

AQUI começa o trabalho colaborativo – TALVEZ tenha muitas subsection

1 Cabo de Guerra

1.1 Descrição

Melhorar o enunciado ... mas o outros estão melhores

Várias crianças se encontram para brincadeira do cabo-de-guerra no pátio da escola. Como será feita a divisão entre as duas equipes? “*Por peso*” grita o mais eufórico. Que seja feita a divisão dos times sobre uma sequência/lista de peso tal como:

$Joao_1$	$Pedro_2$	$Manoel_3$	$Zeca_n$
45	39	79	42

- ➡ Preencha a tabela acima com inteiros e valores de sua família.
- ➡ Sim, por peso, todos concordaram, “*exceto que a divisão deveria respeitar o critério $|N_A - N_B| \leq 1$* ”, disse o mais cauteloso. Sim, nenhum time poderia duas crianças a mais que ou outro time.

1.2 Especificação

Que seja feita a divisão:

$Joao_1$	$Pedro_2$	$Manoel_3$	$Zeca_n$
45	39	79	42

- Divisão por peso
- Respeitar critérios como: $|N_A - N_B| \leq 1$
- Todos devem brincar
- Bem, esta simples restrição ($|N_A - N_B| \leq 1$), de nosso cotidiano tornou um simples problema em mais uma questão combinatória. Um arranjo da ordem de $\frac{n!}{(n/2)!}$. Casualmente, nada trivial para grandes valores!

➡ Dois detalhes:

1. Na tabela de pesos, use valores **inteiros**;
2. Use os pesos de seus familiares para completar esta tabela com um quantidade significativa;
3. No lugar de *array* como estrutura base, use *sets* para armazenar e manipular estes valores. Com certeza ficará mais *elegante*, e possivelmente mais ineficiente. Teste e comprove!

1.3 Modelagem

- Usando uma variável de decisão: análogo a árvore do SAT

Nomes (n_i):	n_1	n_2	n_3	n_n
Peso (p_i):	45	39	79	42
Binária (x_i):	0/1	0/1	0/1	0/1

- Assim $N_A \approx N/2$, $N_B \approx N/2$ e $|N_A - N_B| \leq 1$
- $x_i = 0$: n_i fica para o time A
- $x_i = 1$: n_i fica para o time B
- Logo a soma:

$$\sum_{i=1}^n x_i p_i$$

é o peso total do time B (P_B)

- Falta encontrar peso total do time A (P_A), dado por:
- $P_A = P_{total} - P_B$
- ou

$$P_A = \sum_{i=1}^n p_i - \sum_{i=1}^n x_i p_i$$

- Finalmente, aplicar uma minimização na diferença: $|P_A - P_B|$

1.4 Estratégia

Uma árvore de decisão binária descreva como voce implementou ou a fundamentacao

1.5 Implementação

O código fonte se encontra em:

https://github.com/claudiosa/minizinc/blob/master/CEAVI_2015/cabo_de_guerra.mzn

1.6 Resultados e Análise

Considerando pesos aleatórios de 1 a 150 para as pessoas

Usando um *solver* médio do Minizinc (*G12 lazyfd*) padrão:

Referência: cpu 4-core, 4 G ram, SO: Linux-Debian

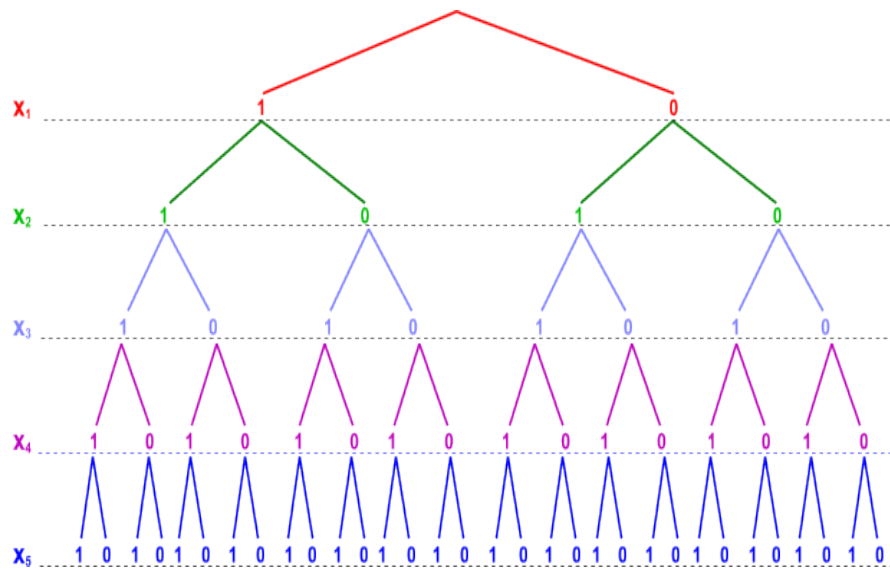


Figura 1: Se $x_i = 0$, então n_i segue para o time A , caso $x_i = 1$, então n_i vai para o time B

Tabela 1: Resultados

n	tempo	P_A	P_B
5	40msec	276	278
10	46msec	518	519
25	98msec	1198	1197
50	411msec	2290	2291
75	2s 485msec	3133	3133
100	470msec	4142	4142
125	7s 2msec	4992	4992
150	605msec	5823	5823
175	642msec	6777	6778
200	> 10min	—	—