1
Introdução

Nesta tese de doutorado, consideramos dois problemas de otimização combinatória bastante distintos: o problema de transporte em redes de dutos (PTD) e o problema de busca com custos de acesso variados (PBC).

O primeiro problema utiliza um modelo matemático baseado nas operações da Transpetro – a empresa responsável pelo transporte de produtos de petróleo através dos oleodutos brasileiros. Apesar do transporte por oleodutos ter características tanto discretas quanto contínuas, nosso modelo utiliza várias hipóteses simplificadoras que permitem descrever este transporte como um problema de otimização combinatória. Com isto, mantemos apenas as características que consideramos essencias. Vale mencionar que estas simplificações permitiram determinar a complexidade de diversas variações do PTD. O estudo deste problema é motivado pelo projeto RSB[1], que é produto de uma parceria entre o Laboratório LEARN da PUC-Rio e o CENPES/Petrobras.

O segundo problema considera a busca por um elemento em um vetor ordenado e armazenado em meios físicos heterogêneos. Observe que, se todos os custos e probabilidades de acesso são iguais, então a melhor estratégia de busca é a busca binária, onde o próximo elemento acessado sempre divide o vetor corrente em duas partes (quase) iguais. No PBC, desejamos obter estratégias de busca que minimizam o custo esperado ou o custo do pior caso num vetor onde o custo de cada acesso pode variar em função da posição do elemento acessado. O estudo deste problema é motivado por aplicações como mecanismos de busca na Web e ferramentas para mapeamento de seqüências de DNA, que manipulam enormes bases de dados. Além disso, este problema está relacionado com projeto AMYRI, que é produto de uma cooperação entre pesquisadores íbero-americanos na área de Recuperação de Informações.

[1]Financiado pela FINEP/CTPetro

1.1 Transporte em Redes de Dutos

Oleodutos são a forma mais segura e barata de transportar petróleo e seus derivados por longas distâncias. Este meio de transporte é diferente de todos os demais por permanecer parado enquanto a carga se move. Outra característica importante é que os oleodutos estão sempre cheios. Por isso, assumindo que os fluidos são incompressíveis, quando uma certa quantidade de produto é inserida em uma extremidade, a mesma quantidade deve ser retirada da extremidade oposta. Além disso, um duto tem tipicamente menos do que um metro de diâmetro e dezenas de quilômetros de comprimento. Como resultado disso, grandes quantidades de fluidos distintos podem ser transportadas através do mesmo oleoduto com uma taxa de mistura relativamente baixa. Após o transporte, a porção misturada é diluída em um dos produtos sem que ele perca suas características.

A otimização do transporte por oleodutos é uma problema de grande relevância, pois tem um impacto significativo no preço dos derivados de petróleo. Algumas dissertações de mestrado [21, 24, 31, 34] já trazem métodos propostos para este problema, utilizando redes específicas da Petrobras para estudo de caso. Todos estes métodos são baseados em heurísticas e formulações por programação inteira mista. Poucos trabalhos encontrados na literatura apresentam resultados teóricos sobre o problema de transporte em redes de dutos [16, 22]. A primeira parte desta tese de doutorado considera um modelo simplificado para o problema de transporte por oleodutos. Acreditamos que os resultados teóricos obtidos com este modelo proporcionam um maior entendimento da estrutura combinatória inerente ao problema.

O PTD é definido da seguinte forma. Um duto é um tipo abstrato de dados que respeita a regra FIFO (*first in first out*) e cujo número de elementos deve se manter fixo. Chamamos estes elementos de bateladas. Como conseqüência do número fixo de bateladas, sempre que uma batelada é inserida (*push*) no final de um duto, a batelada do início deve ser removida (*pop*). Desta forma, as operações *push* e *pop* são simultâneas, resultando numa operação híbrida que denominamos *push-pop* (PP). Uma rede de dutos é definida por um grafo orientado $G = (N, A)$ onde cada arco $a \in A$ tem um duto associado com $v(a)$ posições, cada posição contendo uma batelada. No PTD, também é dado um conjunto L de bateladas. Cada batelada b tem um nó de destino $d_b \in N$ e uma posição inicial. Esta posição inicial pode ser um nó ou uma posição de um duto da rede. Assumimos que os nós da rede têm capacidade ilimitada. Para cada arco $a = (i, j)$, uma operação PP consiste em inserir no duto uma batelada contida no nó i e guardar no nó j a batelada removida deste duto. Observe que esta operação não modifica o número de bateladas contidas no duto. Este número permanece sempre constante. Por isto, é dado um subconjunto $F \subset L$ de bateladas *proteláveis*, que não precisam chegar aos seus destinos. Estas bateladas podem ser utilizadas para preencher os dutos de rede enquanto as outras são entregues. O objetivo do problema é encontrar uma seqüência de operações que transporte todas as bateladas não-proteláveis aos seus respectivos nós de destino.

Em [16, 22], os autores consideram redes de dutos cujos grafos associados são caminhos simples. Neste caso, os dutos são representados de forma semelhante à descrição anterior. No entanto, os autores assumem que as requisições de transporte e a movimentação das bateladas nos dutos são cíclicas. Esta hipótese não é utilizada no PTD, pois ela não se aplica aos casos práticos que motivaram o estudo deste modelo.

1.1.1 Resultados

Nesta tese de doutorado, apresentamos vários resultados em relação ao PTD [27, 28, 35, 36]. A própria definição do PTD e de suas variações é original. Além disso, demonstramos que encontrar uma solução viável para o PTD é um problema $\mathcal{NP}$-difícil, mesmo que o grafo G seja acíclico. Por isso, consideramos o PTD síncrono (PTDS), uma variação do PTD onde nenhuma batelada protelável está inicialmente contida em um duto. Assumindo que cada batelada b tem um peso $w(b)$ associado, o custo de uma operação PP em um duto é modelado como uma combinação linear dos pesos das bateladas envolvidas. Neste caso, o PTDS de custo mínimo (PTDSC) consiste em encontrar uma solução de custo mínimo para o PTDS, onde o custo de uma solução é dado pela soma dos custos de suas operações. Para este problema, apresentamos o algoritmo BPA (*Batch-to-Pipe Assignment*), um algoritmo polinomial que obtém uma solução viável para o PTDSC. Caso o grafo G seja acíclico, a solução dada pelo algoritmo BPA é ótima. Além disso, o algoritmo BPA é estendido para o caso em que bateladas com mesma origem, destino e peso podem ser agrupadas em ordens. Neste caso, o tempo de execução do algoritmo permanece polinomial em relação a uma entrada representada de forma mais compacta. Vale mencionar que o PTDSC assume que as bateladas não têm prazos de entrega.

Também consideramos o PTDS de *makespan* mínimo (PTDSM), que consiste em obter uma solução para o PTDS que minimiza o tempo necessário para terminar a última operação PP. Neste caso, assumimos que cada duto pode executar uma operação PP a cada unidade de tempo. Demonstramos que o PTDSM não admite nenhum algoritmo polinomial $\eta^{1-\epsilon}$-aproximado para nenhum $\epsilon > 0$, a menos que $\mathcal{P} = \mathcal{NP}$, onde η é o tamanho da instância. Este resultado ainda vale se o grafo G é acíclico e planar. Além disso, mostramos que o algoritmo BPA fornece uma solução $|A|$-aproximada para o caso em que o grafo G é acíclico. Apesar deste fator de aproximação ter um crescimento assintótico elevado, o nosso resultado anterior mostra que qualquer fator de aproximação que não apresente um crescimento semelhante implica em $\mathcal{P} = \mathcal{NP}$.

1.2 Busca com Custos de Acesso Variados

Seja $\mathcal{A} = [a_1, \ldots, a_n]$ um vetor ordenado onde $c(a_i)$ é o custo de acessar o seu i-ésimo elemento. O objetivo do problema é encontrar uma estratégia de busca de custo mínimo para o vetor $\mathcal{A}$. Se todos os custos de acesso são iguais, então a melhor estratégia de busca possível é a busca binária, que permite encontrar qualquer elemento após $O(\log n)$ acessos. Por outro lado, se os custos de acesso são variados, então a busca binária não é necessariamente uma estratégia ótima. Neste caso, chamamos este problema de PBC.

Consideramos duas formas de calcular o custo de uma estratégia de busca para o PBC: pelo custo médio e pelo pior custo de uma busca. Chamamos os respectivos problemas de problema da busca de custo médio (PBCM) e de problema da busca de pior custo (PBPC). No caso do PBCM, assumimos que todos os elementos de $\mathcal{A}$ têm a mesma probabilidade de acesso. Em ambos os casos, o melhor algoritmo exato conhecido utiliza programação dinâmica, executando em tempo $O(n^3)$ e utilizando um espaço de memória $O(n^2)$ [11, 29]. No caso do PBCM [11], o desenvolvimento do algoritmo é motivado por um problema de projeto de filtros. O autor também considera o caso particular onde a chave a_i tem custo proporcional a i^k, para um dado valor inteiro positivo de k. Neste caso, ele prova que o custo de uma busca binária está a um fator constante da estratégia ótima.

1.2.1 Trabalhos Relacionados

Outras variações do problema de busca também são encontradas na literatura. Por exemplo, um grande número de trabalhos aborda o caso onde as chaves têm custos de acesso iguais e probabilidades de acesso variadas [2, 4, 10, 17]. Neste caso, o melhor algoritmo exato conhecido foi proposto por Knuth. Também baseado em programação dinâmica, este algoritmo executa em tempo e espaço $O(n^2)$.

Em [29], Navarro et. al. consideram uma generalização do PBC, onde o custo de acesso de uma chave depende do acesso anterior. Este estudo é motivado por aplicações que fazem busca em textos indexados por arranjos de sufixo, onde deseja-se minimizar o tempo gasto pela cabeça do disco magnético durante a busca. Neste caso, se a cabeça do disco está na trilha i, então o tempo necessário para ler um dado na trilha j é dado por $c(i, j)$. Os autores utilizam programação dinâmica em dois algoritmos exatos: um para minimizar o custo esperado de uma busca e outro para minimizar o custo da busca mais cara. Ambos os algoritmos rodam em tempo $O(n^3)$ e utilizam uma espaço $O(n^2)$. Observe que o PBCM e o PBPC são casos particulares do primeiro e do segundo problema, respectivamente.

Em [12], Wachs considera o PBC com uma estrutura de custos particular observada em uma aplicação de busca em uma fita. Neste caso, o autor estende as desigualdades quadrandulares de F. F. Yao [8] para obter um algoritmo exato que roda em tempo $O(n^2)$.

Em [37], Charikar et. al. consideram a seguinte função de custo para o PBC: a maior razão entre o custo de uma busca e o custo de acessar apenas o elemento procurado. Neste caso, a motivação dos autores é apresentar uma análise competitiva [23] de algoritmos determinísticos que buscam uma informação em um vetor ordenado com custos sem saber a sua posição contra um adversário adaptativo, ou seja, que escolhe a posição da informação desejada. Sob esta ótica, a razão competitiva de uma estratégia de busca é exatamente o custo descrito anteriormente. Para este problema, os autores propõem um algoritmo eficiente que sempre calcula uma estratégia cuja razão competitiva é menor ou igual a $\log_2 n + O(\sqrt{\log n} \log\log n)$. Vale mencionar que, no caso de custos uniformes, não é possível obter uma razão competitiva melhor do que $\log_2 n$. Além disso, eles apresentam um algoritmo exato pseudo-polinomial baseado em programação dinâmica para o problema.

1.2.2 Resultados

Nesta tese de doutorado, apresentamos algoritmos aproximados para o PBCM e para o PBPC [25, 38, 32, 39].

Para o PBCM, apresentamos o Algoritmo da Razão. Este algoritmo sempre obtém uma solução de custo menor ou igual a $4\ln(n+1)/n$, assumindo que a soma dos custos de todas as chaves de $\mathcal{A}$ é 1. Vale mencionar que esta solução é sempre $4\ln(n+1)$-aproximada. No pior caso, este algoritmo roda em tempo $O(n^2)$ e utiliza um espaço linear. No entanto, este tempo de execução pode ser $O(n \log n)$, dependendo da estratégia de busca obtida. Também apresentamos alguns experimentos comparativos para duas estruturas de custo específicas: uma foi proposta em [11] e a outra é aleatória. Estes experimentos sugerem que o Algoritmo da Razão é o melhor algoritmo não exato para estas estruturas de custo. Nestes experimentos, o Algoritmo da Razão obteve soluções cujos custos apresentaram erros abaixo de 3% em relação aos respectivos custos ótimos.

Além disso, introduzimos uma nova abordagem para desenvolver algoritmos aproximados para problemas de busca com custos de acesso variados. Esta abordagem utiliza uma técnica conhecida como escala de custos, que também é usada em problemas de fluxo em redes [18] e em outras variações do PBC [37]. No entanto, a característica mais interessante desta abordagem é a demonstração de limites inferiores para o custo a partir de um esquema de decomposição do problema. Com base nesta abordagem, desenvolvemos os algoritmos de Escala de Custos para Busca Média (ECBM), para o PBCM, e de Escala de Custos para a Pior Busca (ECPB), para o PBPC. Ambos constroem soluções $(2+\epsilon+o(1))$-aproximadas, para qualquer $\epsilon > 0$, em tempo linear e utilizando um espaço linear. Em ambos os casos, não utilizamos nenhuma hipótese sobre a estrutura dos custos da entrada.

1.3 Organização

Este documento está organizado da seguinte forma. No capítulo 2, apresentamos os resultados relacionados ao PTD. As seções 2.1, 2.2 e 2.3 apresentam respectivamente uma formalização do PTD, uma demonstração de que o PTD é $\mathcal{NP}$-difícil e uma descrição da hipótese de sincronismo, que caracteriza o PTDS. Em seguida, na seção 2.4, apresentamos e analisamos o algoritmo BPA e a sua versão estendida. Além disso, na seção 2.5, provamos os resultados de inaproximabilidade do PTDSM. No capítulo

3, descrevemos e analisamos algoritmos aproximados para o PBCM e para PBPC. As seções 3.1 e 3.2 apresentam respectivamente notações e definições utilizadas ao longo do capítulo e propriedades da busca binária, para o caso em que os custos de acesso são uniformes. Em seguida, na seção 3.3, descrevemos do Algoritmo da Razão e apresentamos uma análise de aproximação para este algoritmo. Além disso, nas seções 3.4, 3.5 e 3.6, apresentamos respectivamente uma descrição dos algoritmos ECBM e ECPB, as respectivas análises de aproximação e uma implementação para estes algoritmos. Finalmente, no capítulo 4, apresentamos nossas conclusões e trabalhos futuros.