

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LEONARDO PERDOMO

Algoritmo de Cristofides

Porto Alegre, 2017

1.1 Tarefa

O objetivo deste trabalho foi a implementação prática em laboratório e a avaliação da qualidade do algoritmo de Cristófides, que obtém até $3/2_{opt}$ no pior caso de aproximação para o problema do caixeiro viajante (TSP) [1], devendo utilizar variantes usando emparelhamento perfeito máximo, por meio da biblioteca Blossom V, e de emparelhamento perfeito por um algoritmo guloso (*greedy*).

1.2 Solução

Para a solução do problema proposto foi implementado o algoritmo de Prim, para extração da árvore geradora mínima (MST), utilizada a biblioteca Blossom V e implementada uma alternativa gulosa para o emparelhamento perfeito máximo do grafo completo de vértices ímpares da MST, obtendo um grafo euleriano pela união das arestas emparelhadas à MST, implementado o algoritmo de Hierholzer para obtenção do circuito euleriano, e por fim, implementada a extração do circuito hamiltoniano pulando vértices repetidos do circuito euleriano (prática possível para grafos métricos). Para garantir a viabilidade na realização dos experimentos com as instâncias de grafos métricos fornecidas, foi realizada a implementação do trabalho tratando as arestas de forma implícita, com os vértices definidos por coordenadas permitindo o cálculo de distância euclidiana entre os mesmos. Foi identificado em laboratório que a biblioteca Blossom V possui algum *bug* em sua implementação geométrica que não permite a obtenção de resultados para algumas instâncias ("usa13509", "brd14051" e "d18512") pois sua execução não termina. Para contornar o problema e obter os resultados foi necessário uso com representação explícita das arestas por meio da opção *SolveComplete*, consequentemente aumentando de forma significativa o tempo de execução e consumo de memória do algoritmo para estas três instâncias.

1.3 Ambiente de teste

O dispositivo utilizado para realização dos testes foi o *notebook* pessoal do autor deste trabalho, um ASUS X451CAP com Intel i3 3217U de 1.8GHz, 4GB de RAM DDR3 de 798MHz com Ubuntu 16.04 64bit LTS.

1.4 Resultados

Os experimentos realizados consistiram no uso dos grafos métricos fornecidos, listados na tabela 1.1, como entrada para o algoritmo implementado nas configurações utilizando a biblioteca Blossom V ou uma abordagem *greedy* desenvolvida, para a etapa de emparelhamento perfeito dos vértices ímpares da MST. O primeiro vértice da entrada de cada grafo foi selecionado para construção da árvore geradora mínima usando Prim. Os resultados do TSP obtidos com o algoritmo foram tomados, sendo calculado seu desvio percentual em relação às melhores soluções conhecidas para cada grafo. O tempo de execução foi medido em 20 repetições para cada configuração desenvolvida.

Os resultados coletados foram analisados e são apresentados na tabela 1.1 para cada grafo e configuração de emparelhamento testados. O resultado do algoritmo consiste na distância física percorrida pelo circuito hamiltoniano nos grafos métricos, e seu desvio relativo percentual aos melhores valores conhecidos.

Tabela 1.1: Resultados do algoritmo de Cristofides

Instância (melhor)	<i>Perfect Matching</i>	Resultado	Desvio Relativo %	Tempo Médio (<i>ms</i>)	Desvio Padrão (<i>ms</i>)
berlin52 (7542)	Blossom V	8595	0,13961814	4,1	0,71818485
	<i>Greedy</i>	9514	0,26146911	2,45	1,39453822
vm1748 (336556)	Blossom V	381341	0,13306849	837	7,78662821
	<i>Greedy</i>	431757	0,28286823	818,25	8,84292759
pr2392 (378032)	Blossom V	426845	0,12912399	1218,85	3,36037279
	<i>Greedy</i>	452964	0,19821602	1183,75	10,73545528
pcb3038 (137694)	Blossom V	155310	0,12793586	1734	12,05251666
	<i>Greedy</i>	164143	0,19208535	1721,1	10,76495288
fnl4461 (182566)	Blossom V	207888	0,13870053	3337,25	14,33389431
	<i>Greedy</i>	215202	0,17876275	3181,65	16,203557111
rl5934 (556045)	Blossom V	612923	0,10229028	6460,7	18,15459924
	<i>Greedy</i>	677418	0,21827910	6433,8	36,72171307
rl5915 (565530)	Blossom V	621882	0,09964458	5749,4	22,53511879
	<i>Greedy</i>	680986	0,20415539	5746,05	35,90407542
usa13509 (19982859)	Blossom V*	22562104	0,12907287	36088,1	1159,36220670
	<i>Greedy</i>	24005622	0,20131068	27560,4	95,33619184
brd14051 (469385)	Blossom V*	532967	0,13545810	41465,95	1640,88636109
	<i>Greedy</i>	553589	0,17939218	25548,15	96,18199580
d18512 (645283)	Blossom V*	734937	0,13893749	77365,7	1973,88119143
	<i>Greedy</i>	759937	0,17768018	41350,6	171,07443748

* Devido ao *bug* no *perfect matching* geométrico da Blossom V, este resultado foi obtido com a opção *SolveComplete*, impactando no seu tempo médio de execução.

Foi observado que o algoritmo respeitou a aproximação de até $3/2_{opt}$ no pior caso, tendo atingido desvio relativo ao valor ótimo mais alto no grafo vm1748 com emparelhamento perfeito guloso (0,28286823), e mais baixo para o grafo rl5915 usando a Blossom V (0,09964458). Foram atingidas soluções mais próximas ao melhor valor conhecido utilizando Blossom V em relação à *greedy*, sem aumento significativo em tempo médio de execução do algoritmo, excetuando nas três maiores instâncias onde um *bug* da biblioteca, comentado em aula, exigiu o uso de solução que carrega todas as arestas em memória e, consequentemente, aumenta significativamente o tempo de execução do algoritmo.

1.5 Conclusão

Através da implementação prática e experimentação conduzidas neste trabalho, foi possível observar que o algoritmo de Cristofides implementado respeita a condição de até $3/2_{opt}$ no pior caso de aproximação para o problema do caixeiro viajante (TSP). O impacto da abordagem de emparelhamento perfeito dos vértices ímpares da MST foi visível com resultados menos distantes do valor ótimo conhecido obtidos com a Blossom V em relação à alternativa gulosa, em tempos médios de execução semelhantes.

REFERÊNCIAS

- [1] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem” Technical Report 388, Carnegie-Mellon University, 1976.