

5

Conclusões e Trabalhos Futuros

Esta tese teve a finalidade de estudar e aprimorar técnicas de compressão de dados. A principal motivação deste estudo é a necessidade de armazenar, comunicar e computar uma grande quantidade de dados utilizados por aplicações. Os tipos de dados abordados foram as cadeias binárias e os grafos web. Apresentamos a seguir as conclusões e trabalhos futuros para cada um desses tipos de dados.

Compressão de seqüência de bits Abordamos o problema de comprimir seqüência de bits e avançamos nesta área ao demonstrarmos a relação entre algoritmos de intercalação e codificadores de fonte binária.

Para demonstrá-la, criamos uma função de conversão que associa um algoritmo de intercalação a um codificador de fonte binária. Ou seja, provamos que *qualquer* algoritmo de intercalação pode ser convertido em um codificador de fonte binária. Em específico, se um algoritmo de intercalação é ótimo no modelo de comparações então o codificador de fonte também é ótimo.

Em seguida, aplicamos a função de conversão aos seguintes algoritmos de intercalação binário (Hwa72), recursivo (Dud81) e probabilístico (Veg93), que gerou respectivamente os codificadores de entropia baseado em comprimentos de carreiras codificados com o código de Rice (Ric79), o codificador de intercalação binária (Mof00) e o codificador de Rice aleatório, na qual é um novo variante do código de Rice. Estes codificadores de fonte pertencem a uma nova classe que intitulamos de *Codificadores de Fonte de Intercalação (Merge Source Coding)* (Avi09).

Mais investigações da conexão estabelecida aqui pode levar à novos e úteis codificadores de fontes. No entanto, o estado da arte em teoria da informação está avançado e propor codificadores com novas abordagens é uma tarefa difícil. Percebemos que a maior contribuição nesta área reside na função inversa de conversão, ou seja, no mapeamento de codificadores de fonte em algoritmos de intercalação. De fato, algoritmos podem ser vistos como se estivessem codificando uma fonte de dados. E assim, podemos desenvolver algoritmos mais eficientes utilizando técnicas conhecidas em teoria da informação. O

desenvolvimento desses algoritmos já foi realizado e os resultados práticos demonstram os avanços previstos na teoria. A consolidação desses resultados é proposto como trabalho futuro.

Compressão de grafos web Abordamos o problema de comprimir grafos web e avançamos nesta área ao propormos uma nova representação compacta para grafos web, intitulada *árvore-w*, construída especificamente para memória externa (disco), sendo a primeira nesse gênero.

Para construí-la, organizamos a lista de adjacência do grafo web em uma árvore binária e a particionamos em blocos. Cada bloco é armazenada em uma página de disco. Apresentamos várias consultas que podem ser executadas e otimizadas. A *árvore-w* é a primeira representação à suportar a execução de consulta de leitura aleatória em lote e a otimização de consultas avançadas, inclusive em memória principal. Propomos também um novo tipo de layout projetado especificamente para grafos web, intitulado *layout escalado*. Além disso, mostramos como construir um layout *cache-oblivious* para explorar a hierarquia de memória, sendo a primeira representação desse tipo para grafos web.

Finalmente, executamos uma série de experimentos e comparamos com o *webgraph framework* (Frw10). Os resultados mostraram que a *árvore-w* apresenta taxas de compressão e de tempo de execução competitivas com outras representações compactas. Desta forma, demonstramos empiricamente a viabilidade da *árvore-w* para memória externa, contrariando a afirmação de vários pesquisadores (Sue01) (Bue08). Entre os layouts testados, verificamos que o melhor layout para a *árvore-w* é o escalado. Ressaltamos que isso também se verifica na teoria quando mostramos que a altura da *árvore-w* com layout escalado é assintoticamente menor em relação ao layout normal (Seção 4.2).

A maneira em que a *árvore-w* foi projetada teve a intenção de suportar operações de atualização. As operações de adição e remoção de nós do grafo web já foram concebidas. No entanto, a consolidação dessas operações é deixado como trabalho futuro. Finalmente, demonstrar a propriedade de compressibilidade da *árvore-w* (Chi09) é interessante e também é proposto como trabalho futuro.