Documentação TP1

Jorge Augusto de Lima e Silva

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais

(UFMG) – Belo Horizonte – MG – Brasil

Professores: Jussara Marques de Almeida – Disciplina: Algoritmos I – Semestre: 2022/2

E-mail: jorgesilva@dcc.ufmg.br

1. Introdução:

O problema proposto toma a seguinte forma: um candidato está fazendo um experimento para descobrir quais propostas devem incluir na campanha de forma que atraia o maior número possível de eleitores.

Esse experimento consiste em reunir um grupo de pessoas e, cada pessoa irá avaliar as propostas. Avaliadas as propostas, cada pessoa irá indicar entre 0 e 2 que deseja manter na campanha, e entre 0 e 2 que deseja que sejam removidas da campanha.

A partir do feedback dado pelas pessoas, o candidato considera que cada pessoa estará satisfeita se, dentre as propostas indicadas por essa, uma das propostas que a pessoa gostou fosse mantida na campanha e uma das que a pessoa não gostou não fosse usada na campana.

Uma vez que temos o feedback e o conceito de satisfabilidade, o candidato deseja saber se, dentre o conjunto das suas propostas, existe um subconjunto que ele possa usar durante a campanha (não usando, então, as propostas que não fazem parte desse subconjunto) tal que todas as pessoas que participaram do experimento fiquem satisfeitas.

2. Modelagem:

Considerando o feedback dado por um indivíduo, temos que ele enumera até quatro propostas, duas para serem aceitas (a1 e a2), e duas para serem rejeitadas (r1 e r2). Assumindo que se uma proposta não é aceita ela é rejeitada, podemos assinalar símbolos para indicar como uma proposta foi assinalada. Se definirmos que Falso significa que uma proposta foi rejeitada e Verdadeiro que ela foi aceita, temos, assim, que as variáveis a1, a2, r1, r2 só podem assumir esses dois valores, sendo, então, variáveis de uma expressão booleana.

Agora, levado em consideração a assinalação de valores definida, podemos avaliar a satisfação do indivíduo na forma de uma expressão booleana, que retorna Verdadeiro quando ele está satisfeito e Falso quando não. Temos, portanto, a seguinte expressão booleana:

$$(a1+a2)(!r1+!r2);$$

a qual trás o seguinte significado: retorne verdadeiro se a proposta a1 foi aceita ou a proposta a2 foi aceita, ao mesmo tempo que a proposta r1 foi rejeitada ou a proposta r2 foi rejeitada.

Alguns pequenos detalhes que devem ser explicitados ocorrem quando o feedback do indivíduo retorna menos do que quatro propostas. No caso de ele indicar apenas uma proposta para ser aceita, essa proposta tem que ser obrigatoriamente aceita para que o indivíduo fique satisfeito. O análogo é válido para o caso de ele definir apenas uma proposta para ser rejeitada. O outro caso é quando ele não define nenhuma proposta para ser aceita, o que implica que nenhuma

proposta em específico deve ser aceita para que esse indivíduo seja satisfeito, sendo a recíproca também válida para quando ele não indica nenhuma proposta para ser rejeitada.

Uma vez definida a expressão de satisfabilidade para um único indivíduo, desejamos que todos os indivíduos fiquem satisfeitos simultaneamente, que é logicamente equivalente a dizer que o primeiro indivíduo está satisfeito e o segundo está satisfeito e o terceiro e o quarto e assim até o último indivíduo. Essa satisfabilidade geral pode ser modelada como a concatenação de todas as expressões booleanas que satisfazem cada um dos indivíduos. Desta forma, dado que o segundo indivíduo indicou as propostas a3 e a4 para serem aceitas e as propostas r3 e r4 para serem rejeitadas, o terceiro indicou a5 e a6, r5 e r6, assim até chegar no último indivíduo, que indicou as propostas ai e aj, ri e rj. Por fim, a expressão booleana final que obtemos é a seguinte:

$$(a1+a2)(!r1+!r2)(a3+a4)(!r3+!r4)...(ai+aj)(!ri+!rj).$$

Definida a equação geral de satisfabilidade, o que desejamos saber é se existe alguma assinalação de variáveis, ou seja, aceitação e rejeição de propostas, tal que que a expressão retorne verdadeiro, em outras palavras, queremos saber se é possível satisfazer essa equação.

O problema de satisfabilidade de expressões booleanas é um problema famoso no mundo da computação, sendo este um problema np-complexo, ou seja, que necessita de um algoritmo com complexidade assintótica exponencial para ser resolvido. Porém, existe uma instância deste problema que pode ser modelada de forma que exista um algoritmo polinomial capaz de resolvê-la. Essa instância é o 2-Sat, na qual a expressão booleana se encontra na forma produto de somas e todas as somas tenham no máximo duas variáveis. Este é o caso da expressão encontrada, ou seja, existe um algoritmo em tempo polinomial capaz de responder a questão feita pelo prefeito.

A modelagem do problema segue o seguinte algoritmo, observando ele apenas dentro de uma soma de variáveis, temos que a1+a2 tem o mesmo significado de dizer que, se a1 foi rejeitado, ou seja, !a1 for Verdadeiro, então a2 obrigatoriamente deve ser escolhido; o mesmo vale para a2, ou seja, se a2 for rejeitado, então a1 obrigatoriamente deve ser escolhido.

Definida a forma como será tratada uma única soma, expandimos o método para todas as somas no contexto. Assim, obteremos relações transitivas, uma vez que a aceitação de uma proposta pode ser definida pela aceitação ou não de uma outra proposta, ao mesmo tempo que esta também pode vir a definir a aceitação ou não de outras propostas.

Sabendo então desta transitividade, temos que se a aprovação da proposta a1 resulta, transitivamente, que essa mesma proposta deva ser rejeitada, então a única opção restante para a1 é sua rejeição, de modo que a cadeia de aceitações nunca comece e não se chegue numa contradição. A recíproca também é verdadeira, ou seja, se a rejeição da proposta a1 leva também a aceitação da proposta a1, o único valor possível que a1 pode assumir é verdadeiro, para que não haja uma contradição.

O problema surge então quando a aprovação de uma proposta implica na rejeição da mesma, ao mesmo tempo que a rejeição desta implica na sua aprovação, de modo que teremos inevitavelmente uma contradição, a qual resulta em pelo menos uma das somas sendo Falsa na expressão de satisfabilidade, o que, por sua vez, implica que algum dos indivíduos vai ficar, inevitavelmente, insatisfeito com o resultado.

Agora que sabemos a condição com a qual é impossível satisfazer a expressão, podemos armazená-la em uma estrutura de dados para processá-la. A estrutura escolhida foi um grafo direcionado. Supondo que o candidato possua N propostas, o grafo terá então 2N vértices, um

para representar a aceitação e outro para representar a rejeição de cada uma das propostas. As arestas desse grafo são as relações de implicação descritas acima, ou seja, se temos que uma aresta sai de um vértice A para um vértice B do grafo, temos que se assinalarmos o vértice A como sendo verdadeiro, o vértice B também será assinalado como verdadeiro.

Tendo então a forma como as assinalações devem ser feitas armazenadas no grafo, devemos verificar se existe a contradição citada a cima em alguma das propostas, ou seja, se existe um caminho que leve da aceitação de uma proposta até sua rejeição, ao mesmo tempo que existir um caminho que leve da sua rejeição até sua aceitação.

A forma de verificar essa existência mútua de caminhos pode ser feita encontrando as componentes fortemente conectadas do grafo, ou seja, subgrafos do grafo onde existe um caminho de um vértice para o outro independente do vértice de partida e do destino. A aceitação de uma proposta estar no mesmo componente da sua rejeição implica, portanto, que existe um caminho da sua aceitação até sua rejeição, e um caminho de sua rejeição até sua aceitação.

Para encontrar os componentes fortemente conectados, foi usado o algoritmo de Kosaraju, o qual consiste em realizar uma busca em profundidade no grafo até que os tempos de término de todos os vértices estejam computados, e, em seguida, no grafo transposto, realizar novamente a busca em profundidade, sendo a fonte sempre o vértice que não faz parte de nenhum componente no momento e que tem o maior tempo de término, de modo que a árvore gerada por esse seja um componente fortemente conectado.

Desta forma, sabendo as condições de satisfabilidade do problema e o algoritmo que permite dizer se ele é satisfazível ou não, podemos então executá-lo e obter uma resposta, finalizando assim a modelagem do problema.