

## Trabalho Prático 2

Como exemplo de problema de difícil solução computacional foi mostrado o problema da laminação de barras de ferro, que é um processo iterativo pelo qual cada barra inicialmente tem que passar por dois cilindros com vão ajustável, para diminuir a sua espessura. Então, o problema é encontrar a sequência de reduções necessárias para atingir um custo mínimo no final do processo. Como primeiro caso foi fornecida a tabela 1.

**Tabela 1 – Custo de redução de espessuras para várias espessuras de entrada**

Espessura de entrada	Redução de espessura		
	1mm	2mm	3mm
10mm	4	6	8
9mm	4	6	9
8mm	4	7	12
7mm	4	10	16
6mm	5	12	x
5mm	7	x	x

Então, para resolver esse problema foram fornecidos quatro arquivos, cada um com uma tabela de redução de espessuras. Sendo que para o primeiro arquivo, a peça de metal começa com uma espessura de 10mm, e para os arquivos seguintes, as barras começam respectivamente com 15, 20 e 25. Para essa solução foi pedida a implementação de três métodos diferentes: algoritmo guloso, programação dinâmica e backtracking.

O algoritmo guloso foi implementado com a melhoria de um custo/benefício, priorizando o menor custo e o maior benefício (maior redução por iteração). Já na programação dinâmica, a cada célula da tabela havia a comparação entre manter o valor da linha de cima ou alterar, retirando a quantidade de itens necessários para adicionar um item da linha corrente. E o Backtracking foi feito considerando a entrada com ordem crescente de redução das espessuras para cada nó e também com ordem decrescente. Então, foram encontrados os resultados da tabela 2.

**Tabela 2. Resultados encontrados**

Espessura de entrada	Alg Guloso		Programação Dinâmica		Backtracking (ascendente)		Backtracking (decrescente)	
	Valor encontrado	Tempo (ms)	Valor encontrado	Tempo (ms)	Valor encontrado	Tempo (ms)	Valor encontrado	Tempo (ms)
10	24	7	24	1	24	4	24	1
15	51	6	55	1	51	7	51	2
20	58	6	65	1	57	20	57	6
25	95	8	95	2	95	87	95	55

A coluna que chama mais a atenção na tabela é a coluna da programação dinâmica que sempre tem o melhor tempo, e nem sempre tem o resultado certo, o que mostra um erro na implementação, já que o esperado era ter um tempo mais alto e sempre o resultado certo. Analisando o resultado para o tamanho de espessura de entrada 15mm, foi visto o erro de que ela não está considerando todas as possibilidades a cada célula, mas só a linha de cima e a célula anterior, quando o resultado mais interessante passa a ser uma combinação da redução de 3mm com reduções de 1mm (e não de 2mm), ela apresenta esse erro. Mas mesmo assim conseguiu ter o resultado certo para duas espessuras de entrada com o menor tempo. Porém, não houve tempo hábil para corrigir o problema.

Outro fato interessante é que o backtracking com uma ordenação decrescente conseguiu tempos bem interessantes para quase todas as espessuras de entrada, isso deve-se ao fato de reduções de 3mm serem frequentemente interessantes. Mas quando as espessuras de entrada, consequentemente a quantidade de possibilidades, começam a aumentar, o tempo também aumenta consideravelmente, como já aponta a teoria sobre o assunto. O algoritmo guloso agiu de acordo como esperado tendo respostas rápidas, mas nem sempre precisas.