

Um algoritmo exato para o problema de corte a 3 dimensões

Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study

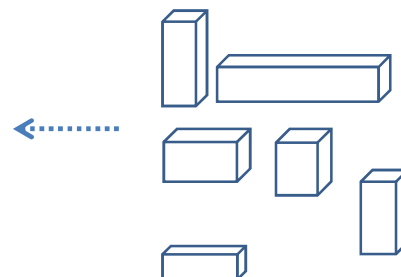
Mhand Hifi

João Rebelo



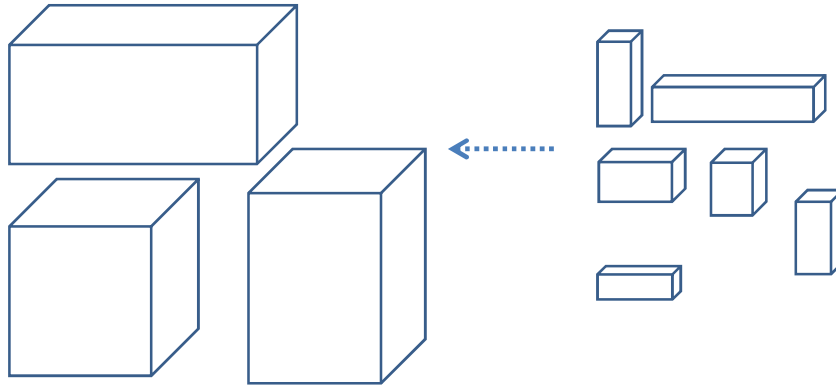
Introdução

1 palete - (U_3DC)



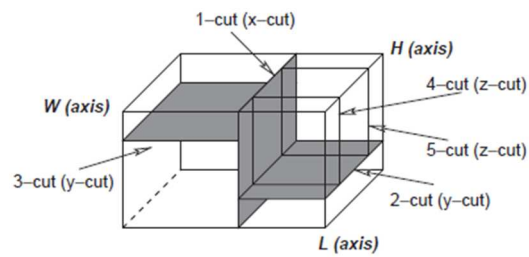
Introdução

Diferentes paletes - (U_G3DC)



Introdução

- Comprimento - l_i
- Largura - w_i
- Altura - h_i
- Lucro - c_i

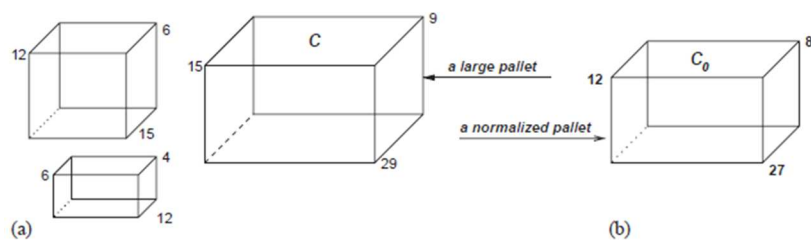


Introdução

O problema pode ser:

- *com restrições* ou *sem restrições* ($b_i, i = 1, \dots, n$)
- *fixo* ou *'rodado'*
- *pesado* ou *não pesado* ($c_i = l_i \cdot w_i \cdot h_i$)

Normalização



Peças - (15,12,6) e (12,6,4)
 Paleta - (29,15,9)

$X_{(29,15,9)} = \{12,15,24,27\}$
 $Y_{(29,15,9)} = \{6,12\}$
 $Z_{(29,15,9)} = \{4,6,8\}$

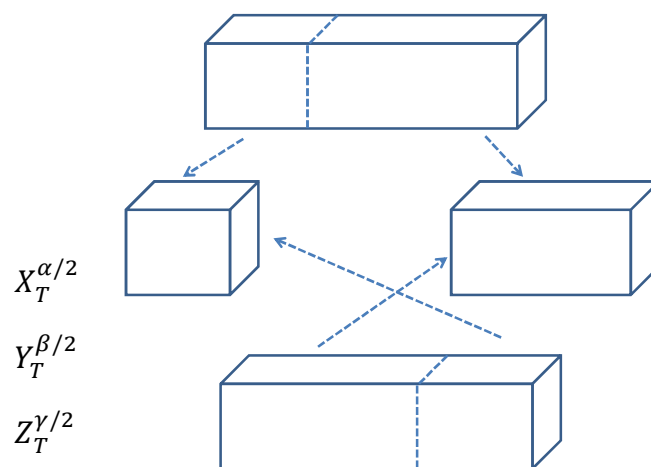
$\alpha_0 = \max\{X\}$
 $\beta_0 = \max\{Y\}$
 $\gamma_0 = \max\{Z\}$

$(\alpha_0, \beta_0, \gamma_0) = (27, 12, 8)$

Conjuntos Normalizados

- $X_{(\alpha,\beta,\gamma)} = \{x | x = \sum_{i=1}^n l_i t_i \leq \alpha, t_i \text{ é um inteiro não negativo}, (w_i, h_i) \leq (\beta, \gamma)\}$
- $Y_{(\alpha,\beta,\gamma)} = \{y | y = \sum_{i=1}^n w_i t_i \leq \beta, t_i \text{ é um inteiro não negativo}, (l_i, h_i) \leq (\alpha, \beta)\}$
- $Z_{(\alpha,\beta,\gamma)} = \{z | z = \sum_{i=1}^n h_i t_i \leq \gamma, t_i \text{ é um inteiro não negativo}, (l_i, w_i) \leq (\alpha, \beta)\}$

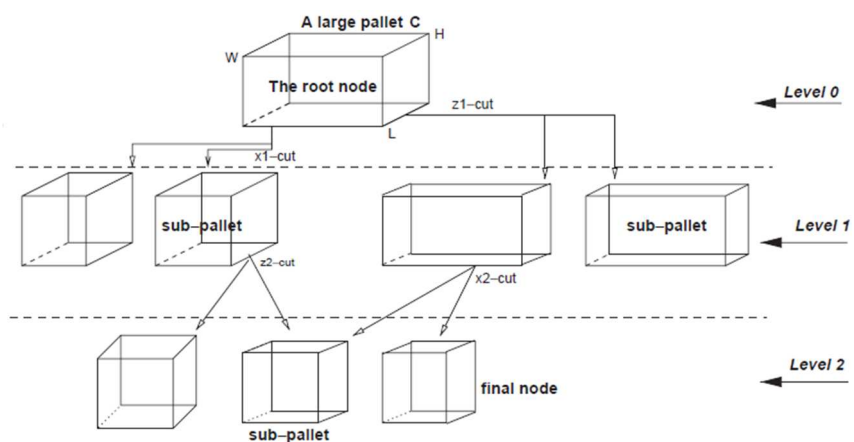
Simetria



Algoritmo SGT

- *Search Graph Technique*
- O algoritmo resolve o problema U_3DC usando uma estratégia de procura em árvore.
- Gera todos os padrões de corte possíveis criando uma árvore onde as ramificações representam cortes.
- Os nós representam estados das (sub-) paletes.

Algoritmo SGT



Limite Inferior

- O limite inferior para a sub-paleta (α, β, γ) actual é obtido enchendo a sub-paleta normalizada $(\alpha_0, \beta_0, \gamma_0)$ com a k -ésima peça que produz o valor da melhor solução homogénea.

$$k = \arg \max_{1 \leq i \leq n} \left\{ \left\lfloor \frac{\alpha_0}{l_i} \right\rfloor \left\lfloor \frac{\beta_0}{w_i} \right\rfloor \left\lfloor \frac{\gamma_0}{h_i} \right\rfloor \times c_i \text{ tal que } (l_i, w_i, h_i) \leq (\alpha_0, \beta_0, \gamma_0) \right\}$$

$\lfloor x \rfloor$ indica o maior inteiro não superior a x

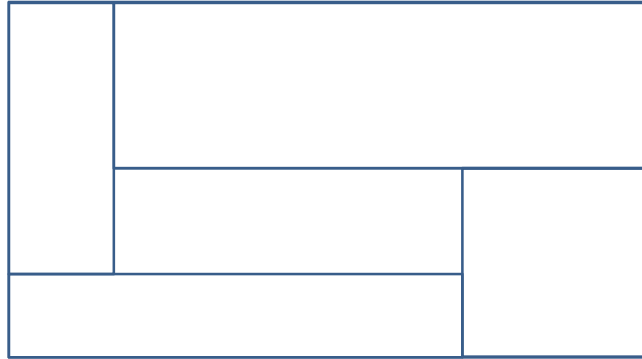
$$c_k = l_k \cdot w_k \cdot h_k \text{ para } k = 1, \dots, n$$

Limite Superior

$$\left\{ \begin{array}{l} U_{\alpha_0, \beta_0, \gamma_0} = \max \sum_{i \in S_{(\alpha_0, \beta_0, \gamma_0)}} c_i \cdot x_i \\ \text{sujeito a } \sum_{i \in S_{(\alpha_0, \beta_0, \gamma_0)}} (l_i \cdot w_i \cdot h_i) \cdot x_i \leq \alpha_0 \cdot \beta_0 \cdot \gamma_0 \\ x_i \leq \left\lfloor \frac{\alpha_0}{l_i} \right\rfloor \cdot \left\lfloor \frac{\beta_0}{w_i} \right\rfloor \cdot \left\lfloor \frac{\gamma_0}{h_i} \right\rfloor \\ x_i \text{ é um inteiro não negativo, } i = 1, \dots, n \end{array} \right.$$

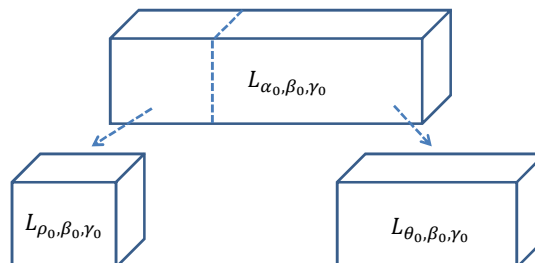
- Onde $S_{(\alpha_0, \beta_0, \gamma_0)}$ é o conjunto do tipo de peças a colocar na sub-paleta normalizada $(\alpha_0, \beta_0, \gamma_0)$
- x_i indica o número de vezes que o i -ésimo tipo de peça é colocada em $(\alpha_0, \beta_0, \gamma_0)$

Limite Superior



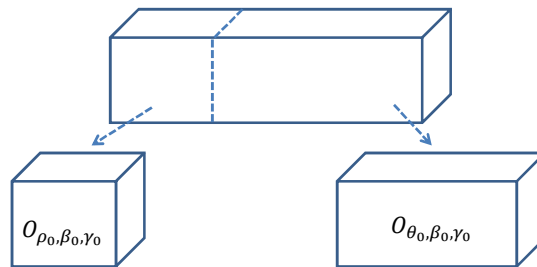
Estratégia 1

- O Limite inferior $L_{\alpha_0, \beta_0, \gamma_0}$ para uma dada sub-paleta é actualizado se e só se a soma de $L_{\rho_0, \beta_0, \gamma_0} + L_{\theta_0, \beta_0, \gamma_0}$ é maior do que o valor do limite inferior actual $L_{\alpha_0, \beta_0, \gamma_0}$.



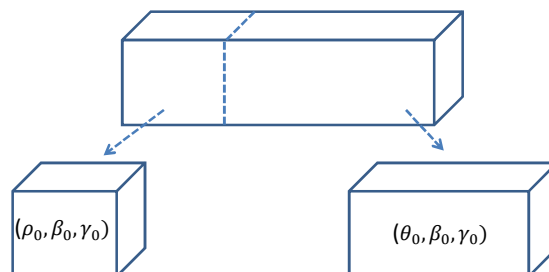
Estratégia 2

- Se os valores da solução ótima para as próximas duas sub-paletes já foram calculados, então paramos a ramificação. O limite inferior deste nó interno é substituído por $\max\{L_{\alpha_0, \beta_0, \gamma_0}, O_{\rho_0, \beta_0, \gamma_0} + O_{\alpha_0, \beta_0, \theta_0}\}$.



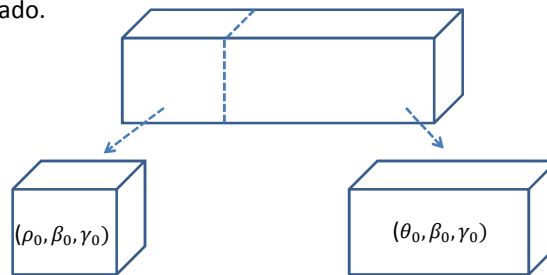
Estratégia 3

- Se $B - \min\{U_{\rho_0, \beta_0, \gamma_0}, O_{\rho_0, \beta_0, \gamma_0}\} \geq U_{\theta_0, \beta_0, \gamma_0}$, então não é necessário investigar para a sub-paleta $(\theta_0, \beta_0, \gamma_0)$, pois não é possível melhorar a solução na direcção deste nó.



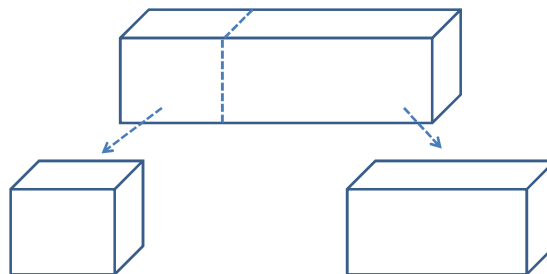
Estratégia 4

- Se $\min\{U_{\rho_0, \beta_0, \gamma_0}, O_{\rho_0, \beta_0, \gamma_0}\} + \min\{U_{\theta_0, \beta_0, \gamma_0}, O_{\theta_0, \beta_0, \gamma_0}\} \leq L_{\alpha_0, \beta_0, \gamma_0}$, então o corte em x pode ser negligenciado sem perda de otimalidade. Neste caso, o valor da melhor solução de $(\alpha_0, \beta_0, \rho_0)$ e $(\alpha_0, \beta_0, \theta_0)$ não pode ser melhorado.



Estratégia 5

- A seguir a cada corte, examinamos o nó que realiza o $\max\{L_{\alpha_0, \beta_0, \rho_0} / U_{\alpha_0, \beta_0, \rho_0}, L_{\alpha_0, \beta_0, \theta_0} / U_{\alpha_0, \beta_0, \theta_0}\}$.



Observações

- Se a sub-paleta normalizada $(\alpha_0, \beta_0, \gamma_0)$ coincide com uma peça do conjunto $S_{(\alpha_0, \beta_0, \gamma_0)}$, então a solução obtida é uma solução ótima do nó actual.
- Se o valor da melhor solução homogénea coincide com o volume da sub-paleta actual $(\alpha_0, \beta_0, \gamma_0)$, então o valor é uma solução ótima para o nó actual.

Observações

- Se a cardinalidade do conjunto $S_{(\alpha_0, \beta_0, \gamma_0)}$ é menor ou igual a um, então a solução homogénea obtida representa uma solução ótima para o nó actual.
- Se $(l_p, w_q, h_r) > \frac{1}{2}(\alpha_0, \beta_0, \gamma_0)$, então a melhor solução homogénea é uma solução ótima para $(\alpha_0, \beta_0, \gamma_0)$, onde $l_p = \min_{1 \leq i \leq |S|} \{l_i\}$, $w_q = \min_{1 \leq i \leq |S|} \{w_i\}$, $h_r = \min_{1 \leq i \leq |S|} \{h_i\}$.

O algoritmo

Passo inicial:

Seja $INIT=(\alpha, \beta, \gamma)$ o nó inicial aberto.

Seja $L_{\alpha, \beta, \gamma}$ o valor da melhor solução homogénea e $NOD=INIT$.

Gerar os 3 conjuntos de pontos $X_{(\alpha, \beta, \gamma)}$, $Y_{(\alpha, \beta, \gamma)}$ e $Z_{(\alpha, \beta, \gamma)}$.

O algoritmo

Passo principal:

Enquanto $INIT$ não está fechado, faz:

1. Gerar os nós sucessores do NOD até que uma das estratégias seja satisfeita. Para cada nó gerado (α, β, γ) , considerar a sua sub-paleta normalizada e calcular os limites inferior e superior para evitar ramificações desnecessárias. Actualizar o limite inferior até que o nó antecessor é alcançado.
2. Usar a estratégia 5 para escolher o caminho e fechar os nós que não têm sucessores ou que os sucessores estão fechados.
3. Seleccionar um nó, no caminho escolhido, para actualizar o nó actual.

Aplicação

- 64 casos
 - 32 não pesados ($c_i = l_i \cdot w_i \cdot h_i$)
 - 32 pesados
1. Comparar a performance do SGT com DPT (*Dynamic Programming Techniques*).
 2. Comparar a performance do pesado com o não pesado pelo algoritmo SGT.
 3. Comparar a performance do fixo com o rodado pelo algoritmo SGT.
 4. Comparar a sensibilidade de ambos os algoritmos ao tamanho da Grande Palete.
 5. Comparar a sensibilidade de ambos os algoritmos à variação do número do tipo de peças.

Casos

Inst	L	W	H	n	Inst	L	W	H	n
xinst1	25	45	33	5	xinst17	201	205	159	50
xinst2	30	55	65	9	xinst18	231	185	207	50
xinst3	59	65	33	8	xinst19	54	63	51	10
xinst4	75	76	81	12	xinst20	63	61	74	10
xinst5	128	100	96	5	xinst21	74	84	78	10
xinst6	100	98	96	20	xinst22	95	88	92	10
xinst7	125	105	112	20	xinst23	93	102	92	10
xinst8	95	125	89	25	xinst24	100	89	112	10
xinst9	145	150	147	30	xinst25	105	119	121	10
xinst10	100	156	137	10	xinst26	115	135	123	10
xinst11	253	294	281	10	xinst27	126	112	125	10
xinst12	128	483	275	10	xinst28	135	133	127	5
xinst13	306	214	475	20	xinst29	135	133	127	10
xinst14	100	156	137	20	xinst30	135	133	127	15
xinst15	485	248	254	10	xinst31	135	133	127	20
xinst16	487	249	255	20	xinst32	135	133	127	25

Resultados

Inst	The Fixed case: FU_3DC			The rotated case: RU_3DC		
	Opt	CT-DPT	CT-SGT	Opt	CT-DPT	CT-SGT
Uinst1	36975	0.4	0.1	37125	0.4	< 0.1
Uinst2	102681	1.6	0.5	106831	1.7	0.8
Uinst3	119608	2.1	0.4	126246	2.2	1.3
Uinst4	445396	11.4	2.9	461392	12.1	3.0
Uinst5	1177386	47.5	2.2	1225728	48.4	9.0
Uinst6	939456	32.9	17.9	940800	35.1	2.4
Uinst7	1455406	62.0	32.8	1470000	65.7	7.6
Uinst8	1034815	39.5	9.4	1054874	43.2	13.3
Uinst9	3181370	185.8	92.4	3197250	195.5	25.5
Uinst10	2002179	96.8	2.0	2116335	101.8	14.0
Uinst11	19844592	◇	55.3	20611491	◇	155.8
Uinst12	16777773	◇	365.2	16954353	◇	742.5
Uinst13	31011819	◇	928.2	31102140	◇	1200.3
Uinst14	2085881	100.9	11.3	2132736	107.7	33.9
Uinst15	28712050	◇	22.0	29822830	◇	101.4
Uinst16	29843556	◇	94.4	30569400	◇	381.3
Uinst17	6139059	493.8	8.2	6409134	536.6	23.6
Uinst18	8678052	721.2	34.8	8802495	777.6	76.8

Não pesado

Resultados

Inst	The Fixed case: FU_3DC			The rotated case: RU_3DC		
	Opt	CT-DPT	CT-SGT	Opt	CT-DPT	CT-SGT
Uinst19	158714	2.8	0.1	171317	3.2	0.2
Uinst20	263228	5.7	0.2	279629	6.0	0.8
Uinst21	458712	12.1	0.7	480858	12.9	2.4
Uinst22	729068	23.8	2.0	766795	25.1	4.6
Uinst23	837573	28.5	3.2	868991	29.9	6.6
Uinst24	953408	33.9	4.4	993108	35.2	8.6
Uinst25	1480221	61.3	11.1	1511895	63.7	14.6
Uinst26	1870785	86.7	21.4	1907561	90.1	23.4
Uinst27	1728065	77.8	18.3	1762936	80.3	20.2
Uinst28	2160194	111.2	1.8	2221874	111.7	5.2
Uinst29	2199756	113.3	12.1	2267651	113.8	16.7
Uinst30	2215217	113.7	30.1	2273091	116.6	26.5
Uinst31	2228631	114.8	39.2	2278951	120.6	27.7
Uinst32	2237110	116.0	44.5	2279696	122.8	29.4

Não pesado

Resultados

Inst	The Fixed case: FU_3DC			The rotated case: RU_3DC		
	Opt	CT-DPT	CT-SGT	Opt	CT-DPT	CT-SGT
Winst1	26298	0.3	0.1	27450	0.4	0.3
Winst2	3242	1.6	0.5	3376	1.7	0.9
Winst3	3131	2.1	0.5	3692	2.2	0.9
Winst4	12746	11.5	3.3	13434	12.1	3.8
Winst5	35845	47.8	2.5	37614	49.0	9.8
Winst6	32508	32.7	19.8	32766	35.0	12.5
Winst7	42414	62.5	36.3	43792	65.8	26.5
Winst8	29247	40.2	11.2	30601	43.1	14.8
Winst9	117056	185.4	103.5	124645	195.2	57.5
Winst10	19260	98.3	2.4	22633	101.4	14.3
Winst11	48919	◇	60.0	54586	◇	152.8
Winst12	335060	◇	374.6	363880	◇	1002.8
Winst13	338822	◇	1028.2	384336	◇	1532.9
Winst14	57578	102.9	13.1	60144	106.4	37.5
Winst15	76902	◇	21.9	85720	◇	166.3
Winst16	81510	◇	78.4	91840	◇	531.2
Winst17	16616	492.5	9.5	18053	540.2	25.3
Winst18	24288	728.7	42.7	26188	779.3	92.4

Pesado

Resultados

Inst	The Fixed case: FU_3DC			The rotated case: RU_3DC		
	Opt	CT-DPT	CT-SGT	Opt	CT-DPT	CT-SGT
Winst19	4470	2.9	< 0.1	5084	3.2	0.3
Winst20	7539	5.7	0.2	8512	6.1	0.8
Winst21	13674	12.3	0.8	14594	12.8	2.7
Winst22	22337	23.9	2.3	24300	24.7	5.4
Winst23	25453	28.9	3.4	27471	29.6	7.0
Winst24	29065	34.7	4.8	31325	34.7	9.1
Winst25	44600	62.4	11.7	47509	62.5	16.2
Winst26	59388	87.2	22.5	60649	88.5	25.6
Winst27	52524	77.8	19.6	56359	79.5	22.2
Winst28	63872	111.3	1.9	74956	114.6	5.3
Winst29	73321	114.1	12.7	83086	115.8	17.2
Winst30	82929	115.3	32.0	88947	117.7	27.0
Winst31	82929	115.6	41.5	88947	119.0	30.6
Winst32	82929	116.1	48.7	88947	121.2	33.5

Pesado

Conclusões

- $(tempoDPT - tempoSGT)/tempoDPT$
- 80,41% - não pesado FU_3DC
- 79,17% - não pesado RU_3DC
- 78,35% - pesado FU_3DC
- 73,34% - pesado RU_3DC

- 77,82% - Total

- O SGT supera constantemente o DPT.

- O DPT mantém-se um algoritmo útil, pois resolve o problema geral (U_G3DC), enquanto o SGT está limitado ao caso de apenas uma paleta (U_3DC).

Conclusões

- O SGT tem melhor desempenho com problemas não pesados.
- O SGT no geral tem melhor desempenho com problemas Fixos.
- O tempo computacional aumenta com o aumento das dimensões da grande paleta, em ambos os algoritmos.
- O tempo computacional aumenta com o aumento do nº do tipo de peças, em ambos os algoritmos. O SGT é mais sensível a essa variação, mas continua mais eficiente.

Bibliografia

- Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study. *Computer & Operations Research* 2004
- Gilmore PC, Gomory RE. The theory and computation of knapsack functions. *Operations Research* 1966
- Hifi M, Zissimopoulos V. A recursive exact algorithm for weighted two-dimensional cutting. *European Journal of Operational Research* 1996
- Christofides N. Whitlock C. An algorithm for two-dimensional cutting problem: a best-first branch-and-bound exact algorithm. *International Transactions in Operational Research* 2000.