3
Programação Inteira e resolvedores MIP

Um problema de programação inteira é um problema que pode ser modelado da seguinte forma [29]:

$$
\begin{aligned}
\min \quad & c'x \\
\text{sujeito a:} & \\
& Ax = b \\
& x \geq 0 \\
& x \quad \text{inteiro}
\end{aligned}
\tag{3-1}
$$

Essa definição é semelhante à definição de um problema de programação linear, com exceção da restrição de integralidade nas variáveis x. Um problema que misture variáveis inteiras e variáveis lineares é chamado de problema de programação inteira mista (*mixed integer programming*, MIP). Apesar da simplicidade do modelo 3-1, programação inteira é uma ferramenta poderosa, sendo capaz de modelar a maioria dos problemas de otimização combinatória mais famosos, como o Problema do Caixeiro Viajante (PCV), problemas de alocação (*scheduling*), coloração de grafos, entre outros. Embora existam subclasses de problemas MIP fáceis de resolver, como os problemas que possuem matrizes de restrições totalmente unimodulares e que podem ser resolvidos em tempo polinomial, o problema geral de programação inteira mista é NP-difícil. A grande atração de um modelo único para vários problemas é que, uma vez que os problemas foram colocados no modelo, as mesmas técnicas podem ser usadas para todos. Desta maneira, as técnicas são abrangentes e as melhorias nas técnicas atingem a todos os problemas.

3.1
Resolvendo MIPs

A mais conhecida técnica para resolver problemas MIP é o *Branch and Bound* (B&B), parte da família dos algoritmos enumerativos, assim chamados porque enumeram de maneira inteligente todas as possíveis soluções. De maneira simplificada, o B&B forma sua árvore de busca restringindo o valor das variáveis inteiras - por exemplo, uma variável binária vai gerar duas ramificações na árvore de busca, uma onde ela recebe o valor zero e outra onde ela recebe o valor um. A cada ponto na enumeração, pode-se encontrar um limite inferior resolvendo a versão relaxada do problema, ou seja, o problema linear que possui as mesmas variáveis e restrições, com excessão da restrição de integralidade. No momento em que é encontrada uma solução da versão relaxada onde as variáveis têm valor inteiro, encontrou-se a solução ótima daquela sub-árvore e uma solução viável do problema.

Outra família de algoritmos bastante utilizados para resolver problemas MIP são os algoritmos de *cutting plane*. Nestes algoritmos, primeiro resolve-se a relaxação linear do problema. Se a solução for inteira, ela é também a solução ótima do problema. Caso contrário, é garantido que exista uma inequação linear (um plano) que separa a solução encontrada do fecho convexo do conjunto de soluções viáveis. A inequação é chamada de corte, enquanto o problema de encontrá-la é chamado de problema de separação. O processo pode então ser repetido até se encontrar uma solução ótima (ou seja, uma solução cujas variáveis são inteiras). Este algoritmo foi inicialmente proposto por Gomory [17], que também provou que ele converge, em tempo finito, à solução inteira ótima. Finalmente, um algoritmo *Branch and Cut* mistura a enumeração de um B&B com geração de *cutting planes*, gerando planos de corte a cada nó da enumeração.

Devido à sua grande aplicabilidade, resolvedores MIP vêm sido desenvolvidos nas últimas décadas com grande sucesso. Evoluindo de simples algoritmos Simplex, os resolvedores MIP são, atualmente, ferramentas bastante sofisticadas, desenvolvidas para prover, em um tempo de computação aceitável, a solução ótima do modelo ou ao menos uma solução heurística com uma distância aceitável do ótimo [12].

3.1.1
Resolvedores comerciais

Devido à característica generalizadora da programação inteira, um resolvedor MIP pode ser usado para resolver diversos problemas diferentes, tornando-os atraentes do ponto de vista comercial. Entre os vários resolvedores comerciais existentes, sem dúvida o mais famoso é o CPLEX [1], desenvolvido pela empresa ILOG, parte da IBM. O CPLEX é distribuído junto com vários pacotes comerciais de otimização e sua performance vem melhorando consideravelmente a cada versão [10]. Neste trabalho, o CPLEX versão 11 será utilizado como resolvedor MIP.

3.2 Dificuldades

Apesar dos grandes avanços nos últimos anos, mesmo os melhores resolvedores MIP comerciais ainda não conseguem resolver certos problemas mais complexos e, mesmo para problemas mais simples, instâncias de tamanho considerável. Nestes casos, além de não conseguirem resolver o problema em um tempo aceitável, os resolvedores normalmente utilizam uma quantidade excessiva de memória (muitas vezes terminando prematuramente a busca devido ao esgotamento de memória).

Outra dificuldade é que, apesar de serem imaginados como algoritmos "caixas-pretas", os resolvedores MIP modernos são produtos extremamente complexos. Existem dezenas de opções que podem ser escolhidas para gerar soluções melhores ou mais rapidamente, outras dezenas se você desejar que o resolvedor se comporte de alguma maneira específica. Embora as configurações originais sejam razoáveis, conseguir excelentes resultados em vários tipos de instâncias diferentes exige bastante configuração.

Finalmente, uma das grandes desvantagens dos resolvedores comerciais como o CPLEX vem do fato de que eles não são código aberto. Dessa maneira, os usuários ficam sujeitos a somente utilizar o resolvedor das maneiras que a fabricante especifica. Esse problema é mitigado até certo ponto por *frameworks open source*, nada mais do que *wrappers* em volta dos resolvedores, que fornecem uma interface mais limpa e simplificam o desenvolvimento de algoritmos que utilizam estes resolvedores.