

1

Introdução

Informalmente, um algoritmo pode ser descrito como uma sequência de passos que processam uma entrada e produzem uma saída como resposta [19]. Em computação, um algoritmo representa um procedimento computacional para resolução de uma classe de problemas. Tais problemas podem ser classificados em três diferentes grupos segundo a existência e a eficiência de algoritmos que resolvem estes problemas.

O primeiro grupo é composto pelos problemas que podem ser resolvidos de forma eficiente por um computador. O segundo deles abriga os problemas ditos intratáveis, ou seja, aqueles que podem ser resolvidos, mas não de forma eficiente. Finalmente, existem ainda problemas para os quais não se conhece um algoritmo que possa ser executado por um computador, mesmo que não se tenha limitação alguma de tempo. Estes são ditos não-computáveis.

Os problemas pertencentes ao primeiro grupo descrito acima podem ser resolvidos por algoritmos determinísticos cujo tempo de processamento é limitado por um polinômio no tamanho da entrada. Estes algoritmos têm complexidade polinomial e os problemas de decisão, aqueles cuja resposta final é sempre sim ou não, resolvidos por eles são conhecidos por pertencerem à classe P.

Os problemas de decisão ditos intratáveis são também conhecidos como NP-completos, o que significa dizer que para estes problemas existem algoritmos não-determinísticos polinomiais. Porém, no pior caso, um computador necessita de um tempo exponencial no tamanho da entrada para executar tais algoritmos. Apesar de nunca ter sido provada sua inexistência, nenhum algoritmo eficiente para estes problemas foi encontrado. Em geral, problemas dessa classe são menos beneficiados pelos avanços tecnológicos de *hardware*, pois como somente são conhecidos algoritmos de complexi-

dade exponencial, um grande aumento na velocidade das máquinas permite a resolução de problemas apenas um pouco maiores. Uma relação com diversos exemplos de problemas NP-completos pode ser encontrada em [32].

Uma abordagem interessante para a solução dos problemas NP-difíceis (versão de otimização de problemas NP-completos) é a utilização de algoritmos aproximados, que consistem em métodos não-exatos que buscam uma solução próxima da ótima e com tempo de processamento baixo. Uma forma de tratar tais problemas de maneira exata é através das técnicas de programação inteira, como por exemplo, algoritmos de *branch-and-bound* e *branch-and-cut*.

Problemas de reconhecimento estrutural de cenas são exemplos de problemas de otimização e consistem na caracterização dos objetos envolvidos em uma determinada cena, assim como das relações existentes entre eles. Uma aplicação para este problema na área de imagens médicas é a de efetuar-se o reconhecimento de estruturas 3D do cérebro humano, a partir de imagens obtidas por ressonância magnética. Tais imagens são previamente processadas por algum método de segmentação automática e o processo de reconhecimento consiste na busca da correspondência estrutural entre a imagem e um modelo genérico, tipicamente definido como um atlas de imagens médicas.

Além da aplicação na área de reconhecimento de imagens médicas, o problema de reconhecimento de cenas pode ser encontrado em diversas outras áreas como o reconhecimento facial, o reconhecimento de imagens aéreas e a identificação de imagens previamente conhecidas, tais como estradas e mapas de cidades, entre outras. Outros exemplos de aplicações nesta área podem ser encontrados em [13, 41].

Neste trabalho, assume-se que as imagens são processadas por um método de supersegmentação, ou seja, a cena gerada possui mais regiões que o modelo. Além das dificuldades naturais causadas pela supersegmentação, a presença em uma cena de diferentes objetos similares e as variações na forma que um objeto é visto sob diferentes ângulos de uma mesma cena tornam imprescindível uma descrição precisa das características relevantes da cena para o processo de reconhecimento. Além disso, este contexto torna muito difícil uma abordagem de mapeamento biunívoco entre imagem e modelo.

Em geral, o formalismo utilizado na representação de cenas é a estrutura de um grafo, no qual seus nós representam os objetos da cena e as

arestas as relações entre eles [14]. Neste caso, uma vez que a abordagem por isomorfismo de grafos é muito difícil, visto que as imagens sofreram uma supersegmentação, o problema em questão passa a ser o de correspondência inexata entre dois grafos distintos, um representando a imagem a ser reconhecida e o outro representando o modelo genérico daquela imagem.

Problemas de correspondência inexata de grafos podem ser definidos como a busca de uma associação entre vértices e arestas de dois grafos distintos para os casos nos quais uma abordagem por isomorfismo é forte demais, ou seja, quando não se pode supor uma associação biunívoca entre os dois grafos. Basicamente, o processo de correspondência deve rotular cada um dos vértices do grafo que representa a imagem a ser reconhecida, de forma a associá-lo a algum vértice do modelo, respeitando-se eventuais restrições, definidas normalmente a partir de características das imagens.

1.1

Objetivos do trabalho

Baseado na formulação apresentada em [14] para o problema de correspondência inexata de grafos (PCIG) aplicado ao reconhecimento de cenas, o objetivo principal deste trabalho foi desenvolver novos algoritmos aproximados e exatos, melhores que aqueles disponíveis na literatura. Dentre os novos algoritmos aproximados propostos encontram-se um algoritmo construtivo guloso aleatorizado e um procedimento de reconexão por caminhos, ambos utilizados em conjunto com uma heurística GRASP. Em relação à abordagem exata, desejava-se obter uma formulação linear inteira, uma vez que a formulação proposta por Boeres apresenta componentes não lineares.

1.2

Contribuições alcançadas

A principal contribuição alcançada por este trabalho foi a formulação por programação linear inteira do PCIG, que possibilitou o desenvolvimento de algoritmos exatos para problemas de pequeno e médio porte, além de um melhor conhecimento sobre a estrutura do problema de correspondência de grafos em reconhecimento de cenas.

Além disso, um novo algoritmo guloso aleatorizado foi proposto, com melhor desempenho do que as heurísticas já conhecidas.

1.3

Estrutura da dissertação

O Capítulo 2 apresenta uma breve revisão bibliográfica sobre o tema, além da descrição detalhada do problema e da formulação como um problema de otimização combinatória.

No Capítulo 3 é proposta uma formulação original do PCIG como um problema de programação linear inteira.

Com o objetivo de se aplicar um tratamento heurístico ao problema, o Capítulo 4 propõe um algoritmo construtivo guloso aleatorizado, um procedimento de reconexão por caminhos (*path relinking*), e um algoritmo GRASP que combina estes dois procedimentos com uma busca local proposta por Boeres em [14]. São analisadas as complexidades de pior caso dos algoritmos propostos e apresentados os respectivos pseudo-códigos.

O Capítulo 5 trata da discussão dos resultados das execuções dos algoritmos aproximados e da resolução do modelo de programação inteira por um resolvidor, realizando-se comparações destes resultados entre si e com os resultados apresentados em [14, 15].

O Capítulo 6 finaliza as discussões levantadas neste trabalho, apresenta as devidas conclusões e propõe possíveis extensões.