E-Appendix to: Joint Inventory Replenishment and Component Allocation Optimization in an Assemble-to-Order System

Yalçın Akçay[1] Susan H. Xu[2]

Test Problem Parameters and Additional Results

1 Test Problem 1 (Zhang 1997)

The problem parameters for this four product, five component system are given in Table 1.1. It is assumed that demands are normally distributed and the response time window for all products is one period.

Table 1.1: Problem setting of the ATO system from Zhang

				Products			
			j	1	2	3	4
			Mean	100	150	50	30
			Std. Dev.	25	30	15	11
Components			r_j	1	1	1	1
i	c_i	L_i		*Usage Rates*			
1	2	3		1	1	0	0
2	3	2		2	1	1	0
3	6	2		1	1	1	0
4	4	4		0	0	1	1
5	1	4		0	0	0	1

2 Test Problem 2 (Chen et al. 2000)

This is a PC assembly system with realistic problem data from the IBM Personal Systems Group. In this system, a family of six desktop computers are assembled from a set of seventeen components. The demand for each product is normally distributed with a mean of 100 and a standard deviation of 20. Other problem parameters are given in Table 2.2. The reward of filling an order for a particular PC before its response time window is taken as 10% of the selling price of that PC.

3 Test Problem 3 (Agrawal and Cohen 2001)

The problem parameters for this four product, two component system are given in Table 3.3. Each product demand follows a normal distribution in each period, and the response time window of all products is zero.

[1]College of Administrative Sciences and Economics, Koç University, Rumeli Feneri Yolu, Sariyer, Istanbul, 34450, Turkey, Email: yakcay@ku.edu.tr

[2]Department of Supply Chain and Information Systems, The Smeal College of Business Administration, Penn State University, University Park, Pennsylvania 16802, USA, Email: shx@psu.edu

Table 2.2: Problem settings of the PC assembly system

				PRODUCTS						
				j	1	2	3	4	5	6
				r_j	1,363	1,595	1,765	1,494	1,494	1,628
i	COMPONENTS	c_i	L_i	*Component Usage Rates*						
1	Shell	42	5		1	1	1	1	1	1
2	Shell (Common Parts 1)	114	8		1	1	1	1	1	1
3	Shell (Common Parts 2)	114	8		1	1	1	1	1	1
4	Processor 450 MHz	307	12		1	0	0	0	0	0
5	Processor 500 MHz	538	12		0	1	0	0	0	0
6	Processor 550 MHz	395	12		0	0	1	1	1	0
7	Processor 600 MHz	799	12		0	0	0	0	0	1
8	Memory 64MB	290	15		1	1	1	1	1	1
9	Hard drive 6.8GB	155	18		1	1	0	0	0	1
10	Hard drive 13.5GB	198	18		0	0	1	1	1	0
11	Hard drive (Common Parts)	114	8		1	1	1	1	1	1
12	Software Pre-load 1	114	4		1	1	1	0	1	0
13	Software Pre-load 2	114	4		0	0	0	1	0	0
14	CD-ROM	43	10		0	0	1	0	0	0
15	CD-ROM (Common Parts)	114	10		0	0	1	0	0	0
16	Video Graphics Card	114	6		1	1	1	1	1	0
17	Ethernet Card	114	10		0	0	1	0	0	0

Table 3.3: Problem setting of the ATO system from Agrawal and Cohen

				PRODUCTS			
			j	1	2	3	4
			Mean	15	18	18	15
			Std. Dev.	3	3	3	3
COMPONENTS			r_j	1	1	1	1
i	c_i	L_i		*Usage Rates*			
1	10	4		1	3	3	1
2	10	6		2	1	1	2

The effect of joint optimization for this ATO system is depicted in Figure 3.1. Similar to our observations from the other test problems, we note that the system performance under SAA-OBCA$_\alpha$ approximately achieves the performance of the system under the optimal base-stock levels and the optimal allocation rule (UB). Moreover, the jointly optimized system achieves a higher service level for a fixed budget, and needs a smaller inventory investment for a desired service level, as compared to other heuristics.

Table 3.4 reports the base-stock levels and their corresponding type-II service levels using SAA, EF$_{\mathrm{I}}$ and EF$_{\mathrm{II}}$ for inventory replenishment and the optimal policy for component allocation. At the first glance, the reader might be surprised that, unlike the previous test problems from literature, SAA is only marginally better than EF$_{\mathrm{I}}$ and EF$_{\mathrm{II}}$, even at low service levels. A close examination of Table 3.3 shows that this is due to the symmetric nature of the component usage matrix, demand distributions and cost structures. The only difference between the two components is that they have different lead times. It then comes to no one's surprise that EF$_{\mathrm{I}}$ and EF$_{\mathrm{II}}$ can perform quite well for the ATO system where the parameters associated with individual components are similar.

In Table 3.5, the service levels using the SAA solution and the heuristic allocation rules, OBCA$_\alpha$, with $\alpha = 0$, 0.5 and 1.0, FS and FP, are reported. Notice that OBCA$_\alpha$ significantly outperforms the other two rules. In fact, for this "symmetric" system, the superiority of SAA-OBCA$_\alpha$, as compared to other methods, is mainly contributed by OBCA$_\alpha$.

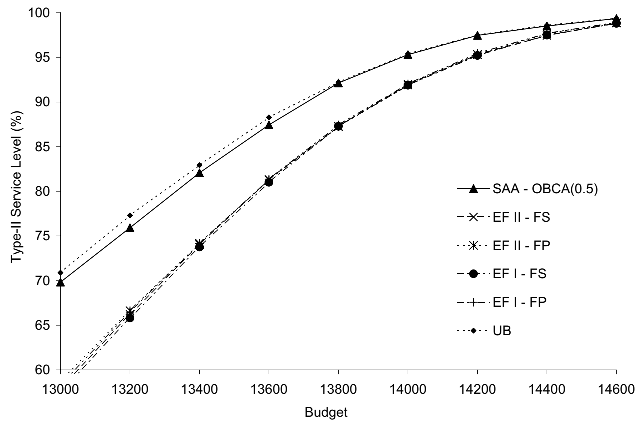

Figure 3.1: Service levels of Agrawal and Cohen's system under different inventory replenishment and component allocation policies

Table 3.4: Base-stock levels and type-II SL of Cohen and Agrawal's system using different replenishment policies and the optimal allocation policy

	SAA			EF_{II}			EF_{I}		
B	S_1	S_2	OA	S_1	S_2	OA	S_1	S_2	OA
13,000	645	655	69.83	653	647	68.28	656	644	67.56
13,200	657	663	75.92	664	656	75.36	667	653	74.84
13,400	669	671	82.07	676	664	81.68	678	662	81.37
13,600	681	679	87.44	687	673	87.28	689	671	87.04
13,800	693	687	92.13	698	682	91.61	700	680	91.52
14,000	703	697	95.30	709	691	94.83	711	689	94.78
14,200	714	706	97.46	720	700	97.06	722	698	97.01
14,400	725	715	98.51	731	709	98.47	732	708	98.46
14,600	736	724	99.34	742	718	99.28	743	717	99.27
Average			88.67			88.20			87.98

Finally, Table 3.6 summarizes the performance statistics of all allocation rules. These results are consistent with our observations from the previous test problem.

4 Test Problem 5 (An ATO System with different response time windows)

We consider the same system as in Test Problem 4. However, we assume that the response time window of each product is equally likely to be either 0, 1, or 2 periods. Products are grouped according to their response time windows, and the ones with the same response time window are gathered in the same product family. The members of the product families are different in different problem instances due to the stochastic nature of the problem. The base-stock level of each component is set as in Test Problem 4.

We generate 100 random problem instances and obtain the reward ratio of the system under OBCA_α and its prioritized version, OBCA^p_α. The results are summarized in Table 4.7. We note, firstly, that the reward ratio is increased compared to its counterpart in Test Problem 4. This is expected, since the response time windows allow the system a

Table 3.5: Service levels of Agrwwal and Cohen's system using SAA base-stock levels and different allocation rules

B	Type-I	Type-II						
		UB	OA	$\mathrm{OBCA}_{0.1}$	$\mathrm{OBCA}_{0.5}$	$\mathrm{OBCA}_{1.0}$	FS	FP
13,000	2.4	70.89	69.83	69.83	69.83	68.46	57.97	58.64
13,200	6.8	77.30	75.92	75.92	75.92	75.11	66.21	66.28
13,400	15.2	82.93	82.07	82.07	82.07	81.46	74.17	73.47
13,600	27.6	88.28	87.44	87.44	87.44	87.06	81.19	80.23
13,800	41.0	92.22	92.13	92.13	92.13	91.74	87.25	86.46
14,000	55.6	95.39	95.30	95.30	95.30	94.83	91.68	90.77
14,200	70.4	97.51	97.46	97.46	97.46	97.14	95.59	95.58
14,400	81.2	98.60	98.51	98.51	98.51	98.51	97.48	97.32
14,600	90.8	99.37	99.34	99.34	99.34	99.34	98.93	98.89
Average	89.16	88.67	43.4	88.67	88.67	88.18	83.38	83.07

Table 3.6: Performance statistics of Agrawal and Cohen's system with SAA base-stock levels and different allocation rules

	$\mathrm{OBCA}_{0.1}$			$\mathrm{OBCA}_{0.5}$			$\mathrm{OBCA}_{1.0}$			FS			FP		
B	Err.	Hit	CPU	Err.	Hit	CPU	Err.	Hit	CPU	Err.	Hit	CPU	Err.	Hit	CPU
13,000	0.0	100.0	0.17	0.0	100.0	0.05	2.11	68.0	0.00	18.14	2.4	1.39	16.68	10.8	1.52
13,200	0.0	100.0	0.03	0.0	100.0	0.01	1.16	78.2	0.03	14.43	6.8	1.37	14.02	20.2	1.49
13,400	0.0	100.0	0.05	0.0	100.0	0.03	0.82	84.8	0.05	10.93	15.2	1.29	11.58	29.2	1.46
13,600	0.0	100.0	0.16	0.0	100.0	0.05	0.58	89.4	0.01	7.95	27.6	1.36	8.95	41.0	1.36
13,800	0.0	100.0	0.10	0.0	100.0	0.05	0.35	93.0	0.01	5.42	41.0	1.18	6.26	54.2	1.63
14,000	0.0	100.0	0.17	0.0	100.0	0.01	0.18	97.0	0.03	3.57	55.8	1.40	4.56	65.2	1.31
14,200	0.0	100.0	0.03	0.0	100.0	0.05	0.08	98.6	0.01	2.16	70.4	1.26	2.69	78.0	1.50
14,400	0.0	100.0	0.10	0.0	100.0	0.05	0.03	99.6	0.05	1.14	81.2	1.11	1.46	87.0	1.54
14,600	0.0	100.0	0.05	0.0	100.0	0.05	0.01	99.8	0.0	0.52	90.8	1.32	0.66	93.6	1.61
Average	0.0	100.0	0.09	0.0	100.0	0.039	0.59	89.82	0.02	7.14	43.47	1.29	7.43	53.24	1.49

longer duration for filling demands and collecting rewards. Secondly, the reward ratio with OBCA_α does not measure up to the one with OA, which indicates that OBCA_α cannot handle different response time windows very effectively. Nevertheless, we can improve the performance of OBCA_α considerably by prioritizing product families and using OBCA^p_α.

Table 4.7: Reward ratios for Test Problem 5 under $\mathrm{OBCA}_{0.5}$ and $\mathrm{OBCA}^p_{0.5}$, with positive time windows

z	OA	$\mathrm{OBCA}_{0.1}$	$\mathrm{OBCA}_{0.5}$	$\mathrm{OBCA}_{1.0}$	$\mathrm{OBCA}^p_{0.1}$	$\mathrm{OBCA}^p_{0.5}$	$\mathrm{OBCA}^p_{1.0}$
0.0	79.71	74.35	73.85	72.60	77.89	76.85	75.37
0.5	86.54	83.13	82.80	81.54	85.29	84.63	83.28
1.0	92.38	90.65	90.45	89.55	91.76	91.40	90.46
1.5	96.43	95.72	95.61	95.12	96.19	96.01	95.52
2.0	98.78	98.57	98.52	98.47	98.71	98.65	98.58
2.5	99.62	99.56	99.55	99.53	99.60	99.59	99.56