Trabalho 4 - Laboratório de programação Matemática e Otimização Combinatória[1]

Luiz Henrique Rigo Faccio[2]
Robert Biasoli Drey[3]

Resumo

O trabalho objetiva descrever matematicamente o problema do conjunto independente máximo e adicionar restrições à sua representação matemática. É proposta uma implementação em Python de um código capaz de resolver este problema de programação linear inteira e, além disso, é feita uma comparação entre a eficiência de o modelo de programação linear básico e o modelo fortalecido.

1. PROBLEMA DO CONJUNTO INDEPENDENTE MÁXIMO

O problema do conjunto independente máximo consiste em encontrar o maior conjunto independente possível no grafo, ou seja, o subconjunto de vértices S com o maior número de vértices, onde nenhum vértice em S está conectado diretamente a outro vértice em S.

O problema do conjunto independente máximo é NP-difícil, o que significa que, até onde se sabe, não existe um algoritmo eficiente (polinomial) para resolvê-lo em grafos gerais. No entanto, o problema tem várias aplicações práticas, incluindo otimização de redes, problemas de alocação de recursos, bioinformática, e muitos outros.

1.1. REPRESENTAÇÃO MATEMÁTICA

É possível representar este problema de forma matemática através de:

Maximizar: $\sum_{i=1}^{n} x_i$ (Função Objetivo)

Sujeito à: $x_i + x_j <= 1$, $\forall_{i,j}$ tal que os vértices i e j sejam vizinhos e $x_i \in \{0, 1\}$ com i = {1, . . . , n}.

Aonde x_i identifica se um vértice i faz parte do conjunto final dos vértices independentes (1 = Faz parte; 0 = Não faz parte).

[1] Trabalho apresentado como requisito parcial para aprovação na disciplina de Tópicos Especiais em Computação XXXVIII, no curso de Ciência da Computação da Universidade Federal da Fronteira Sul (UFFS)

[2] 2211100003 - luiz.faccio@estudante.uffs.edu.br.
[3] 2211100013 - robert.drey@estudante.uffs.edu.br.

1.2. ADIÇÃO DE RESTRIÇÕES

A representação matemática do item 1.1 é o suficiente para descrever o problema, entretanto, é possível adicionar mais restrições no modelo para que ele se torne mais eficiente.

O processo de fortalecimento do modelo, neste contexto, significa identificar cliques[4] dentro do grafo e adicioná-los, de forma matemática, como restrições no modelo. Isto é, se um dos vértices dentro de um clique for escolhido para estar no conjunto final, nenhum outro vértice do mesmo clique pode ser escolhido também.

Por exemplo, se o clique **{1, 2, 3}** existir no grafo, é possível adicionar a seguinte restrição: $x_1 + x_2 + x_3 <= 1$. Esta restrição significa, em suma, que apenas um dos vértices daquele clique pode fazer parte da solução válida.

Cliques de vértices únicos ou de dois vértices não são muito úteis na etapa de adição de restrições extras pois suas representações já são capturadas pela representação matemática base do problema (1.1).

2. EXECUÇÃO E DOCUMENTAÇÃO

Para cada uma das 8 instâncias do problema do conjunto independente máximo foram executadas dois algoritmos resolvedores:

1. Básico: algoritmo que utiliza de um modelo matemático simples, como descrito na seção 1.1.

2. Fortalecido: algoritmo que utiliza de um modelo matemático fortalecido com **uma** restrição extra, descrito na seção 1.2. A restrição escolhida varia de acordo com o tamanho dos cliques encontrados no grafo. A restrição extra representará sempre o maior dos cliques encontrados.

A tabela 1 documenta as informações de execução de cada uma uma das oito instâncias.

3. CONCLUSÃO

De acordo com a tabela de execução dos algoritmos, observando os tempos de execução e o número de nós explorados nas árvores, fazem-se algumas considerações.

O modelo básico tem tempos de execução menores em instâncias mais simples e explora menos nós, e conforme as instâncias aumentam de complexidade, sua eficiência cai, aumentando significativamente o tempo de execução.

[4] Um clique é um subconjunto de vértices que estão adjacentes dois a dois. Ou seja, para cada par de vértices distintos do conjunto, existe uma aresta que os liga.

Tabela 1 – Documentação da execuçãao dos algoritmos

Instância	Valor ótimo	Modelo Básico		Modelo Fortificado	
		Tempo de Execução	**Num. de nós da árvore**	**Tempo de Execução**	**Num. de nós da árvore**
1	32	<1 seg	1	<1 seg	1
2	14	1 seg	9	<1 seg	5
3	8	2 seg	277	<1seg	121
4	11	21 seg	10.869	573 seg	609.585
5	58	10 seg	1	5 seg	1
6	30	278 seg	155.642	427 seg	447.377
7	16	67 seg	17.297	100 seg	30.253
8	100	3464 seg	470.713	3326 seg	640.883

O modelo fortificado age de diferentes maneiras de acordo com o tamanho das instâncias. Nas menores, tende a ter um desempenho superior ou equivalente ao modelo básico, explorando uma quantidade significativamente maior de nós. Nas instâncias maiores, seu desempenho parece equivalente, por vezes, inferior ao modelo básico.

Em relação a implementação dos códigos, existem duas versões fortalecidas:

1. Adiciona apenas uma restrição ao modelo, é a versão utilizada na seção z2;

2. Adiciona o máximo de restrições possíveis. Esta versão procura todos os cliques de um grafo e adiciona-os como restrições no modelo. Esta versão não foi utilizada em virtude do tempo de execução elevado.