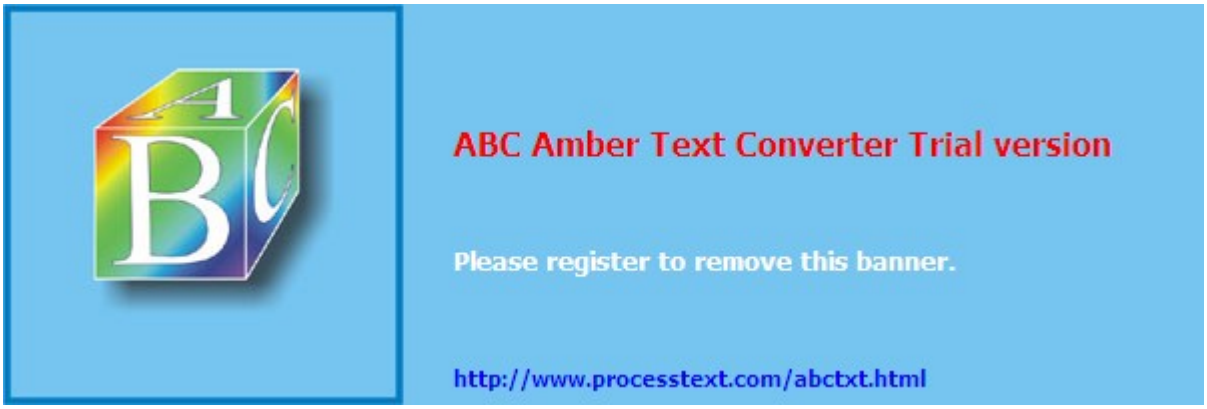
ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html



Definisi

Proses Disaggregasi adalah proses penyamaan (generalisasi) dari satuan aggregat kedalam satuan *end item* berdasarkan faktor konversi, hasil disaggregasi ini berupa jadwal induk produksi/MPS.

Tujuan

Untuk menyusun jadwal induk produksi (MPS) setelah diketahui jadwal produksi agregatnya (P_t)

Metoda

Terdapat banyak cara, baik yang bersifat analitis atau heuritis, antara lain:

- Pendekatan *Hax & Meal*
- Pendekatan *Britan & Hax*
- Rencana yang lebih tinggi menjadi pembatas atau kendala bagi rencana tingkat rendah
- Agregat taktis (operasional)

Metoda Britan & Hax

Langkah-langkah pengerjaan:

1. Pilih *family* yang akan diproduksi pada perioda yang bersangkutan.

- Suatu *family* akan diproduksi bila salah satu *item* dari suatu *family* tersebut memenuhi syarat berikut:

$$I_{ij.t-1} - D_{ij.t} \leq S_{ij}$$

ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html

Dimana:

$I_{ij.t-1}$ = Tingkat persediaan pada akhir perioda t – 1 dari *item j family I*

$D_{ij.t}$ = Permintaan *item j family i* pada perioda t

S_{ij} = Cadangan pengaman *item j* dalam *family i*

Contoh:

Tabel 1

Family *i*	*Item* *j*	Inventory $I_{ij.t-1}$	Demand $D_{ij.t}$	Safety Stock S_{ij}	Expected Quantity $I_{ij.t-1} - D_{ij.t}$	Konversi K_{ij}
A	1	240	170	50	70	0.85
	2	285	200	75	85	1.10
	3	122	100	40	**22**	0.90
B	4	223	130	50	93	1.15
	5	290	170	50	120	1.05
	6	193	110	40	83	1.20
	7	420	210	60	210	1.15
C	8	235	150	40	85	0.75
	9	135	100	50	**35**	0.85
	10	180	140	50	**40**	0.80

Family A dan C diproduksi

- *Item* yang berjumlah kurang dari *safety stock* harus segera dibuat supaya tidak terjadi kekurangan.

2. Tentukan jumlah yang akan diproduksi dari *family* yang terpilih dengan menggunakan model *knapsack.*

Min: Z =

Subject to:

$x_i \geq LB_i$

$x_i \leq UB_i$

Dimana:

h_i = Holding cost untuk *item* dalam *family i*

x_i = Jumlah unit *family i* yang diproduksi

S_i = Ongkos set up untuk *family* i

ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html

k_{ij} = Faktor konversi untuk unit *item* j dalam famili i terhadap unit produk agregat

D_{ij} = Permintaan untuk *item* j dalam *family* i selama masa produksi t

x^* = Jumlah produksi menurut perencanaan agregat

LB_i = Batas bawah untuk memproduksi *family* i

UB_i = Batas atas untuk memproduksi *family* i

z = set dari *family* yang akan diproduksi

Batas bawah:

$$LB_i = \sum_{\forall j \in i} \text{Max}\left[0, k_{ij}\left(D_{ij.t} - I_{ij.t-1} + SS_{ij}\right)\right]$$

Batas bawah bila dikehendaki ada *safety stock*

Batas atas:

$$UB_i = \sum_{\forall j \in i} k_{ij}\left[\left(\sum_{k=0}^{n-1} D_{ij.t+k}\right) - I_{ij.t-1} + SS_{ij}\right]$$

Batas atas bila tidak diinginkan akumulasi *inventory* terlalu banyak

Batas atas dan batas bawah bisa diabaikan bila tidak dikehendaki atau tidak sesuai dengan rencana produksi.

Bila dikehendaki ada batas, dan: , maka dapat dilakukan algoritma disagregasi *family* dan algoritma disagregasi *item*.

ALGORITMA DISAGREGASI *FAMILY*

- Langkah 1:

  Hitung untuk semua $i \in z'$:

- Langkah 2:

  Untuk setiap $i \in z'$:

ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html

Jika $LB_i \leq y_i^{\beta} \leq UB_i$, maka buat $y_i^* = y_i^{\beta}$

Untuk *family* lain lain, teruskan ke langkah 3.

- Langkah 3:

  Bagi *family* lain dibagi ke dalam dua kelompok

  $z_+^{\beta} = \{ i \in z^{\beta} : y_i^{\beta} > UB_i \}$ set dari semua *family* dimana : $y_i^{\beta} > UB_i$

  $z_-^{\beta} = \{ i \in z^{\beta} : y_i^{\beta} < LB_i \}$ set dari semua *family* dimana : $y_i^{\beta} < LB_i$

  Hitung:

  $$\Delta^+ = \sum_{i \in z_+^{\beta}} \left(y_i^{\beta} - UB_i \right)$$

  $$\Delta^- = \sum_{i \in z_-^{\beta}} \left(LB_i - y_i^{\beta} \right)$$

- Langkah 4:

  Bila:

  D^+ / D^-, buat $y_i^* = UB_i$ untuk semua $i \in z_+^{\beta}$

  $D^+ < D^-$, buat $y_i^* = LB_i$ untuk semua $i \in z_-^{\beta}$

  Buat $\beta = \beta + 1$

  $z^{\beta+1} = z^{\beta}$ - {semua *family* yang y_i^* telah diperoleh}

  $P^{\beta+1} = P^{\beta} - y_i^*$ (untuk semua i yang dijadwalkan dalam iterasi β)

  Bila $z^{\beta+1} = 0$, maka stop

  Bila $\neq 0$, maka kembali ke langkah 1 (iterasi 2)

CONTOH

Lihat tabel 1

Diketahui:

- P_t = 450 unit (rencana agregat)
- Ongkos setup untuk *family* A dan C masing-masing adalah Rp. 2.000 dan Rp. 1.000
- Batas atas ditentukan 3 kali ramalan demand dikurangi inventori perioda sebelumnya

$\beta = 1$, $P' = x^*$, dan $z' = z$

ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html

$$y_i' = \frac{\sqrt{S_i . \sum_{\forall j \in i} (k_{ij} . D_{ij.t})}}{\sum_{i \in z} \sqrt{S_i . \sum_{\forall j \in i} (k_{ij} . D_{ij.t})}} . P'$$

$$y_A' = \frac{\sqrt{S_A . \sum_{\forall j \in i} (k_{ij} . D_{ij.t})}}{\sum_{i \in z} \sqrt{S_A . \sum_{\forall j \in i} (k_{ij} . D_{ij.t})}} . x^*$$

Family i	*Item* j	Inventory I_{ijt-1}	Demand D_{ij}	Konversi k_{ij}	$K_{ij}.D_{ij}$
A	1	240	170	0.85	144.5
	2	285	200	1.10	220
	3	122	100	0.90	90
				$\sum k_{Aj} . D_{Aj} = 454.5$	
C	8	235	150	0.75	112.5
	9	135	100	0.85	85
	10	180	140	0.80	112
				$\sum k_{Cj} . D_{Cj} = 309$	

$$\sqrt{S_A \sum_{\forall j \in A} (k_{Aj} . D_{Aj.t})} = \sqrt{2000 \text{x} 454.5} = 953.41$$

dibulatkan 953

dibulatkan 556

$x^* = P_t = 450$

Cek batas atas dan batas bawah

Batas Bawah:

$LB_i =$

Family A:

j = 1 Max [0 , 0.85(170 – 240 + 50)]

Max [0 , -17] = 0

ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html

j = 2 Max [0 , 1.10(200 – 285 + 75)]

Max [0 , -11] = 0

j = 3 Max [0 , 0.90(100 – 122 + 40)]

Max [0 , 16.2] = 16.2

$$\sum_{\forall j \in A} \text{Max}\left[0,0+0+16.2\right]=16.2$$

maka $LB_A = 16.2$

Family C:

j = 8 Max [0 , 0.75(150 – 235 + 40)]

Max [0 , -33.75] = 0

j = 9 Max [0 , 0.85(100 – 135 + 50)]

Max [0 , 12.75] = 12.75

j = 10 Max [0 , 0.80(140 – 180 + 50)]

Max [0 , 8] = 8

$$\sum_{\forall j \in A} \text{Max}\left[0,0+12.75+8\right]=20.75$$

maka $LB_C = 20.75$

Batas Atas (tiga kali ramalan demand - $I_{ij.t-1}$)

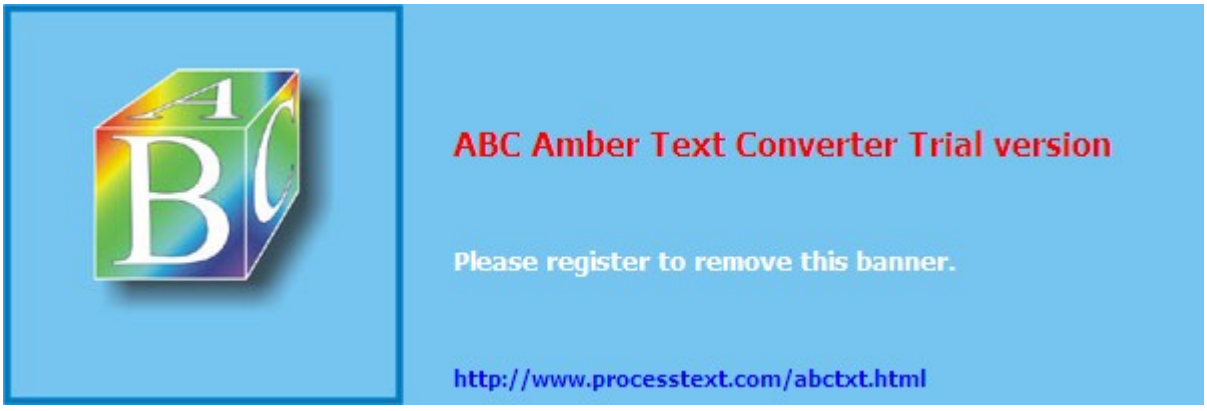
Family A:

$$UB_A = \sum_{\forall j \in A} k_{ij}\left(3D_{ij} - I_{ij.t-1}\right)$$

= (0.85(3 x 170) - 240) + (1.1(3 x 200) – 285) + (0.9(3 x 100) – 122)

= 736.2

Family C:



UB_C =

= (0.75(3 x 150) - 235) + (0.85(3 x 100) – 135) + (0.8(3 x 140) – 180)

= 493

ABC Amber Text Converter Trial version, http://www.processtext.com/abctxt.html

$LB_A = 16.2$ $LB_C = 20.75$

$UB_A = 736$ $UB_C = 493$

$y_I^A = 284$ $y_I^C = 166$

Dinyatakan valid

