Catalog Segmentation with the Objective of Satisfying Customer Requirements in Minimum Number of Catalog

Conference Paper · December 2009

DOI: 10.1109/IEEM.2009.5373003

4 authors:


Kamran Kianfar
University of Isfahan


Michel Fathi
University of North Texas


Amir Zadeh
Wright State University


Amir Sadeghi
Islamic Azad University, South Tehran Branch

Catalog Segmentation with the Objective of Satisfying Customer Requirements in Minimum Number of Catalog

Kamran Kianfar [1], Mehdi Fathi [2], Amir Hasanzadeh [3], Amir Sadeghi [4]

[1] Department of Industrial Engineering, Isfahan University of Technology, Isfahan, Iran
[2] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
[3] Department of Industrial Engineering, AmirKabir University of Technology, Tehran, Iran
[4] Department of Industrial Engineering ,Tehran Payamenoor University, Tehran, Iran
(mfathi@iust.ac.ir)

Abstract - **This work is concerned with customer-oriented catalog segmentation that each catalog consists of specific number of products. In this problem, requirements of a specific ratio of customers should be satisfied. According to the definition, when a customer is satisfied that at least *t* required products exist in his/her catalog. The objective of this problem is to minimize the number of catalogs, regarding to minimum number of customers constraint that was comply. In this paper, we present a mixed-integer programming model for this clustering problem. This problem is NP-Hard in large scales and the optimum solution is almost impossible to reach. Hence, a solution procedure is developed based on genetic algorithm. Then, the results of computational experiments are reported, in which the GA solution is compared with exact solution of mixed-integer programming model.**

Keywords - **Catalog Segmentation, Customer Clustering, Mixed-Integer Programming, Genetic Algorithm**

I. INTRODUCTION

In today's global competition, one of the business and marketing organizations goals is customer attraction. Catalog design has known as the most common way in customer attraction, satisfaction, and retention cycle of customer relationship management. Meanwhile one of the challenges that companies using catalogs face is the optimization problem of available products; this means which products are covered by the catalog and for which cluster of customers the catalog is designed. The catalog optimization has an effective role in satisfying customers' requirements and company profitability.

With the fast growing number of products offered by retailers and e-retailers alike, the design of one catalog that can contain all products is not feasible. In many cases, some customers are interested in only a small portion of the products the company carries. Nowadays, the catalog design has changed as one of the competitive tools for marketing among companies. The latest Benchmark Survey on Critical Issues and Trends conducted by Catalog Age magazine in 2003 indicates that catalog companies that participated in the survey spent a mean 26.1% of their revenue on marketing expenses and that print and postage costs account for nearly half of those expenses. The number of catalogs that companies send to customers and potential buyers is increasing at a fast pace [1]. Efficiency and profitability of one catalog is evaluated by the number of available products of interest to customers that companies send to them. The companies seek optimum decision making on the customers clusters, not optimum decision making as individual. In this problem, the need of a customer is satisfied if at least the specified minimum number of products of interest to him/her is included in one of the catalogs.

The classical catalog segmentation problem was introduced by Kleinberg et al. [8]. This problem consists of designing *K* catalogs, each of size *r* products that maximize the number of covered customers.

In general, the customer-oriented catalog segmentation problem is a customer clustering problem where:

- The task is to determine *K* clusters of customers Where each cluster is defined by a set of products of interest to the customers and each customer is assigned to the cluster (equivalent to a catalog) that contains the largest number of products of interest to him/her.
- An acceptable clustering should satisfy two constraints:
  1) Each cluster includes *r* products,
  2) The need of a customer can be satisfied (i.e., covered) only by a catalog which contains at least *k* products of interest to the customer.

The remainder of the paper is organized as follows. In section 2, we give the definition and formulation of the problem. In section 3, heuristic technique of Genetic Algorithm is represented briefly. In section 4, circumstance of applying this technique for solving the catalog segmentation problem is studied. In the following section, a numerical example is used for better understanding of the problem. Finally, section 6 is addressed to present a general conclusion of the paper.

II. Problem definition

Assume that customers and their product interests are known. The main input data for catalog segmentation is the customer interest database that contains the set of products that each customer is interested in. The product interests of a customer can be obtained either by

978-1-4244-4870-8/09/$26.00 ©2009 IEEE

aggregating all purchase transactions of the customer or by obtaining his/her explicit preferences from a set of products. The customer-oriented catalog segmentation problem consists of designing some catalogs, each of size *r* products that minimize the number of catalogs and the requirements of a specific ratio of customers should be satisfied.

A. Model Assumptions

1) The number of catalogs is not predefined.
2) Each catalog includes *r* products.
3) At least requirements of a specific ratio of customers should be satisfied.
4) Set of needed products of each customer is not consistent and predetermined.

B. Sets and indices

Set of customers	$k \in N$	$k = \{1,...,N\}$
Set of products	$i \in M$	$M = \{1,...,M\}$
Set of catalogs	$j \in T$	$T = \{1,...,T\}$

C. Model parameters

Parameter's notation	Description
r	*Number of available products on each catalog.*
α	*Ratio of product number of interest to each customer that should be covered in his/her catalog.*
a_{ki}	*1 if customer k is interested in product i, otherwise 0.*

D. Decision variables

Variable	Description
S_j	*1 if catalog k is chosen, otherwise 0.*
V_{kj}	*1 if need of customer k is covered by catalog j, otherwise 0.*
Y_{ji}	*1 if catalog j includes product i, otherwise 0.*

E. Mathematical model

The mathematical programming model is as follows:

$$Min \sum_{j\in T} S_j \qquad (1)$$

$$s.t.$$

$$(\alpha \sum_{i\in M} a_{ki}) V_{kj} \le \sum_{i\in M} a_{ki} . Y_{ji} \qquad \forall k \in N, j \in T \qquad (2)$$

$$\sum_{i\in M} Y_{ji} = r \qquad \forall j \in T \qquad (3)$$

$$\sum_{j\in T} V_{kj} \ge 1 \qquad \forall k \in N \qquad (4)$$

$$V_{kj} \le S_j \qquad \forall k \in N, j \in T \qquad (5)$$

$$S_j, Y_{ji}, V_{kj} \in \{0,1\} \qquad \forall k \in N, j \in T, i \in M \qquad (6)$$

The existing relations in the model can be described as follows:
Objective function (1) minimizes the number of the catalogs. Constraints (2) present percentage of the needs to each customer that should be satisfied by related catalog. Constraints (3) ensure that each catalog includes *r* products. Constraints (4) guarantee that the needs of each customer should be satisfied by at least one catalog. Constraints (5) state that if the customer need is satisfied by a catalog, this catalog needs to be chosen. Constraints (6) impose the binary nature of decision variables' sets.

III. Genetic Algorithm application in solving the proposed model

According to complexity of the proposed model and NP-Hard nature of the catalog segmentation optimization problem, solution procedure of the above model in large-sizes requires long run time and actually it's not possible on *PC*. Consequently Metaheuristic Algorithms can be used for model behavior simulation. However, for validating each heuristic approach in large-size, firstly it's necessary that results of the model and the heuristic approach in small-size to be compared. This task has been performed in computational results section. Considering the different aspects of the problem, we have chosen Genetic Algorithm (GA) which is known to be one of the most efficient heuristic methods in literature. Since GA always represents a set of feasible solutions, it is able to treat a greater part of the feasible region in a short period of time and (relatively) quickly creates high quality solutions to a problem.

Genetic Algorithm is a family of computational models based on principles of evolution and natural selection. These algorithms convert the problem in a specific domain into a model by using a chromosome-like data structure and evolve the chromosomes using selection, recombination, and mutation operators.
The genetic algorithms are typically characterized by the following aspects:

- The GA work with the base in the code of the variables group (artificial genetic strings) and not with the variables in themselves.
- The GA work with a set of potential solutions (population) instead of trying to improve a single solution.
- The GA do not use information obtained directly from the objective function, of its derivatives, or of any other auxiliary knowledge of the same one.
- The GA applies probabilistic transition rules, not deterministic rules.

- The GA stopping condition is defined based on the maximum number of generations or run time criteria.

Here three operators and selection circumstance of parents (chromosomes' current generation) for creating children (chromosomes' next generation) are introduced in brief.

Selection

While there are many different types of selection, we will cover the most common type - roulette wheel selection. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are then chosen randomly based on these probabilities and produce offspring.

Crossover

Crossover operator acts on two parents and creates two children. While there are many different kinds of crossover; the most common type is single point crossover. In single point crossover, you choose a locus at which you swap the remaining alleles from on parent to the other. This is complex and is best understood visually. Procedure of Crossover operator is shown as fig. 1.

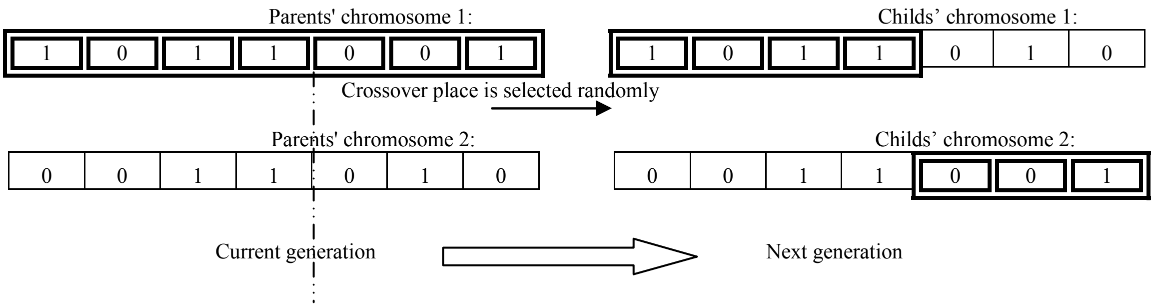

Fig. 1. Single point crossover operator procedure

Mutation

After selection and crossover, you now have a new population full of individuals. Some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not all exactly the same, you allow for a small chance of mutation. You loop through all the alleles of all the individuals, and if that allele is selected for mutation, you can either change it by a small amount or replace it with a new value. The probability of mutation is usually between 1 and 2 tenths of a percent. As a case in point in fig. 2, the value of parents' chromosome has been changed in gene 3th.

100101 ⇨ 101101

Fig. 2. Mutation operator procedure

Reproduction

Reproduction operator transfers a copy of the best parents' chromosome to next generation without any changes.

IV. Proposed Genetic Algorithm

Different components of our proposed Genetic Algorithm are explained in the following sections:

A. Chromosomal representation

Each chromosome is meant to represent a feasible solution of the problem. In our devised scheme, every catalog is indeed a chromosome. If a catalog is going to contain product *i* then the gene located in the *i*th position of the corresponding chromosome is set to 1; otherwise, it is set to 0. It is important to note that since the number of products in each catalog is known and predetermined, we will always have a fixed number of "1"s in the corresponding chromosome. As an example, the string in fig. 3 is representing a catalog which contains products 2,3,6 and 9 out of ten available products.

0	1	1	0	0	1	0	0	1	0

Fig. 3. The string related to catalog containing products 2-3-6-9

B. Generating initial solution

Firstly, our proposed algorithm assigns a string to each customer and sets its elements to either 0 or 1 at random. In order to maintain the solutions feasible, the number of '1's in each string must be equal to the number of required products in the corresponding catalog. Then, in each run of the algorithm two random strings are chosen and two new strings are produced using one-point crossover operator and this process is performed several times. Based on what was explained above, the algorithm checks and sets the number of '1's in every string. When producing the appropriate chromosomes is finished, it is time to identify the customers whose needs are met by each catalog. Accomplishing this task enables us to determine the minimum number of catalogs necessary to satisfy the customers' needs. It must be mentioned that since setting the number of '1's in each chromosome requires random changes in the values of some genes, there will be no need to the mutation operator.

C. *Selection*

In order to achieve an approximate solution, the minimum number of catalogs required to satisfy the needs of all customers, a new generation of solutions must be produced. This new generation must be formed in a way that the new solutions have better fitness values. Thus, after producing each generation, a number of existing chromosomes must be selected from mating pool to produce next generation. It is essential to select good parents for generating offspring. Good parents in the current population are the chromosomes that satisfy the needs of more customers.

D. Stopping criteria

The stopping criteria for our developed algorithm are:

- Maximum number of generations: our algorithm stops when the number of generations exceeds a predetermined value.
- Run time criteria: The algorithm will stop if

[(The time when best solution was obtained)-(the time when solving process began)] > [a predetermined value]

The summary of the other parameter settings related to our proposed GA is given in Table 1.

Table 1
Parameter settings of GA

Population size	25
Crossover type	Two point
Reproduction fraction	0.1
Mutation rate	0.002
Stopping criteria	Maximum 500 generations or 1 hour run time

V. Numerical Example

In this section, results of solving the proposed model by *Lingo 8* optimization software and Genetic Algorithm are compared and analyzed. To solve the above model, a *PC* with P4-2.4 GHz processor has been used and GA also has been developed by *Visual Basic.* Furthermore, *Lingo 8* optimization software has been used for verifying proposed model and its solution procedure. In table 2, results of solution procedure of optimization model and solution procedure of GA have been compared for three small-size cases.

For instance, the first case has been considered for 10 customers and 7 products where given number of available products in each catalog is 6 ($r = 6$) and the ratio of product number of interest to each customer that should be covered in his/her catalog is %75 ($\alpha = 0.75$) . The products of interest to each customer related to case 1 has been shown as Table 3.

After solving the model, final solution has been included 2 catalogs that the first catalog contains products {3,4,5,6,7,10} and the second catalog contains products {1,2,4,7,9,10}.

Table 2
Comparison of GA and Lingo results for small-size cases

Case	Size ($N*M$)	Number of catalogs			Run time (h:mm:ss)	
		Lingo	GA	Error	Lingo	GA
1	10*7	1	1	%0	0:0:06	0:0:03
2	15*10	2	2	%0	0:02:15	0:0:14
3	20*15	4	4	%0	0:18:54	0:02:35
4	25*20	10	10	%0	0:42:11	0:06:72
5	30*25	18	19	%5.5	1:12:28	0:09:54

In the above table, M and N respectively show number of customers and number of products. It's noted that in each three cases, the number of available products in the catalogs is considered to be 6.

Table 3
The products of interest to each customer

Customer	Interest products	Customer	Interest products
1	1,10	9	1,4,6,7,9
2	2,4	10	2
3	1,3,4,5,7	11	4,9

4	1,2,3,4,6,7,9,10	12	1,2,5,7
5	2,9	13	4,5,10
6	1,2,4,6,9	14	5,6,7
7	3,5	15	1,2,3,7
8	5,6,7,8		

Minimum number of the catalogs resulted by GA procedure and lingo for large-size problems as well as related run times show as table 4.

Table 4
Results of GA and lingo solutions

Case	Size ($N*M$)	Number of products in each catalog	Number of catalogs related to GA solution procedure	GA run time (h:mm:ss)	Number of catalogs resulted by Lingo	Lingo run time (h:mm:ss)	Error percentage
1	50*20	8	18	0:04:11	19	2:49:58	% 5.5
2	70*25	10	20	0:09:18	20	5:28:03	% 0
3	100*30	10	23	0:11:17	23	9:32:44	% 0
4	130*40	10	27	0:19:53	28	12:00:00*	% 3.7
5	150*60	15	31	0:25:26	33	12:00:00*	% 6.5
6	200*80	20	37	0:38:01	38	12:00:00*	% 2.7
7	299*100	20	41	0:42:44	44	12:00:00*	% 7.3

*Lower bound achieved by lingo after about 12 hours

As it is seen in the table 4, the error of GA procedure in large-size problems is less than 8 percent while the run times related to the GA algorithm are much less than the time needed for lingo software (in average %5) .

VI. CONCLUSION

In this paper the catalog segmentation problem with the objective of satisfying the customer requirements in minimum number of the catalogs was studied. Firstly, we developed a mixed-integer programming model. Then this model was coded in *Lingo 8* optimization software and was solved for small-size problems. Due to complexity of the larger problems, hence longer run time of exact solution, GA algorithm was used for solution procedure. Comparison between the results of the GA and *Lingo*, for small and large-size problems, showed that the designed genetic algorithm was reached to good solutions in acceptable run time and so we could trust the GA for the other problems.

REFERENCES

[1] S. Chiger, "Benchmark survey on critical issues and trends,Catalog", Age 20 (13), 2003, pp. 32–37.

[2] A. Amiri, "Customer-oriented catalog segmentation: Effective solution approaches", Decision Support Systems 42, 2006, pp. 1860–1871

[3] M. Ester, R. Ge, W. Jin, Z. Hu, "A Microeconomic Data Mining Problem: Customer-Oriented Catalog Segmentation", Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining (Seattle, Washington), 2004, pp. 557–562.

[4] M. Turkey, "A mixed-integer programming approach to the clustering problem with an application in customer segmentation", European Journal of Operational Research 173 , 2006, pp.866–879

[5] M.R. Rao, "Cluster analysis and mathematical programming", *Journal of the American Statistical Association* 66, 1971, pp. 622–626.

[6] P.S. Bradley, U.M. Fayyad, O.L. Mangasarian, "Mathematical programming for data mining: Formulations and challenges", Journal on Computing 11, 1999, pp. 217–238.

[7] C. Rygielski, J.C. Wang, D.C. Yen, "Data mining techniques for customer relationship management", Technology in Society 24, 2002, pp. 483–502.

[8] J. Kleinberg, C. Papadimitriou, P. Raghavan, "Segmentation problems", Proceedings of the Thirtieth Annual ACM Symposiumon Theory of Computing (Dallas, Texas), 1998, pp. 473–482.

[9] F. Glover, G. A. Kochenbereger, editors, *Handbook of Metaheuristics* ,Kluwer academic publishers, Boston, Dordrecht, London, 2003.