.

Sendo assim, afirmamos que todo par de vértices presentes em T possui um caminho entre si.

Já que não estamos considerando neste trabalho digrafos que possuem vértices isolados, cada vértice de G deve ser incidente ao menos a um arco. Além disso, como todos arcos são percorridos em T , já que ele é uma solução do PCC, todos os vértices deverão estar presentes no passeio fechado T . Por consequência para todo par de vértices de G existe um caminho entre eles o que define G como um grafo fortemente conexo, contradizendo a hipótese inicial.

Provando assim que um digrafo que possua solução para o PCC é necessariamente fortemente conexo.

(\Leftarrow) Provaremos agora que todo grafo fortemente conexo G possui uma solução para o PCC.

Seja v um vértice qualquer de G . Podemos construir uma solução P para o problema do carteiro chinês do seguinte modo:

Para qualquer arco uw de G definimos um circuito C_{uw} que passa por uw e pelo vértice v como a concatenação do caminho de v a u , do arco uw e do caminho de w a v .

A existência de um caminho entre quaisquer dois vértices é garantida, já

que G é fortemente conexo, garantindo assim a existência de C_{uv} .

Para construir a solução P , finalmente, basta realizar a concatenação dos circuitos C_e para todo arco e presente em G .

Deste modo garantimos que toda aresta de G é percorrida, provando assim a volta do lema: todo digrafo fortemente conexo possui uma solução para o problema do carteiro chinês.

□

Em linhas gerais, o algoritmo que soluciona o problema do carteiro chinês para digrafos envolve multiplicar arestas do grafo original até que o mesmo se torne euleriano. O circuito euleriano desse digrafo modificado será a solução do problema do carteiro chinês para o digrafo original, do modo similar à solução para o PCC em grafos não direcionados.

Uma grande diferença na solução do problema do carteiro chinês é que, para digrafos, não vale o lema 2, que garante a existência de uma solução para todo PCC em que cada aresta do grafo é percorrida no máximo duas vezes. Um contra-exemplo disso é o caso ilustrado na figura 7.

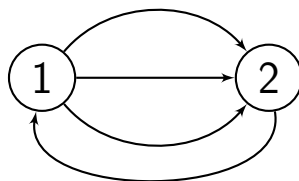


Figura 7: Contra exemplo do lema 2 para os digrafos

Para que um passeio fechado percorra os três arcos de 1 a 2 da figura 7, é necessário que esse mesmo passeio percorra três vezes o arco de 2 a 1.

O lema 2 é utilizado para justificar a utilização de um algoritmo de emparelhamento perfeito para definir quais arestas duplicar na solução do PCC. Como tal lema não vale para digrafos, a escolha dessas arestas deverá ser feita de um modo diferente, utilizando um algoritmo de fluxo máximo de custo mínimo, como veremos a seguir, no detalhamento do algoritmo que soluciona o PCC.

Solução:

A solução do PCC para um digrafo G fortemente conexo pode ser dividida em 4 passos:

- 1º Definimos como F o conjunto dos vértices que possuam grau de saída maior que o grau de entrada e S o conjunto dos vértices que possuam grau de entrada maior que o grau de saída.

Computamos então o custo do menor caminho entre todos os pares de vértices u, v onde $u \in F$ e $v \in S$.

- 2º Modelamos então um problema de transporte: Os vértices de F serão vértices de oferta, cada vértice $u \in F$ terá que escoar $\delta^-(u) - \delta^+(u)$ unidades de fluxo, já os vértices de S serão vértices de demanda, cada vértice $v \in S$ terá que receber $\delta^+(v) - \delta^-(v)$ unidades de fluxo.

Todo par de vértices u, v com $u \in F$ e $v \in S$ será ligado por um arco de capacidade infinita e custo igual ao custo de um menor caminho de u a v , calculado no primeiro passo.

Deve-se então resolver tal problema de transporte minimizando o custo total.

- 3º A partir de uma solução ótima encontrada para o problema de transporte devemos derivar um digrafo euleriano baseado em G .

Cada arco uv criado na modelagem do problema de transporte representa um caminho de mesmo custo no digrafo G entre os vértices u e v . Se um arco da modelagem é escolhido para a solução que estamos analisando, então todos arcos do caminho que tal arco condensado representa devem ser duplicados no digrafo G .

Após a duplicação dos arcos de um caminho entre dois vértices u e v a diferença absoluta dos graus de entrada e saída de u e v diminuirão em uma unidade, essa mesma diferença de graus se manterá constante para vértices internos ao caminho.

Ao final das duplicações necessárias, todos vértices de G possuirão um mesmo valor de grau de entrada e saída, sendo assim euleriano segundo o teorema 1.2.

- 4º Encontrar o circuito euleriano do novo grafo G , que será também a solução do problema do carteiro chinês para o grafo original G .

Aplicaremos agora o passo a passo apresentado acima em um exemplo.

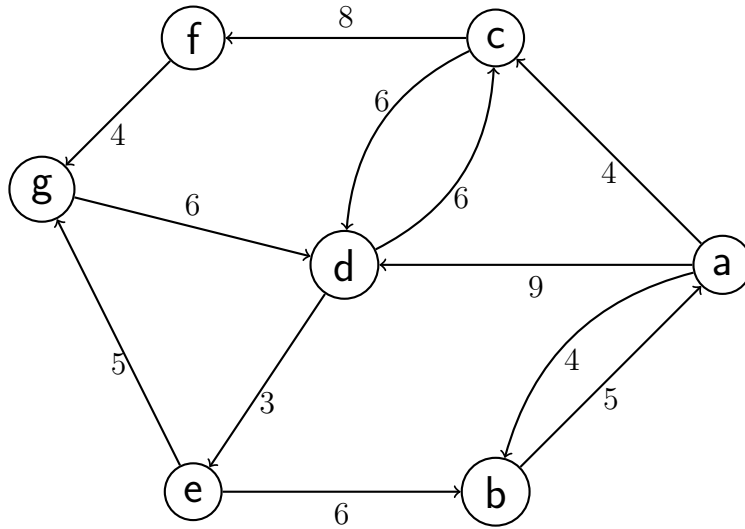


Figura 8: Digrafo retirado de exercício do site do MIT[9]

1º Definimos então os conjuntos F e S :

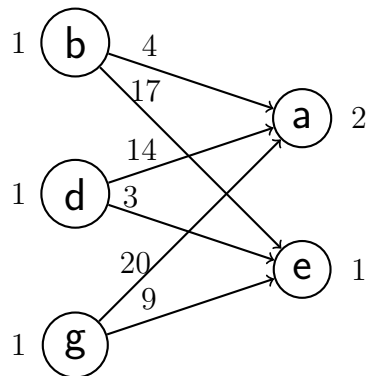
$$F = \{b, d, g\}$$

$$S = \{a, e\}$$

Computamos também a distância mínima entre os pares de vértices dos dois conjuntos:

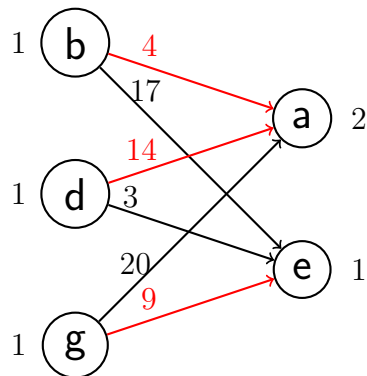
	a	e
b	4	17
d	14	3
g	20	9

2º Modelamos então o problema de transporte, os vértices b, d, g escoando uma unidade de fluxo cada e os vértices a, b com demandas 2 e 1, respectivamente.



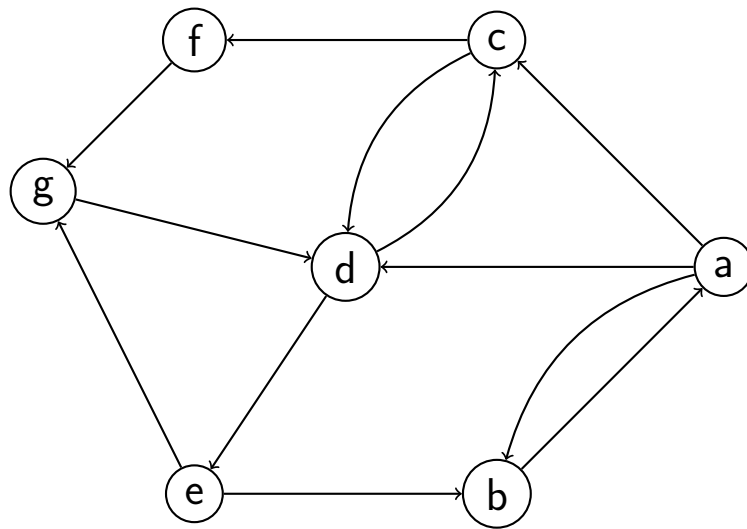
Os arcos definidos entre os vértices de F e S possuem capacidade infinita e custo igual à distância mínima dos mesmos vértices em G , como já calculado no primeiro passo.

Uma solução ótima para o problema apresentado consiste em utilizar os arcos de custo 4, 14 e 9, escoando assim toda a demanda dos vértices b, d e g .

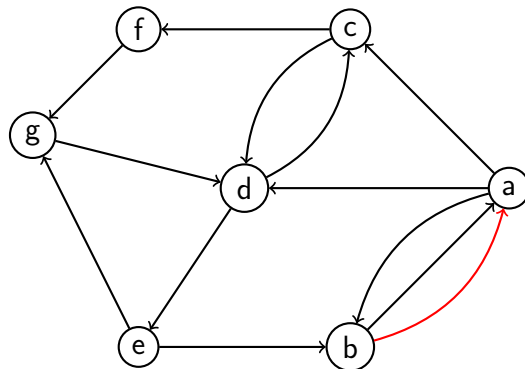


3º Agora devemos realizar a duplicação dos caminhos representados pelas arestas ba, da e ge .

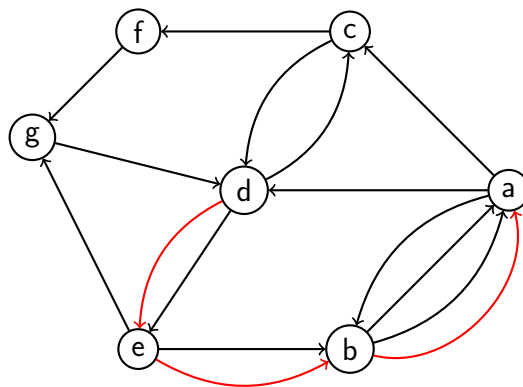
Representamos o estado inicial do digrafo G , sem os pesos nos arcos:



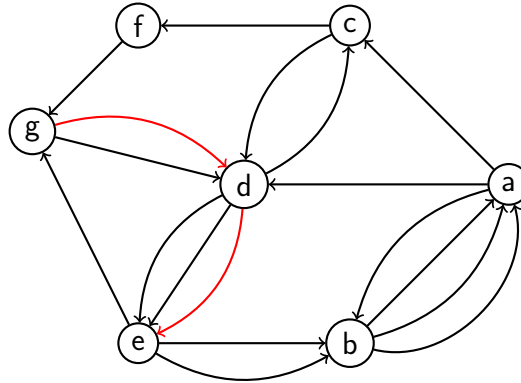
Faremos tal duplicação passo a passo, começando a duplicar os arcos do caminho mínimo de b a a . Representamos as arestas adicionadas em vermelho.



Agora duplicaremos os arcos do caminho mínimo de d a a :



Finalmente, duplicamos os arcos do caminho mínimo de g a e :



As sucessivas duplicações de caminho realizadas em G fazem com que todos os vértices possuam grau de saída e entrada iguais. Pelo teorema 1.2 temos que o novo grafo é euleriano. No quarto passo derivaremos um circuito euleriano para o digrafo construído.

4º Um possível circuito euleriano \mathcal{C} que se inicia no vértice a é o seguinte:

$$\mathcal{C} = \{a, c, f, g, d, c, d, e, g, d, e, b, a, d, e, b, a, b, a\}$$

Representamos tal circuito no grafo abaixo, indicando em cada arco a ordem em que o mesmo é percorrido:

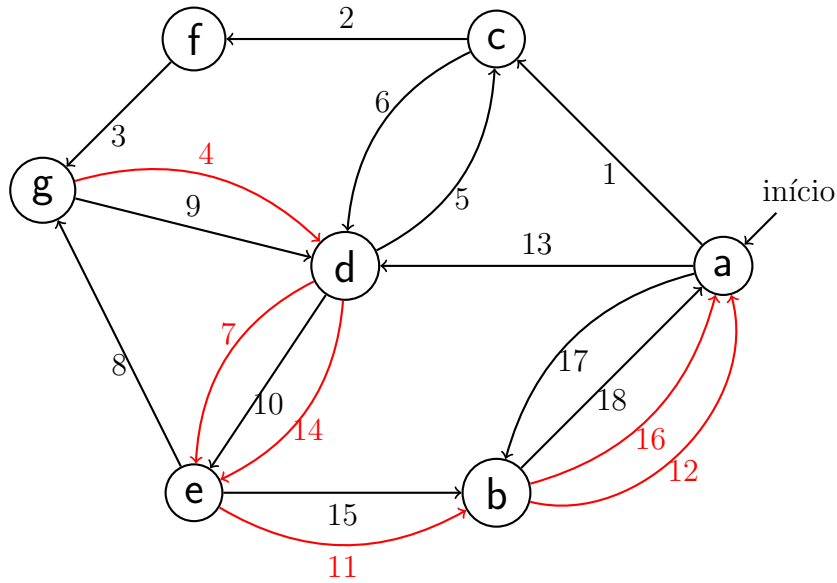


Figura 9: Representa-se em preto os arcos presentes no grafo G original, e em vermelho os arcos resultantes das duplicações realizadas no terceiro passo da solução.

O circuito \mathcal{C} percorrido no grafo original G representa o passeio fechado que soluciona o Problema do Carteiro Chinês, já que percorre todo arco de G ao menos uma vez.

Analisando o peso dos arcos de G concluímos que a solução \mathcal{C} tem custo $66 + 27 = 93$, 66 é o custo de se percorrer todas os arcos de G , representados em preto, uma única vez, e 27 é o custo de se percorrer alguns arcos mais que uma vez.

2.3 Grafos mistos

Já analisamos o Problema do Carteiro Chinês aplicado a grafos e digrafos, agora discutiremos o mesmo problema aplicado a grafos mistos, ou seja, grafos que tenham tanto arestas quanto arcos.

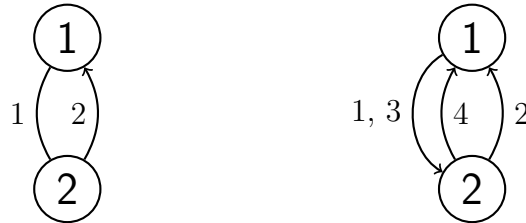
Ao contrário do que uma análise breve pode sugerir, não é possível reduzir o PCC para grafos mistos ao caso de grafos direcionados simplesmente transformando as arestas não direcionadas em dois arcos de sentidos opostos.

Um contra-exemplo para este pensamento é o seguinte:



Na esquerda temos um grafo não direcionado qualquer, e na direita temos o mesmo grafo com sua aresta substituída por dois arcos em sentidos opostos.

Apesar da similaridade dos grafos, suas soluções para o problema do carteiro chinês diferem. Uma solução do problema para o primeiro grafo é $\{1, 2, 1\}$, enquanto que uma solução possível para o segundo grafo é $\{1, 2, 1, 2, 1\}$, como representado visualmente a seguir.



Apesar de conseguirmos encontrar uma solução para o caso direcionado derivado, não há um modo trivial de transformar uma solução para tal caso em uma solução do caso não direcionado.

Diferentemente dos grafos ou digrafos, o PCC aplicado a grafos mistos não possui solução polinomial, sendo um problema NP-hard.

2.4 Problema do carteiro chinês rural

Nesta variação, além do grafo $G(V, A)$, sendo V o conjunto de vértices e A o conjunto de arestas, uma função de custo $c(e), e \in A$, é também necessário se especificar um subconjunto $R \subseteq A$.

Nesta versão mais geral do PCC devemos realizar o planejamento de uma rota de custo mínimo que se inicia e termina no mesmo vértice e passa ao menos uma vez por toda aresta pertencente a R . Isto é, não há necessidade de se percorrer as arestas de $A \setminus R$.

Denomina-se esta variação de Problema do Carteiro Rural (PCR).

Esse problema possui aplicações mais gerais que a sua versão original, por exemplo: pode-se modelar a rota de um carteiro que não necessariamente precisa percorrer todas as ruas de uma cidade, ou mesmo a rota de um ônibus escolar, que precisa passar apenas nas ruas dos alunos que ele atende, e não necessariamente por todas ruas de um bairro.

Assim como a variação do PCC para grafos mistos, o PCR é um problema NP-hard.

De modo similar aos outros casos do PCC, a solução padrão para este problema consiste em estender G adicionando cópias de arestas de modo que o mesmo se torne euleriano e em sequência construir uma solução a partir do circuito euleriano do grafo estendido.

Para resolver a primeira parte do problema, criar uma extensão de G que seja euleriana, serão apresentadas tanto heurísticas quanto discussões das soluções ótimas existentes.

2.4.1 PCR em grafos não direcionados

Heurística da árvore geradora mínima

Em 1979 Christofides sugeriu um algoritmo aproximado para o problema do caixeiro viajante que possui o melhor fator de aproximação para grafos gerais provada [10].

Uma adaptação de tal algoritmo foi desenvolvida e pode ser aplicada ao PCR.

Pode-se provar que tal algoritmo possui uma razão de aproximação de $3/2$ para grafos cuja função de custo respeita a desigualdade triangular.

2.4.2 PCR em grafos não direcionados

Para resolver o PCR em grafos não direcionados define-se um grafo $G' = (V', A')$, onde $V' = \{u \in V : (u, v) \in R \text{ para algum } v \in V\}$. A' é obtido por adicionar ao conjunto A uma aresta (v_i, v_j) para todo par $v_i, v_j \in V'$ cujo custo é igual ao custo do menor caminho de v_i a v_j ; e retirar-se de A' duplicatas de arestas paralelas de mesmo custo e todas arestas $(u_i, u_j) \in A \setminus R$ tal que $c_{ij} = c_{ik} + c_{kj}$ para algum k .

Pode-se visualizar um exemplo de tal separação a seguir:

O conjunto de arestas R induz uma separação no grafo G' de n componentes conexas G_1, G_2, \dots, G_n . Ou seja, os conjuntos G'_i são compostos por subgrafos conexos maximais que possuem apenas arestas pertencentes a R .

Passo 1. Construir uma árvore geradora mínima em G' que conecte os subgrafos G_i ,

Passo 2.

Passo 3.

2.4.3 PCR em grafos direcionados

2.5 Problema do carteiro chinês com vento

A motivação dessa variação é que podem existir estradas com uma corrente de vento que o carteiro deve percorrer. A corrente de vento tanto pode ajudar o carteiro que anda no mesmo sentido que a corrente, quanto pode atrapalhar o carteiro que anda no sentido contrário.

A abstração em grafos dessa variação é que uma mesma aresta pode possuir custos diferentes dependendo do sentido que a mesma é percorrida.

Esta variação do problema do carteiro chinês também é 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.

ADICIONAR PROVA

2.6 Problema do Carteiro Hierárquico

Nesta variação são definidos k subconjuntos de arestas $\{A_1, A_2, \dots, A_k\}$. Como no problema do cartiro chinês original deve-se encontrar um passeio que passe por todas arestas de um grafo G , porém esse passeio só poderá percorrer uma aresta pertencente a um subconjunto A_j se o mesmo já tiver percorrido todas arestas pertencentes aos conjuntos A_i com $1 \leq i < j$.

Esta variação também é NP-hard.

3 Problemas resolvidos

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

3.1 Tanya and Password[3]

Este problema fornece um multiconjunto \mathcal{C} de n strings de tamanho 3 e pede para que se construa, se possível, uma string S de tamanho $n + 2$ com a restrição de que o conjunto das substrings de tamanho 3 da string S deve corresponder a \mathcal{C} .

Uma instância desse problema é a seguinte, para $n = 5$:

$$\mathcal{C} = \{aca, aba, aba, cab, bac\}$$

Uma string S que resolve este exemplo é $abacaba$, já que existe uma bijeção entre toda substring de tamanho 3 de S e \mathcal{C} . A string $abacab$, no entanto, não satisfaz a restrição do problema já que a mesma não possui duas substrings aba , como ocorre em \mathcal{C} .

A solução que será apresentada para este problema envolve a teoria de caminhos eulerianos apresentada.

Iniciaremos por explicar a modelagem realizada. Cada vértice do digrafo que construiremos representará um conjunto de duas letras.

Representaremos cada string $w \in \mathcal{C}$ por dois vértices ligados por um arco. Um dos vértices, que chamaremos de u , representa os dois primeiros caracteres de w , enquanto que o segundo vértice, v , representa os dois últimos caracteres de w , cria-se ainda um arco de u a v .

Para exemplificar tal procedimento tome $w = aca$. Criam-se primeiramente dois vértices, um representando a string ac e outro representando ca , e liga-se ambos com um arco, como representado em 10.

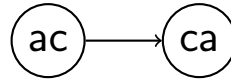
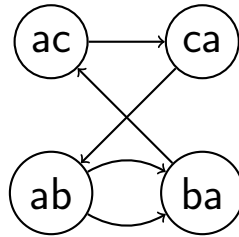


Figura 10: Exemplo da modelagem usada na solução

Deve repetir-se tal procedimento para toda string $w \in \mathcal{C}$. Segue a modelagem completa, que chamaremos de G , do exemplo inicial ($\mathcal{C} = \{aca, aba, aba, cab, bac\}$):



Cada arco criado na modelagem corresponde a uma string de \mathcal{C} , o arco entre ac e ca , por exemplo, representa a string aca . A partir dessa correspondência podemos representar também um passeio, P , em G como uma sequência de strings, seq , subconjunto de \mathcal{C} :

O passeio $P = \{ba, ac, ca, ab\}$, por exemplo, percorre os arcos que correspondem à sequência $seq = \{bac, aca, cab\}$.

Por sua vez, a partir de seq é possível montar uma string S que possua todas strings de seq como substrings:

$$seq = \{bac, aca, cab\} \rightarrow S = "bacab"$$

Tal procedimento permite que, a partir de um passeio P em G construa-se uma string S tal que:

- O tamanho de S é igual ao número de arcos percorridos em P mais 2;
- Se um arco e é percorrido em P , então a string de \mathcal{C} que e representa será uma substring de S .

Como o problema pede que encontremos uma string de tamanho $n + 2$ que possua todas strings de \mathcal{C} como substrings, basta encontrar, se existir, uma trilha que percorra todo arco de G uma única vez, isto é, uma trilha ou um circuito euleriano.

Deste modo a solução do problema consiste em checar as propriedades necessárias para a existência de uma trilha euleriana, que são estabelecidas neste trabalho pelo corolário 2.

Se uma trilha ou circuito euleriano existir em G , podemos usar o algoritmo de Hierholzer para encontrar tal passeio.

Esta solução foi implementada em C++ e está disponível no github[4].

3.2 Sereja and the Arrangement of Numbers[5]

Definição do 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 à 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 os 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.

Descrição da 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 , que possui x vértices, 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:

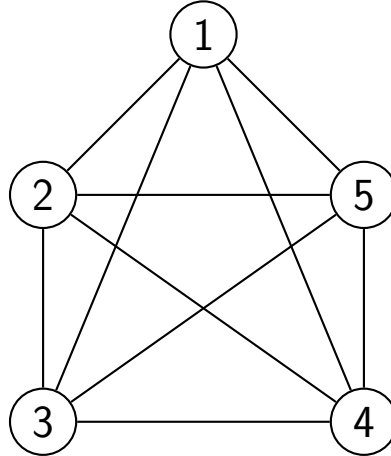


Figura 11: Exemplo com $x = 5$

Para $x = 5$, o grafo induzido é o K_5 , como representado na figura 11. Como todos os vértices do K_5 possuem grau par, a solução ótima do Problema do 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 12.

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 emparelhamento perfeito de custo mínimo. Basta apenas criar uma duplicação

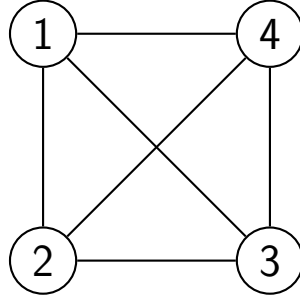


Figura 12: Exemplo com $x = 4$

de aresta entre todo par de vértices de índices i e $i + 1$ para todo índice i ímpar, como representado na figura 13 para o exemplo em questão.

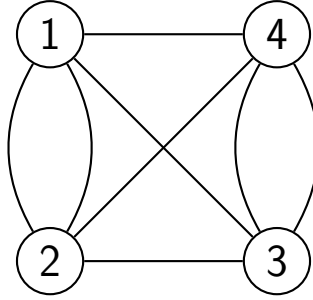


Figura 13: K_4 com arestas duplicadas

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:

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

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 $|P|$.

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 $|P| \leq n$.

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 possuímos 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 foi adicionada a referência [6].

3.3 Jogging Trails[7]

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.

Este problema é uma aplicação direta do problema descrito nesse trabalho. Entretanto, em virtude do limite do número de vértices das instâncias, é possível resolver o problema de maneira mais simples.

Um dos últimos passos do algoritmo que resolve o problema do carteiro chinês em grafos não direcionados descrito neste trabalho é realizar a condensação do grafo original usando os vértices de grau ímpar e posteriormente construir o emparelhamento perfeito de custo mínimo usando o algoritmo de Edmonds.

Uma alternativa viável para este problema é realizar a construção do emparelhamento perfeito usando, ao invés do algoritmo de Edmonds, um algoritmo de programação dinâmica:

Para esta solução devemos memoizar o custo mínimo de emparelhamento perfeito para todo subconjunto de vértices do grafo condensado.

Seja K o grafo completo condensado, S um subconjunto dos vértices de K , $dist$ uma função que retorna a distância entre dois vértices quaisquer de K e min_cost_S o menor custo de emparelhamento perfeito do subconjunto S de vértices, podemos definir a seguinte recorrência:

$$\min_cost_S = \begin{cases} 0 & S = \{\} \\ \min_{u,v \in S} \min_cost_{S-u-v} + dist(u,v) & \text{caso contrário} \end{cases}$$

Para representar os subconjuntos de vértices pode-se utilizar uma bit-mask. Como foi utilizado na solução[8] adicionada à referência.

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 D do round #288 (Div. 2) retirado do Codeforces codeforces.com/contest/508/problem/D
- [4] Solução para o problema Tanya and Password, desenvolvida em C++ github.com/gafeol/competitive-programming/blob/master/ojs/cf/508/D.cpp
- [5] Problema C do round #215 (Div. 1) retirado do Codeforces codeforces.com/problemset/problem/367/C
- [6] Solução para o problema Sereja and the Arrangement of Numbers, desenvolvida em C++ github.com/gafeol/competitive-programming/blob/master/ojs/cf/367/C.cpp
- [7] Problema 10296 retirado do UVa onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=1237
- [8] Solução para o problema Jogging Trails, desenvolvida em C++ github.com/gafeol/competitive-programming/blob/master/ojs/UVa/1237.cpp
- [9] Exemplo retirado do site do MIT, exercício 6.6.c web.mit.edu/urban_or_book/www/book/chapter6/problems6/6.6.html
- [10] Página da wikipedia sobre o algoritmo de Christofides wikipedia.org/wiki/Christofides_algorithm