



KTH Matematik

Laboration 1, Systemteknik del 2

Syfte

Denna laboration syftar till att ge förståelse för prognoser och lagerstyrning speciellt för situationen med periodisk inspektion. Laborationen sker i grupper om två personer. Det förutsätts att ni inom gruppen egenhändigt skriver den Matlab-kod som efterfrågas. Det betraktas som fusk att kopiera andra gruppers lösningar. Laborationen skall dels redovisas skriftligt i form av en rapport som innehåller resultat och dokumentation till de Matlab program som tagits fram, dels muntligt i form av en demonstration av programmet för Ulf Brännlund. Tid för den muntliga redovisning skall bokas på listan utanför Uffes kontor. Den skriftliga delen skall skickas per e-mail till uffemath@kth.se åtminstone två arbetsdagar före den muntliga redovisningen. Observera att den muntliga redovisningen inte syftar till att se om ni har gjort ett tillfredställande datorprogram utan till att kontrollera att ni (båda) förstått teorin. Laborationen motsvarar 2 poäng.

Nödvändig teori för att klara denna laboration finns i Axsäters bok i kapitlen 2 till och med 4.

Bakgrund

Företaget Koff Ein AB säljer tre olika sorters kaffebryggare. Bryggarna skiljer sig åt dels vad gäller designen, dels vad gäller maximalt antal koppar som kan bryggas på en gång.

Din uppgift är att ta fram ett prognos- och lagerstyrningssystem för detta företag.

Koff Ein AB har som policy att hålla bristerna i lagret på en mycket låg nivå. De önskar att det förväntade antal tillfällen då det blir brist är högst 2 gånger per år och produkt. Dessutom vill man att minst 96 procent av efterfrågan skall betjänas direkt från lagret.

Olika data för kaffebryggarna (A,B,C) finns i Tabell 1.

	A	B	C
Ledtid (veckor)	2	3	4
Produktvärde efter tillverkning (kr/st)	260	230	215
Uppsättningskostnad (kr/gång)	2100	2200	2400
Bristkostnad (kr/st)	100	100	100
Ingångslager (st)	263	280	300

Tabell 1: Produktdata

Lagerhållningsräntan är 20 % per år. I tabell 2 återfinns efterfrågan på de tre kaffebryggarna under de två senaste 26-veckors perioderna.

Datorlaboration

Datorlaborationen går ut på att skriva ett eget prognos- och lagerstyrningssystem i Matlab och att sedan simulera detta mot en tänkt efterfrågan.

Vecka	1	2	3	4	5	6	7	8	9	10	11	12	13
A	73	82	57	80	60	76	83	52	76	62	85	53	86
B	90	96	78	94	79	90	94	72	89	78	94	71	93
C	76	83	63	80	63	75	81	55	74	63	80	54	104

Vecka	14	15	16	17	18	19	20	21	22	23	24	25	26
A	56	86	74	73	65	82	83	72	82	88	75	73	83
B	72	93	84	82	76	88	88	80	86	90	81	79	85
C	55	79	69	67	60	74	73	64	72	76	65	63	70

Vecka	27	28	29	30	31	32	33	34	35	36	37	38	39
A	99	71	79	89	100	92	85	91	73	97	94	105	84
B	96	76	81	88	95	89	84	87	74	90	87	95	80
C	82	60	65	73	82	75	68	73	58	76	73	82	89

Vecka	40	41	42	43	44	45	46	47	48	49	50	51	52
A	83	92	91	94	99	93	123	87	97	92	98	110	103
B	78	85	83	85	88	83	104	78	85	81	85	92	87
C	63	70	68	70	74	68	91	62	70	65	70	78	73

Tabell 2: Efterfrågan senaste 52 veckorna

Målet är att minimera de totala uppsättningskostnaderna, lagerhållningskostnaderna och bristkostnaderna.

Beslut om beställning skall tas vecka för vecka och man kan bara inspektera lagernivåerna veckovis.

På hemsidan för denna kurs,

<http://www.math.kth.se/optsys/studinfo/5B1846/5B1846.html>, finns nedanstående filer

- `olddem.m` ovanstående efterfråge-data,
- `nextweek.m` som genererar efterfrågan veckovis för de tre produkterna,
- `main.m` som är en stomme i vilket ni kan komplettera med nedanstående moduler, och annat som behövs för att kunna simulera några års lagerhållning,
- `plotall.m` plottar lagernivåerna, efterfrågan och produktionen för de tre produkterna som funktion av tiden,
- `normdens.m` täthetsfunktionen (ϕ) för en normalfördelad $N(0,1)$ -variabel,
- `normdist.m` fördelningsfunktionen (Φ) för en normalfördelad $N(0,1)$ -variabel,
- `norminv.m` inversen till fördelningsfunktionen (Φ) för en normalfördelad $N(0,1)$ -variabel,
- `normpart.m` partiellt förväntat värde ($G(k)$) för en normalfördelad $N(0,1)$ -variabel.

OBS: `Plotall.m` och `nextweek.m` använder några gemensamma data som globala variabler. Detta betyder att ni *inte* får använda (eller titta på) följande globala variabler (med stora bokstäver):

DEMAND, TOTALCOST, PRODUCTION, INVENTORY, DEMANDINIT, DRIFT, TBREAK, DEMANDSIGMA, INVLOT, DEMLOT, PRODLOT.

Uppgifter

Nedan följer ett antal uppgifter som ni förväntas göra, fundera över och besvara. Koden skall vara dokumenterad på sådant sätt att den är lätt att följa även av en som inte är expert på området.

1. Kontrollera genom att köra och studera `main.m`, och `nextweek.m`, med någon simpel produktionsbeslutsfunktion att ni förstår i vilken ordning som beslut fattas, hur kostnader beräknas och vad som menas med ledtid. Tänk på att det är frågan om periodisk inspektion. En sådan enkel produktionsbeslutsfunktion skulle kunna vara att beställa varannan vecka och lika mycket som gått åt under de två senaste veckorna)
2. Skriv en Matlab-funktion (`prognos.m`) som beräknar en prognos för efterfrågan under ledtiden för de tre produkterna. Detta skall baseras på exponentiell utjämning med trend. Analysera förra årets data för att få bra initialvärden för prognoserna.
3. Skriv en Matlab-funktion (`kontroll.m`) som gör kontroll av prognoserna. (Se sidorna 34 och 35 i Axsätters bok). Simulera efterfrågan under några år med ert prognosystem. Verkar ert prognosystem fungera tillfredsställande? Kolla på plottarna med efterfrågan: Sker det något trendbrott för någon av produkterna? Verkar ert prognosystem fånga upp detta? (För betyget 4: Bygg in i programmet en dialog med användaren så att han kan modifiera värdena i prognosystemet om det skulle behövas. Du måste också ge användaren en begriplig ledtext, så att han kan förstå vad som är fel.)
4. Skriv en Matlab-funktion (`qs.m`) som bestämmer orderkvantiteter och beställningspunkter för de tre produkterna. (För betyget 4 skall modellen för samtidig bestämning av beställningspunkt och orderkvantitet användas. Dessutom måste man förstås också kolla att servicekraven är uppfyllda och modifiera beställningspunkt och orderkvantitet om det behövs.)

Obs: För att undvika att man får ett alltför "svajigt" system bör inte dessa uppdateras alltför ofta. Det är ofta inte heller praktiskt möjligt att göra detta. Som en tumregel brukar man säga att man bör uppdatera orderkvantiteter och beställningspunkter högst en gång i halvåret. Det betyder att om man uppdaterar dessa halvårsvis bör man basera t.ex. orderkvantiteten på den förväntade medelefterfrågan under det *kommande* halvåret.

5. Skriv en Matlab-funktion (`produce.m`) som bestämmer i en viss vecka om och i så fall hur mycket som skall beställas av varje produkt.
6. Simulera systemet under 4 år (208 veckor). Uppfylls servicekraven? Vad blir i snitt den totala kostnaden per år? Gör 10 stycken körningar vardera på 2 år. Vad blir nu totalkostnaden per år i snitt?
7. Antag nu att det finns möjlighet att göra samtidig beställning av de tre kaffebryggarna. Antag att den gemensamma orderkostnaden är 1900 kronor och att de individuella ordersärkostnaderna är lika med 100, 200 respektive 300 kronor för A, B,

respektive C. Skriv en Matlab funktion (`qgemen.m`) som beräknar lämpligt val av orderkvantiteter för de tre olika produkterna. Ni får anta att de tre kaffebyggarna alltid beställs samtidigt, dvs n_i i kapitel 4.2.1 är lika med 1 för alla produkter. Skriv också en Matlab-funktion (`prodgem.m`) som avgör om en gemensam beställning skall ske. Använd servicekravet $SERV_2 = 0.96$ = andel av efterfrågan i kronor som kan hämtas direkt från lager. Modifiera `nextweek.m` så att kostnaderna beräknas korrekt.

8. Simulera detta system under 4 år (208 veckor). Studera kurvorna över lagernivåerna. Verkar det vara ett rimligt sätt att styra lagren? Föreslå ett annat sätt att styra lagren med "gemensam beställning". Implementera!

För betyget 4 krävs utöver ovanstående att koden är mycket väldokumenterad (+motive-rad), och att båda gruppmedlemmarna kan redogöra för och motivera alla bitar av koden vid redovisningen. För betyget 4 krävs också att ni redovisat laborationen muntligt för Uffe före fredagen 19 november.