

7

Conclusões

Neste trabalho, códigos de permutação vetorial são introduzidos para modulação e desenvolvidos para compressão de fontes.

No contexto de compressão, é demonstrado um teorema que assegura que o VPC pode atingir um desempenho idêntico ao do ECVQ quando se permite que seu comprimento n seja infinitamente grande. A prova do teorema está baseada na observação de que as palavras de um VPC correspondem às *seqüências típicas* produzidas por n aplicações de um ECVQ. A lei dos grandes números implica que, quando $n \rightarrow \infty$, qualquer seqüência produzida por n ECVQ's será típica com probabilidade 1, sendo portanto também uma palavra do VPC. Baseado na conexão entre VPC e ECVQ expressa por esse teorema, é proposto um algoritmo para o projeto completo de VPC. Fazendo-se o alfabeto do VPC igual ao dicionário do ECVQ, e escolhendo-se o vetor de composição do primeiro a partir das probabilidades dos índices do segundo, é possível obter VPC's de comprimento n com desempenhos que se aproximam do desempenho do ECVQ quando n é aumentado. Mesmo para um comprimento n finito, esse método é capaz de fornecer códigos com bom desempenho, um fato que pode ser verificado através dos resultados de simulação apresentados. Para as fontes uniforme e gaussiana, são exibidos códigos de permutação vetorial com desempenhos semelhantes ao do ECVQ e superiores aos do SPC e do ECSQ ótimos. No caso da fonte uniforme, uma etapa adicional de refinamento do alfabeto foi incluída no algoritmo. Um desenvolvimento analítico para este refinamento ótimo de alfabeto é também apresentado neste trabalho.

No contexto de modulação, códigos de permutação vetorial são introduzidos e desenvolvidos. Um método é descrito para a detecção ótima de VPM em canal AWGN. Este método corresponde ao mesmo método utilizado para a codificação ótima de VPC, o qual pode ser implementado pelo algoritmo CSA e tem eficiência moderada. Este método também pode ser aplicado na detecção ótima de VPM em canais com imperfeições, desde que

a matriz do canal H seja não-nula apenas em blocos de tamanho $L \times L$ dispostos ao longo da diagonal principal. Como o SPM é um caso particular de VPM, este método também pode ser utilizado para a detecção ótima de SPM's em canais com desvanecimento, permitindo, nos casos em que $K > 3$ e o desvanecimento é rápido, uma decodificação com eficiência maior que a dos métodos propostos até o momento (*cf.* Seção 6.2.2). O projeto ótimo de VPM é também abordado. Em particular, uma expressão é obtida para o vetor de composição ótimo, e mostra-se que esta corresponde aos valores arredondados de uma distribuição de Maxwell–Boltzmann. Os alfabetos de VPM utilizados neste trabalho são obtidos a partir dos reticulados mais densos conhecidos; a justificativa para essa abordagem baseia-se na teoria do empacotamento de esferas. Em seguida, tais alfabetos são submetidos ao projeto ótimo de vetor de composição. Os resultados obtidos mostram que, a partir de certos valores de comprimento n , VPM's com maior dimensão de alfabeto L são superiores em desempenho a VPM's que possuem menor L . Em particular, até mesmo o desempenho assintótico de um SPM ou VPM pode ser alcançado por outro VPM com maior dimensão de alfabeto e comprimento apenas moderado.

Em ambos os contextos, observa-se que as técnicas baseadas em códigos de permutação vetorial apresentam melhores desempenhos que suas correspondentes versões escalares. Este resultado paraleliza o fato conhecido na literatura de que descrições conjuntas (vetoriais, multidimensionais) são mais eficientes que descrições independentes (escalares, unidimensionais).

7.1

Sugestões para trabalhos futuros

A seguir são apresentadas algumas sugestões para futuras pesquisas nesta área.

Algoritmo simplificado para codificação ótima

Como descrito nos capítulos 4 e 6, a operação de ordenação de vetores (codificação ótima de VPC e detecção ótima de VPM) é uma operação bastante complexa do ponto de vista computacional, com complexidade $O(n^2\sqrt{n}\log n)$. Além ter impacto nas eficiências do projeto e da análise de desempenho, a complexidade desta operação pode representar um obstáculo para a utilização dos códigos de permutação vetorial em algumas aplicações práticas. Um resultado de grande interesse é, portanto, o desenvolvimento

de algoritmos para a codificação de VPC e a detecção de VPM (as quais são problemas aparentemente mais simples que a ordenação geral de vetores) com complexidade mais baixa, por exemplo, $O(n \log n)$. Caso não possam ser encontrados algoritmos ótimos para estes problemas, algoritmos sub-ótimos também são de interesse. Uma possível direção a ser investigada consiste em buscar analogias entre a codificação de VPC e a de ECVQ, e entre a decodificação de VPM e a de constelações multidimensionais não-equiprováveis.

VPC e VPM Variante II

Códigos de permutação escalar para compressão e para modulação são definidos em duas variantes: Variante I e Variante II. Esta segunda variante ainda se divide em Variante II-a e Variante II-b. Em ambos os contextos de aplicação, a Variante II geralmente apresenta melhores desempenhos que a Variante I para comprimentos n finitos. Portanto, é interessante que esta variante seja desenvolvida para VPC e VPM. Ressaltamos que, aparentemente, a Variante II-b pode ser diretamente estendida para o caso vetorial. A extensão da Variante II-a, no entanto, não aparenta ser trivial.

VPC com múltiplas palavras de referência

Em [27], Luzheng *et al.* propõem a utilização de códigos de permutação (escalar) com múltiplas palavras de referência. Nesse tipo de código, as palavras do dicionário correspondem a todas as possíveis permutações de cada uma das J palavras de referência disponíveis. Isto é equivalente a considerar um código de permutação *composto* constituído de J códigos de permutação individuais. Do ponto de vista da codificação, o problema é simples: basta codificar com os J códigos individuais e escolher dentre as J palavras de reprodução obtidas aquela que produz a menor distorção em relação à palavra da fonte. Nesse sentido, a extensão desta técnica para códigos de permutação vetorial pode ser facilmente estabelecida. O desafio se dá na etapa de projeto: deseja-se projetar conjuntamente os J códigos de permutação de forma que o código composto resultante apresente o melhor desempenho possível. Os resultados publicados para SPC são promissores e sugerem que a aplicação desta técnica em VPC também pode produzir bons resultados.

Mapeamento entre bits e permutações

Nos capítulos 5 e 6, assume-se que a confiabilidade de um sistema é dada inteiramente pela distância mínima entre as palavras do código. Numa implementação prática de VPM, no entanto, é preciso que as permutações estejam associadas aos bits da informação que se deseja transmitir. Caso não seja utilizada alguma codificação de canal específica para a modulação VPM, a confiabilidade será calculada diretamente como a probabilidade de erro dos bits transmitidos. Nesse caso, uma simulação do desempenho de um VPM pode ser realizada a partir de qualquer mapeamento entre bits e permutações escolhido; pode-se usar, por exemplo, a técnica de associação de permutações a números inteiros (e portanto a bits) descrita em [2]. É fácil ver, no entanto, que um mapeamento “mal escolhido” pode produzir uma probabilidade de erro de bit de valor comparável à probabilidade de erro de palavra $P(e)$, uma situação geralmente indesejável para uma modulação que emprega nLR bits por palavra como o VPM. É de grande interesse, portanto, encontrar um mapeamento adequado entre bits e permutações, isto é, um mapeamento que minimize a probabilidade de erro de bit dada uma escolha errônea da permutação.