Journal of the Korean Institute
of Industrial Engineers
Vol.24, NO.4, December, 1998

A Contour Line Approach to Storage Location Configurations for Dual Command Operations

등고선 접근방식을 이용한 복식명령작업 저장위치형태의 결정

Byung Chun Park*

박병춘*

Abstract

This paper examines the effect of storage location configurations on dual command cycle times for the efficient operation of automated storage/retrieval systems. We use a contour line approach to determine storage location configurations. We present a contour line configuration generating scheme and a location indexing scheme. Given a contour line configuration, the location indexing scheme provides a unique priority to each location. The location priority is then used for determining the storage location of an incoming load. To investigate the effect of alternative contour line configurations on dual command cycle times, we perform a series of experiments under various storage policies.

1. Introduction

Unit load and miniload automated storage/retrieval systems (AS/RS) are common components of integrated manufacturing and distribution systems. An AS/RS consists of storage racks, storage/retrieval (S/R) machines, and input/output (I/O) stations. The other common components include fire protection systems, storage modules (pallets, baskets, containers), aisle hardware, aisle transfer cars, conveyors and other transportation equipment, and controls. Unit load and miniload AS/RS operations can be classified according to their operating mode: single and dual command. A single command cycle consists of a single storage or a single retrieval transaction in a round trip between the storage or retrieval location and the I/O point. A dual command cycle involves both a storage and a retrieval transaction in the round trip from and to the I/O point.

Storage policies can be classified into three categories: random, dedicated, and class-based storage. In general, *random* (or *floating-slot*) storage implies that an individual stock keeping unit (SKU) can be stored in any available storage location. *Pure random* storage implies that each empty storage slot is equally likely to be chosen for storage when a storage transaction is performed. Pure random storage is often assumed in the literature. To literally assign incoming SKUs to storage locations at random, however, would be rather laborious and inefficient.

In practice, the closest open location (COL) rule is used.

* 계명대학교 산업공학과

In the COL rule, an incoming load is assigned to the closest available location to the input point. When the warehouse is relatively full, there is no significant difference in travel distances obtained via the "equally likely" assumption and those resulting from the COL rule. In warehouses operated well below capacity, however, there can be a significant difference in travel distances.

Dedicated (or *fixed slot*) storage involves the assignment of specific storage locations or storage addresses for each product stored. Two popular variations of dedicated storage are part number sequence storage and throughput-based storage. *Throughput-based* storage, also called *turnover-based* storage, considers the relative difference in activity levels and storage requirements among products. In throughput-based storage, each location is given a priority according to a fixed preference scheme, typically reflecting access time from the I/O point. The product with higher throughput per storage location is assigned to the location with higher priority. Thus, properly assigning the priority to each location is the key to the success of this storage system.

Class-based storage is a compromise between random and dedicated storage. In class-based storage, a class of products are dedicated to a class of storage locations. Random storage is used within a class.

For the implementation of storage policies, each storage location should be given a unique priority in advance. In fact, location priorities are used in the search for an available storage location or the determination of class boundaries. They are also used for the assignment of a specific item to specific location(s) in dedicated storage. We will call a random storage policy the *priority-based open location (POL) rule* if an incoming load takes the location with the highest priority among the available locations.

The priority is usually assigned according to a fixed preference scheme. Since system performance is largely governed by this preference scheme, it is very important to identify good preference schemes. In this study, we use a contour line approach to prioritize the storage locations.

A *contour line* is a line of constant cost; sometimes called an iso-cost line or level curve. In our case, cost will be measured in terms of travel time. Once contour line configurations are determined, the prioritization of the storage locations can be done directly with these contour line configurations. Thus, contour line configurations can be used for any storage systems: random, dedicated, or class-based storage.

Location priorities have been typically based on one-way travel time from the I/O point. The resulting contour line configuration is square-in-time. For single command operations, square-in-time contour lines can provide optimal storage location configurations. An example is the cube-per-order (CPO) based storage [6, 7]. Francis [2], and Francis and White [3] showed that the CPO-based storage is optimal for systems with simple out-and-back selection of each item - single command cycles. With linear programming, Harmatuck [5] also proved that the CPO index provides optimal solutions to stock location problems in which an out-and-back selection method is used and where the cost of an order is independent of the item-type and the number of units ordered.

To determine class boundaries for dual command systems, Graves *et al.* [4] performed discrete evaluations of expected dual command cycle times in square-in-time racks. Square-in-time, concentric-square, and other class boundary shapes were considered. With empirical results, they claimed that the expected dual command cycle time with square-in-time boundaries may be at most 3% above optimal. They were unable to find any boundary shape yielding the expected dual command cycle times lower than those resulting from square-in-time boundaries.

To date, we are not aware of a procedure being developed to generate optimal contour line configurations for dual command operations. The objective of this study is to examine the effect of various contour line configurations on system performance aimed at improving the efficiency of AS/RS dual command operations. We develop a contour line configuration generating scheme and a location indexing scheme which are simple but effective in implementing dual command operations. We perform a series of experiments to investigate the effect of contour line configurations on system performance. We consider alternative storage policies and rack configurations. The storage policies considered include pure random storage, COL rule, POL rule, and turnover-based storage.

2. Storage Location Configurations

Consider a time-normalized rack with a maximum travel time of 1.0 time unit in the horizontal dimension and "b" time units in the vertical dimension. "b" is called the *shape factor* and assumed to be less than or equal to unity [1]. Assume the I/O point is located at the lower left-hand corner of the rack, which will be represented as location (0, 0) in time-normalized racks. Since the S/R machine can travel horizontally and vertically simultaneously, the Tchebyshev metric is used for travel time. Pick-up and deposit times are ignored since they are nearly deterministic and not directly related to our design problem.

The components of a dual command cycle time are *one-way travel* and *travel-between* times. To represent the contributions of the two components on dual command cycle time, we define a *preference index* for storage location (x, y) as follow:

$$\mathrm{PI}(x, y \mid \omega) = \omega \max(x, y) + E[\max(|x - X_2|, |y - Y_2|)]. \quad (1)$$

In Equation (1), random variables (X_2, Y_2) represent a retrieval location in a dual command cycle. We do not consider the term $E[\max(X_2, Y_2)]$ when defining the preference index since the term is a common constant for any storage location (x, y). Locations having the same preference index form a contour line.

We introduce the *weight factor* "ω" to adjust the relative influence of one-way travel and travel-between times on dual command cycle times. When $\omega = 0$, Equation (1) generates a contour line configuration with which the expected travel-between time can be minimized. When $\omega = \infty$, the expected one-way travel time is minimized. Since a dual command cycle time consists of two times one-way travel and a travel-between time, we can loosely say that dual command cycles are physically related to the case with $\omega = 2$.

Equation (1) can be defined in different forms. An alternative is

$$\mathrm{PI}(x, y \mid \theta) = \theta \max(x, y) + (1 - \theta) E[\max(|x - X_2|, |y - Y_2|)]. \quad (2)$$

If we set $\theta = \omega/(1+\omega)$, Equations (1) and (2) generate identical contour line configurations. We can also construct the preference index even when the I/O point is not located at the lower left-hand corner of the rack. As an example, assume the I/O point is located at an arbitrary location (u, v). Then, we can construct the preference index as follows:

$$\mathrm{PI}(x, y \mid \omega) = \omega \max(|x-u|, |y-v|) + E[\max(|x-X_2|, |y-Y_2|)]. \quad (3)$$

We want to evaluate $\mathrm{PI}(x, y \mid \omega)$ to use it for storage location assignments. To evaluate $\mathrm{PI}(x, y \mid \omega)$, the retrieval activity distribution should be known. Unfortunately, the retrieval activity distribution is determined by the storage activity distribution, an outcome of the storage location assignment policy. Thus, the retrieval activity distribution cannot be available in advance for use in evaluating PI $(x, y \mid \omega)$.

We want to generate simple but effective contour line configurations such that, once developed, they can be used for any storage systems. To fit the purpose, it is clear that the contour line configuration generating scheme should be independent of a specific storage policy. These arguments naturally lead us to use a uniform retrieval activity distribution.

Assuming uniform retrieval activity, it can be shown in [3] that

$$E[\max(|x-X_i|,|y-Y_i|)]$$

$$= \frac{\min(x,b-y)}{b}\{[\min(x,b-y)]^2/6+[\max(x,b-y)]^2/2\}$$

$$+\frac{\min(1-x,b-y)}{b}\{[\min(1-x,b-y)]^2/6+[\max(1-x,b-y)]^2/2\} \quad (4)$$

$$+\frac{\min(x,y)}{b}\{[\min(x,y)]^2/6+[\max(x,y)]^2/2\}$$

$$+\frac{\min(1-x,y)}{b}\{[\min(1-x,y)]^2/6+[\max(1-x,y)]^2/2\}.$$

Equation (4) is obtained as follows. Given location (x, y), we divide the entire rack into four regions with boundary lines $X = x$ and $Y = y$. For each region, we derive the expected one-way travel time from (x, y). By multiplying the expected one-way travel time by its normalized area (probability) for each region, and summing over all regions, we have Equation (4).

By varying the weight factor ω, we can generate various contour line configurations. Sample contour line configurations are illustrated in the Appendix. In particular, Figure 7 in the Appendix shows a contour line configuration for the rack with the I/O point at (0.5, 0).

We now develop a location indexing scheme to assign a unique priority to each location. Our scheme is based

159	158	157	156	154	153	151	150	149	148	148	149	150	151	153	154	156	157	158	159
155	152	147	146	143	141	139	138	136	135	135	136	138	139	140	144	146	147	152	158
145	142	137	133	132	130	129	127	125	124	124	125	127	129	130	132	133	137	147	157
134	131	128	123	120	119	116	115	114	112	112	114	115	116	119	121	123	133	146	156
126	122	118	113	110	108	106	105	103	101	101	103	105	106	108	110	121	132	144	154
117	111	107	104	100	97	96	93	92	91	91	92	93	96	97	108	119	130	140	153
109	102	98	95	89	87	84	83	81	79	79	81	83	84	96	106	116	129	139	151
99	94	88	85	80	77	74	72	69	67	68	69	72	83	93	105	115	127	138	150
90	86	78	75	70	65	62	60	58	56	56	58	69	81	92	103	114	125	136	149
82	76	71	64	59	54	52	49	47	45	45	56	68	79	91	101	112	124	135	148
73	66	61	53	50	43	41	37	36	34	45	56	67	79	91	101	112	124	135	148
63	57	51	44	40	33	30	27	25	36	47	58	69	81	92	103	114	125	136	149
55	48	42	35	29	24	20	18	27	37	49	60	72	83	93	105	115	127	138	150
46	39	32	26	19	16	11	20	30	41	52	62	74	84	96	106	116	129	139	151
38	31	23	17	12	8	16	24	33	43	54	65	77	87	97	108	119	130	141	153
28	22	15	10	5	12	19	29	40	50	59	70	80	89	100	110	120	132	143	154
21	14	9	4	10	17	26	35	44	53	64	75	85	95	104	113	123	133	146	156
13	7	3	9	15	23	32	42	51	61	71	78	88	98	107	118	128	137	147	157
6	2	7	14	22	31	39	48	57	66	76	86	94	102	111	122	131	142	152	158
1	6	13	21	28	38	46	55	63	73	82	90	99	109	117	126	134	145	155	159

Figure 1: Ordinal preference index matrix (n_x = 20, n_y = 20 and ω = 1)

on Equation (1). The location indexing scheme is defined such that the location with the highest preference index has the highest priority. Equation (1) generates cardinal preference indices. Instead, we will use *ordinal preference indices* since they are more readable and easier to compare. The ordinal preference indices are obtained by sorting the cardinal preference indices.

Figure 1 shows an example ordinal preference index matrix with n_x=20, n_y=20 and $\omega = 1$, where n_x and n_y are the number of columns and rows in the rack, respectively.

Note that in Figure 1, some locations have the same preference index. Ties must be broken in a real application. In this study, we use the following tie breaking rules.

Tie Breaking Rules

Rule 1: Give higher priority to the location having the shorter one-way travel time from the I/O point.

Rule 2: Give higher priority to the location having the larger min(x, y).

Rule 3: Give higher priority to the location satisfying $x > y$.

The locations having the same ordinal preference index

400	397	393	389	383	379	372	368	364	360	358	362	366	370	377	381	387	391	395	398
385	375	356	352	344	340	336	332	325	321	319	323	330	334	338	346	350	354	373	394
348	342	328	315	311	305	301	295	289	285	283	287	293	299	303	309	313	326	353	390
317	307	297	281	274	272	264	260	256	250	248	254	258	262	270	276	279	312	349	386
291	278	268	252	244	239	233	229	223	217	215	221	227	231	237	242	275	308	345	380
266	246	235	225	213	207	204	196	192	188	186	190	194	202	205	236	269	302	337	376
241	219	209	200	182	178	172	169	163	157	155	161	167	170	201	230	261	298	333	369
211	198	180	174	159	151	145	141	134	128	130	132	139	166	193	226	257	292	329	365
184	176	153	147	136	124	118	114	110	105	103	108	131	160	189	220	253	286	322	361
165	149	138	122	112	99	95	89	85	81	79	102	129	154	185	214	247	282	318	357
143	126	116	97	91	76	72	64	62	58	80	104	127	156	187	216	249	284	320	359
120	107	93	78	70	57	51	45	41	61	84	109	133	162	191	222	255	288	324	363
101	87	74	60	49	40	32	28	44	63	88	113	140	168	195	228	259	294	331	367
83	68	55	43	30	25	15	31	50	71	94	117	144	171	203	232	263	300	335	371
66	53	38	27	17	10	24	39	56	75	98	123	150	177	206	238	271	304	339	378
47	36	23	14	5	16	29	48	69	90	111	135	158	181	212	243	273	310	343	382
34	21	12	4	13	26	42	59	77	96	121	146	173	199	224	251	280	314	351	388
19	9	3	11	22	37	54	73	92	115	137	152	179	208	234	267	296	327	355	392
7	2	8	20	35	52	67	86	106	125	148	175	197	218	245	277	306	341	374	396
1	6	18	33	46	65	82	100	119	142	164	183	210	240	265	290	316	347	384	399

Figure 2: Location priority matrix (n_x = 20, n_y = 20 and ω = 1)

are tested with Rule 1 through Rule 3, sequentially. If the tie is broken during the test, further tests are ignored. Rule 1 is motivated by the fact that, in general, one-way travel time has more influence on dual command cycle time than travel-between time. Furthermore, our empirical study showed that the expected one-way travel time is more sensitive to the storage location configuration than is the expected travel-between time. So, when breaking ties, we give the highest priority to reducing one-way travel time.

If a tie remains after passing the test with Rule 1, it is tested with Rule 2. Rule 2 gives the highest priority to the location closest to the line with tangent $\pi/4$ through the origin. Rule 2 is motivated by the desire to reduce travel-between time. Rule 3 guarantees breaking ties. Rule 3 can also reduce travel-between time for non-square-in-time racks.

After applying Rules 1-3, we have a *location priority matrix*. In the location priority matrix, each location has a unique priority. Figure 2 illustrates an example location priority matrix.

3. Performance Analysis

3.1 Experimental Design

We performed a series of experiments to investigate the effect of contour line configurations on dual command cycle times. We considered the following column-row configurations: (30, 30), (30, 15), (20, 20), (20, 15), (20, 10), and (10, 10). These configurations were selected to examine the effect of shape factors and rack sizes on system performance. For each column-row configuration, contour line configurations were generated for ω = 0, 0.5, 1, 2, 3, 4, 5, 10 and ∞. In each case, we tested various storage policies, including pure random storage, COL rule, POL rule, and turnover-based storage.

Expected dual command cycle times are calculated as follows. Let N be the total number of storage locations in the rack. For each policy, we derive the normalized access frequency for the location with j-th priority, which will be denoted by $p(j)$ $j = 1, 2, \ldots N$. Then, we assign $p(j)$ to the location with j-th priority in a location priority matrix, to represent the normalized access frequency for the corresponding location of the discrete rack. Expected dual command cycle times are calculated via full enumerations over the discrete rack. For convenience, we assumed that travel times between rows are the same as those between columns. To compare the performance of alternative contour line configurations on a common basis, expected dual command cycle times were time-normalized.

3.2 Random Storage

First, we consider pure random storage. For pure random storage, each location is evenly utilized, and hence

$$p(j) = 1/N,\ j = 1, 2, \cdots, N, \tag{5}$$

where $p(j)$ denotes the normalized access frequency for the location with j-th priority.

Next, we determine $p(j)$ for the POL rule. Under the POL rule, each arriving load takes the lowest-numbered available location. Note that if $\omega = \infty$, the contour line configuration is square-in-time and the POL rule reduces to the conventional COL rule.

Let's assume loads arrive to an AS/RS following a Poisson process with rate "λ" and are removed immediately if there is no available location in the rack. Also assume durations of stay are independent and have a finite mean "τ". Then, the average rack utilization "$\hat{\delta}$" is defined as

$$\hat{\delta} = \lambda\tau/N. \tag{6}$$

Let's define

$$B(k, \hat{\delta}) = \frac{(\hat{\delta} N)^k / k!}{\sum_{j=0}^{k} (\hat{\delta} N)^j / j!}. \tag{7}$$

Then, it can be shown in [9] that the probability of hunting the location with j-th priority, on condition that the input load is stored in the rack, is given by

$$p(j) = \frac{B(j\text{-}1,\delta) - B(j,\delta)}{1\text{-}B(N,\delta)}, \quad j = 1, 2, \cdots, N. \tag{8}$$

We calculated expected dual command cycle times for δ = .5, .7, .9, 1, 1.5 and 2. Table 1 shows the expected dual command cycle times under the POL rule, where n_x = 30 and n_y = 30. A summary of the analysis follows.

Table 1. Expected dual command cycle times for the POL rule

		demand(δ)					
		0.5	0.7	0.9	1.0	1.5	2.0
	0.0	1.56572	1.66803	1.75666	1.78948	1.79825	1.79874
	0.5	1.29546	1.51511	1.71285	1.78026	1.79793	1.79856
	1.0	1.28402	1.51239	1.71269	1.78014	1.79795	1.79860
	2.0	1.28067	1.51164	1.71261	1.78008	1.79800	1.79863
ω	3.0	1.28018	1.51165	1.71271	1.78025	1.79803	1.79865
	4.0	1.28007	1.51182	1.71271	1.78023	1.79804	1.79866
	5.0	1.28016	1.51185	1.71272	1.78023	1.79804	1.79866
	10.0	1.28017	1.51185	1.71273	1.78023	1.79804	1.79866
	∞	1.28079	1.51296	1.71411	1.78055	1.79807	1.79868

For any ω, the expected dual command cycle time increased as δ increased. We had larger increments for $\delta \leq 1$, as expected. For $\delta > 1$, increments were negligible and there was no significant difference between the POL rule and "pure" random storage (1.79897 for pure random storage). The reason follows.

For small δ, only a small portion of the rack (*effective rack area*) is utilized under the POL rule. As δ increases, the effective rack area increases accordingly, resulting in longer expected dual command cycle time. For large δ, the rack is utilized almost uniformly over the rack (Park 1987), and the POL rule performs as if it were pure random storage.

We observed that, given system demand δ, the expected dual command cycle time was convex in ω. We also observed that the optimal value of ω decreased as δ increased. The observation implies that for a system with heavy demand, it might be better to assign higher priority to locations around the center of the rack. We can say in a different way that as the activity distribution approaches the uniform distribution, we may obtain better results by more focusing on reducing travel-between times than one-way travel times. In fact, for the rack with n_x = 30, n_y = 15 and $\delta = 2$, we had the best result when $\omega = 0$.

In conclusion, for large δ ($\geq$ 1), contour line configurations did not make significant differences in system performance. For small δ, however, contour line configurations were significant. In both cases, any contour line configurations generated for large ω ($\geq$ 1) performed moderately well. This result supports the claim that the conventional square-in-time configuration is reasonable for practical use [4].

3.3 Turnover-Based Storage

Given skew parameter s, $0 < s \leq 1$, we can express the pallet turnover distribution as

$$G(y) = y^s, \quad 0 \leq y \leq 1. \tag{9}$$

Equation (9) is derived from the well known "ABC" inventory phenomenon, and represents a plot of the ranked cumulative pallet activity versus the cumulative pallet inventory. The smaller s is, the more skewed is the distribution. Note that if s = 1, the distribution becomes uniform. It is clear from Equation (9) that, for turnover-based storage, the normalized access frequency for the location with j-th priority is determined by

$$p(j) = \left(\frac{j}{N}\right)^s - \left(\frac{j\text{-}1}{N}\right)^s, \quad j = 1, 2, \cdots, N. \tag{10}$$

We considered skew parameters .569 (20%/40%), .431 (20%/50%), .317 (20%/60%), .222 (20%/70%), .139 (20%/80%), and .065 (20%/90%), where (20%/40%), as an example, implies that 20% of the pallets represents 40% of the total pallet activity. Table 2 shows expected dual command cycle times for turnover-based storage, where n_x=30 and n_y=30. A summary of the analysis for turnover-based storage follows.

Table 2. Expected dual command cycle times for turnover-based storage

		skew parameter (s)					
		0.569 (20%/40%)	0.431 (20%/50%)	0.317 (20%/60%)	0.222 (20%/70%)	0.139 (20%/80%)	0.065 (20%/90%)
ω	0.0	1.65345	1.57656	1.49053	1.39458	1.28169	1.13519
	0.5	1.55193	1.43424	1.31248	1.18655	1.04864	0.88526
	1.0	1.51883	1.36824	1.19739	1.00069	0.75839	0.44622
	2.0	1.51802	1.36465	1.18933	0.98734	0.74079	0.43109
	3.0	1.51848	1.36503	1.18932	0.98682	0.73990	0.42930
	4.0	1.51868	1.36522	1.18945	0.98677	0.73974	0.42911
	5.0	1.51888	1.36536	1.18955	0.98679	0.73967	0.42904
	10.0	1.51900	1.36541	1.18959	0.98681	0.73965	0.42902
	∞	1.51915	1.36554	1.18971	0.98686	0.73968	0.42903

For any ω, the expected dual command cycle time decreased as the skew parameter s decreased. This phenomenon is natural since, for smaller s, locations having higher priority (or, alternatively, close to the I/O point) have higher density, resulting in shorter expected dual command cycle time.

Given skew parameter s, the expected dual command cycle time was convex in ω. We also observed that the optimal value of ω increased as s decreased. This observation implies that, as the pallet turnover distribution becomes more skewed, we obtain better results by focusing on reducing one-way travel times. In general, there was no significant difference in system performance when $\omega \geq 1$. Furthermore, any contour line configurations generated for large ω (≥ 1) performed moderately well.

4. Conclusions

We developed a general scheme to generate contour line configurations for dual command operations. The contour line configurations generated were then used to develop a location indexing scheme. The location indexing scheme assigns a unique priority to each location. Location priorities are used for storage location assignments. To investigate the effect of alternative contour line configurations on system performance, we performed a series of experiments. The storage policies considered include pure random storage, COL rule, POL rule, and turnover-based storage.

The best contour line configuration approaches square-in-time as the pallet turnover distribution is skewed. The converse is also true. If the pallet turnover distribution tends to be uniform or racks are utilized more uniformly, the best contour line configuration approaches a circle. For $\omega \geq 1$, however, there is no significant difference in system performance. Lastly, the conventional square-in-time configuration performs moderately well in all cases.

References

[1] Bozer, Y. A. and White, J. A., "Travel-Time Models for Automated Storage/Retrieval Systems," *IIE Transactions*, 16(4), 329-338, 1984.

[2] Francis, R. L., "On Some Optimum Facility Design Problem," Unpublished Ph.D. Thesis, Northwestern University, 1967.

[3] Francis, R. L. and White, J. A., *Facility Layout and Location: An Analytic Approach*, Prentice-Hall, 1974.

[4] Graves, S.C., Hausman, W.H. and Schwarz, L.B., "Storage-Retrieval Interleaving in Automatic Warehousing Systems," *Management Science*, 23(9), 935-945, 1977.

[5] Harmatuck, D. J., "A Comparison of Two Approaches to Stock Locations," *The Logistics and Transportation Review*, 12(4), 282-284, 1976.

[6] Heskett, J. L., "Cube-Per-Order Index - A Key to Warehouse Stock," *Transportation & Distribution Management*, 3, 27-31, 1963.

[7] Heskett, J. L., "Putting the Cube-Per-Order Index to Work in Warehouse Layout," *Transportation & Distribution Management*, 4, 23-30, 1964.

[8] Park, B. C., "Analytical Models and Optimal Strategies for Automated Storage/Retrieval System Operations," Unpublished Ph.D. Dissertation, Georgia Institute of Technplogy, 1991.

[9] Park, B. C., "Closest Open Location Rule in AS/RS," *Journal of the Korean Institute of Industrial Engineers*, 13(2), 87-95, 1987.

98년 1월 최초 접수, 98년 10월 최종 수정

APPENDIX

Example Contour Line Configurations

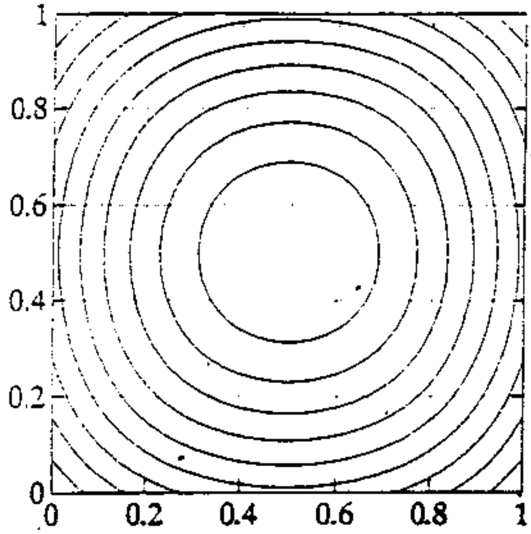

Figure 3. Contour line configuration ($b = 1, \omega = 0$)

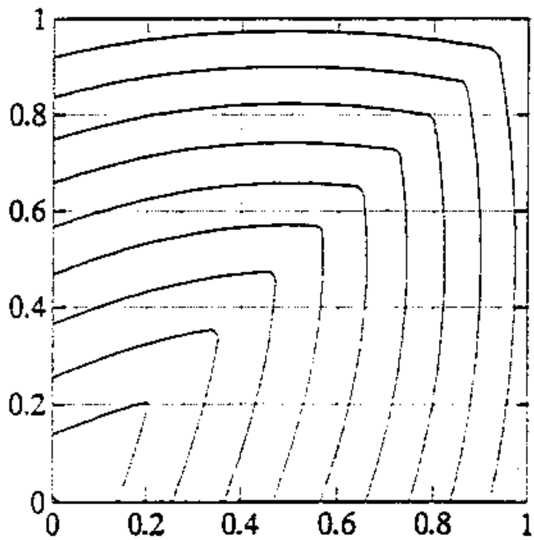

Figure 4. Contour line configuration ($b = 1, \omega = 2$)

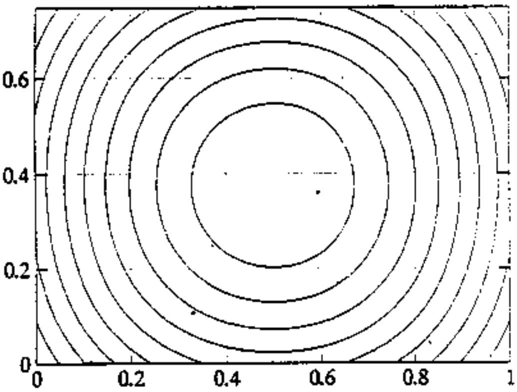

Figure 5. Contour line configuration ($b = 0.75, \omega = 0$)

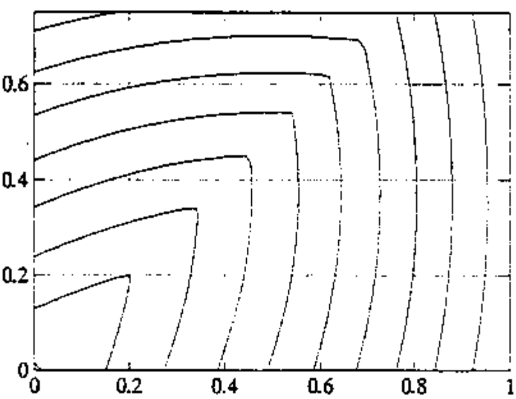

Figure 6. Contour line configuration ($b = 0.75, \omega = 2$)

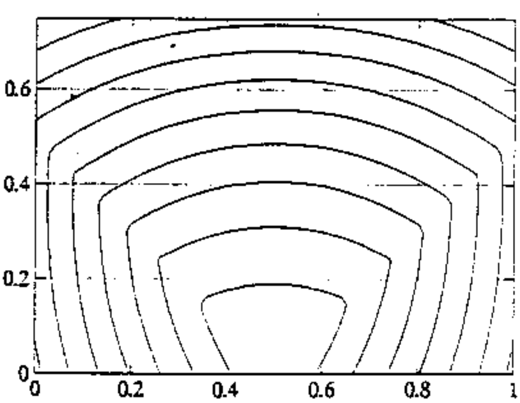

Figure 7. Contour line configuration for $I/O = (0.5, 0)$ ($b = 0.75, \omega = 1$)