1 PCR em grafos direcionados

Para tratar sobre a versão direcionada do PCR, tomaremos $G = (V, A)$ como o digrafo original e um conjunto $R \subseteq A$ de arcos que devem ser percorridos no circuito que resolve o PCR para G.

Assim como no PCR para grafos não direcionados, a solução deste caso se baseia em estender o digrafo G em um digrafo euleriano, cujo circuito euleriano representa uma solução para o PCR do digrafo original G.

A estensão de G em $G' = (V', A')$ se dá de modo similar àquela do caso anterior:

O conjunto de vértices possui a mesma definição, $V' = \{u \in V : (u, v) \in R \text{ para algum } v \in V\}$. De mesmo modo, A' será inicialmente acrescida dos arcos (v_i, v_j) para todo par $v_i, v_j \in V'$, de custo igual ao custo do menor caminho de v_i a v_j. Posteriormente remove-se de A' todo arco que pertence também a $A \setminus R$ e cujo custo c_{uv} é igual a $c_{uw} + c_{wv}$ para algum vértice w, removem-se também os arcos paralelos de mesmo valor que pertencem a $A \setminus R$.

O grafo G' será composto por k conjuntos conexos $G_1, G_2, \ldots, G_k$, induzidos por R. Isto é, cada conjunto G_i será composto apenas por arcos pertencentes a R e será conexo, mas não necessariamente fortemente conexos.

Será apresentada agora uma heurística que resolve esse problema, proposta por Christofides [?], muito semelhente à heurística apresentada para o caso de grafos não direcionados.

Heurística da arborescência geradora mínima

Passo 1. Encontrar uma arborescência geradora T de custo mínimo que conecte os subgrafos $G_1, \ldots, G_k$ e tem raiz em um vértice qualquer.

Para encontrar tal arborescência, pode-se utilizar o algoritmo proposto independentemente pela dupla Yoeng-Jin Chu e Tseng-Hong Liu e por Edmonds[?], sendo, portanto, chamado de algoritmo de Chu-Liu/Edmonds.

Seja $R \cup T$ o subgrafo de G' induzido pelos arcos de R e da arborescência T.

Passo 2. Encontrar um multiconjunto M de menor custo composto por arcos de A' que torna o subgrafo induzido por $R \cup T$ euleriano, ou seja iguala os graus de entrada e saída de todos vértices.

Pode-se determinar M a partir da resolução de um problema de transporte, segue a modelagem de tal problema: O problema de transporte será definido tendo como base o grafo G', porém as funções de oferta e demanda são definidas a partir dos graus dos vértices no grafo induzido por $R \cup T$. Um vértice $v \in V(G')$ cujo grau de entrada é maior que seu grau de saída (em relação ao subgrafo $R \cup T$) possuirá uma oferta igual ao valor absoluto da diferença dos graus, do contrário, o valor absoluto da diferença representará a demanda do vértice v.

A resolução deste problema gera um conjunto de caminhos, representando a distribuição ótima da oferta. O multiconjunto dos arcos pertencentes aos caminhos que solucionam o problema é o multiconjunto M desejado.

Passo 3. O digrafo induzido pelo multiconjunto de arcos $R \cup T \cup M$ é, pela definição de M, euleriano.

Portanto, pode-se derivar um circuito euleriano de tal grafo, a partir deste circuito euleriano, por sua vez, é possível derivar uma solução do PCR para o grafo original G.

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

[11] Artigo "Arc Routing Problems Part II: The Rural Postman Problem" publicado por Michel Gendreau e Gilbert Laporte. pubsonline.informs.org/doi/10.1287/opre.43.3.399

[12] Christofides, Nicos, et al. "An algorithm for the rural postman problem on a directed graph."Netflow at pisa. Springer, Berlin, Heidelberg, 1986. 155-166.
link.springer.com/chapter/10.1007/BFb0121091

[13] nvlpubs.nist.gov/nistpubs/jres/71B/jresv71Bn4p233_A1b.pdf