ISSN:1738-6764, eISSN:2093-7504

A Genetic Algorithm for Solving a QFD(Quality Function Deployment) Optimization Problem

Jaewook Yoo [1,*]

[1] Department of Business Administration, Dong-A University, Republic of Korea; Associate Professor; jyoo@dau.ac.kr
* Correspondence

https://doi.org/10.5392/IJoC.2020.16.4.026

Manuscript Received 11 August 2020; Received 28 September 2020; Accepted 6 October 2020

Abstract: *Determining the optimal levels of the technical attributes (TAs) of a product to achieve a high level of customer satisfaction is the main activity in the planning process for quality function deployment (QFD). In real applications, the number of customer requirements for developing a single product is quite large, and the number of converted TAs is also high so the size of the house of quality (HoQ) becomes huge. Furthermore, the TA levels are often discrete instead of continuous and the product market can be divided into several market segments corresponding to the number of HoQ, which also unacceptably increases the size of the QFD optimization problem and the time spent on making decisions. This paper proposed a genetic algorithm (GA) solution approach to finding the optimum set of TAs in QFD in the above situation. A numerical example is provided for illustrating the proposed approach. To assess the computational performance of the GA, tests were performed on problems of various sizes using a fractional factorial design.*

Keywords: Quality function deployment; Customer requirements; Technical attributes; Multi-segment market; Genetic algorithm

1. Introduction

Quality function deployment (QFD) is a widely adopted customer-oriented product development methodology by analyzing customer requirements (CRs) [1]. The essential concept of QFD is to use a series of charts called the houses of quality (HoQ) to convert CRs into technical attributes (TAs) and in this way into parts characteristics, process plans, and manufacturing operations [2]. An HoQ typically contains information about the relationship between CRs and TAs, as well as TAs and benchmarking data [3]. Based on the information contained in the HoQ, determining the optimal levels of the TAs of the product for achieving a high level of customer satisfaction is the main activity of the QFD planning process.

In QFD studies, the values of TAs are presumed to be continuous while often taken as discrete within real applications, meaning that each TA has a few alternatives [4, 5]. What engineers need to do is to select the best one among all the possible alternatives of TAs. Besides, it is relatively easier to assign a single value to customer satisfaction and associated costs for each alternative of a TA than to clarify the precise relationships between them. Experienced engineers can determine the cost for achieving a certain degree of requirement on a TA, based on their knowledge. In the specified degree of a TA, it is then possible to determine the degree of customer satisfaction for one aspect of a CR. The optimization approach is then used to find a way to obtain the optimum set of TAs from a limited number of their alternatives [4, 6].

Another of the assumptions regarding most of QFD optimization problems is that the heterogeneous customer requirements in a market can be generalized and thus only one HoQ is used to link to CRs. However, customers who have different beliefs about social issues (e.g. religion, politics, labor, drugs, women’s rights) or personal interests (e.g. family, home, work, food, self-fulfillment, health, clubs, friends, shopping) may have different purchase behaviors or preferences [7, 8]. As a result, customers on a product market may have different reactions to a product. Therefore, a product market can be divided into several market segments, each containing a group of customers with homogeneous preferences. HoQs should be developed as many market segments as divided.

On the other hand, in real applications, the number of CRs for developing a single product is quite large, and the number of converted TAs is also high so that the size of one HoQ becomes huge [8, 9]. Furthermore, the number of alternatives each TA has and the number of market segments corresponding to the number of HoQs mentioned above also greatly increase the size of one QFD optimization problem and spend a long time on making decisions unacceptably. The QFD optimization problem with such a large size and complexity makes the existing solution approaches developed for small or medium size QFD optimization problems inapplicable. Therefore, a solution approach for efficiently solving the QFD optimization problem in the above-mentioned situation needs to be developed.

In this study, a genetic algorithm (GA) is proposed to solve such a large size QFD optimization problem for determining optimal discrete values of TAs on the multi-segment market in QFD which is more realistic in practice. To assess the computational performance of the proposed GA, a fractional factorial design is applied for the designs of experimentation, and the time spent on solving the various size problems and the quality of the optimal solutions are measured using a model and example extended from [4, 10].

The amount of the literature regarding QFD is so vast that the scope of the literature review in this paper specifically focuses on determining the optimal levels of TAs in QFD. Meanwhile, there are a few useful literature reviews of general applications of QFD [11-16].

Linear programming is a well-known method that has been used to find the best set of TAs. In general, this model is used to allocate resources to different TAs to maximize the overall customer satisfaction (OCS) [6, 17-22]. In these studies, it is assumed that the values of TAs may be at any point in a continuous range whereas in real applications they are often considered discrete. Lai et al. (2005) built a model for the QFD optimization problem with the discrete TAs values and developed a dynamic programming solution approach to solve the model [4]. Delice & Güngör (2011) also proposed a mixed-integer linear programming model combining with multi-objective decision making for the QFD optimization problem with the discrete TAs values [5]. Integer programming is suggested for product design optimization with a modified HoQ prioritization procedure using a multi-attribute decision method for assigning relationship ratings between CRs and TAs [23]. Non-linear models are also developed to solve the QFD optimization problems by incorporating realistic cost functions [24], under consideration of product lifecycle factors and resource constraints [25], and based on a fuzzy regression approach to model functional relationships between CRs and TAs, and among TAs in the inherent fuzziness [26], respectively.

There appear to be few studies on QFD optimization under the multi-segment market as well as a huge and complex QFD matrix. Luo et al. (2010) developed a methodology that includes a customer survey, fuzzy clustering, QFD, and fuzzy optimization to achieve the optimum target settings for the TAs of a new product on a multi-segment market [27]. Yoo (2015) proposed a dynamic programming methodology to find the optimal discrete levels of TAs under a multi-segment market in QFD [10]. An improved algorithm, a combination of imperialist competitive algorithm and GA, is proposed as a solution approach for efficiently solving a huge and complex QFD matrix [28].

The rest of the paper is organized as follows. Section 2 introduces both the model and the GA approach. In Section 3, a numerical illustration is appeared to demonstrate the proposed approach. Section 4 shows the results of the computational experiments and provides a discussion of relevant aspects. Finally, conclusions are drawn in Section 5.

2. Materials and Methods

2.1 Model

A model extended from [4, 10] is introduced in this section. Suppose a product has I CRs and J TAs, and market segments exist in T. It is also assumed that one segment of the market corresponds to one HoQ. T HoQs should be created. According to the characteristics of each market segment, CRs, and their weights, the relationship between each TA and other TAs, the relationship between each CR and the TAs required to implement it, and benchmarking scores for competitors' products and their existing products are included in each HoQ.

For market segment t, the relative importance of CR i is obtained from the other CRs, w_{it}, which is the scaled weight of the importance of CR i ($0 \leq w_{it} \leq 1$ and $\sum_{i=1}^{I} w_{it} = 1$), and the relationship between CR i and TA j, r_{ijt} ($0 \leq r_{ijt} \leq 1$ and $\sum_{j=1}^{J} r_{ijt} = 1$). Wasserman (1993) proposed a useful approach to normalizing the relationship matrix considering the interrelationships between the TAs [17]. In this paper, the relationship matrix is assumed to have already been normalized.

To reflect it into the traditional HoQ that the levels of individual TAs can be discrete, which means each TA has a few alternatives, we need to incorporate some additional information into the HoQ. We add to the traditional HoQ the alternatives for each TA and the corresponding customer satisfaction information. As a result, the extended HoQ is shown in Fig. 1. The remaining parts of the HoQ remain the same.

It is assumed in this HoQ that TA 1 has *a* alternatives, TA 2 has *b* alternatives, … , TA *j* has *p* alternatives, … , TA *J* has *q* alternatives. TA_{jkt} (j = 1, 2, … , J; k = 1, 2, . . . , K; t = 1, 2, … , T) refers to alternative k of TA j in market segment t. Cr_{ijkt} means the customer satisfaction level (CSL) of CR i acquired by TA_{jkt}.

Then the information relating to costs can be summarized as in Table 1. For market segment t, C_{jkt} is the cost of TA_{jkt} and CR_{jkt} means customer satisfaction achieved by TA_{jkt}. Assume that customer satisfaction, CR_{jkt}, is the weighted sum of each customer satisfaction of each CR obtained in market segment t by TA_{jkt}. Then, CR_{jkt} is calculated from the following formula:

$$CR_{jkt} = \sum_{i=1}^{I} w_{it}\, Cr_{ijkt} \tag{1}$$

where I is the number of CRs.

Table 1. Cost information for market segment *t*

TA	Cost	CSL	TA	Cost	CSL	…	TA	Cost	CSL	…	TA	Cost	CSL
TA_{11t}	C_{11t}	CR_{11t}	TA_{21t}	C_{21t}	CR_{21t}	…	TA_{j1t}	C_{j1t}	CR_{j1t}	…	TA_{J1t}	C_{J1t}	CR_{J1t}
…	…	…	…	…	…	…	…	…	…	…	…	…	…
TA_{1at}	C_{1at}	CR_{1at}	TA_{2bt}	C_{2bt}	CR_{2bt}	…	TA_{jpt}	C_{jpt}	CR_{jpt}	…	TA_{Jqt}	C_{Jqt}	CR_{Jqt}

		Technical attribute 1		**Technical attribute 2**		…	**Technical attribute *j***		…	**Technical attribute *J***	
Customer requirement 1	w_{1t}	r_{11t}		$r12t$		…	r_{1jt}		…	r_{1Jt}	
		TA_{11t}	Cr_{111t}	$TA21t$	$Cr121t$		TA_{j1t}	Cr_{1j1t}		TA_{J1t}	Cr_{1J1t}
		…	…	…	…		…	…		…	…
		TA_{1at}	Cr_{11at}	TA_{2bt}	Cr_{12bt}		TA_{jpt}	Cr_{1jpt}		TA_{Jqt}	Cr_{1Jqt}
Customer requirement 2	w_{2t}	r_{21t}		r_{22t}		…	r_{2jt}		…	r_{2Jt}	
		TA_{11t}	Cr_{211t}	TA_{21t}	Cr_{221t}		TA_{j1t}	Cr_{2j1t}		TA_{J1t}	Cr_{2J1t}
		…	…	…	…		…	…		…	…
		TA_{1at}	Cr_{21at}	TA_{2bt}	Cr_{22bt}		TA_{jpt}	Cr_{2jpt}		TA_{Jqt}	Cr_{2Jqt}
…	…	…		…		…	…		…	…	
Customer requirement i	w_{it}	r_{i1t}		r_{i2t}		…	r_{ijt}		…	r_{iJt}	
		TA_{11t}	Cr_{i11t}	TA_{21t}	Cr_{i21t}		TA_{j1t}	Cr_{ij1t}		TA_{J1t}	Cr_{iJ1t}
		…	…	…	…		…	…		…	…
		TA_{1at}	Cr_{i1at}	TA_{2bt}	Cr_{i2bt}		TA_{jpt}	Cr_{ijpt}		TA_{Jqt}	Cr_{iJqt}
…	…	…		…		…	…		…	…	
Customer requirement I	w_{It}	r_{I1t}		r_{I2t}		…	r_{Ijt}		…	r_{IJt}	
		TA_{11t}	Cr_{I11t}	TA_{21t}	Cr_{I21t}		TA_{j1t}	Cr_{Ij1t}		TA_{J1t}	Cr_{IJ1t}
		…	…	…	…		…	…		…	…
		TA_{1at}	Cr_{I1at}	TA_{2bt}	Cr_{I2bt}		TA_{jpt}	Cr_{Ijpt}		TA_{Jqt}	Cr_{IJqt}

Figure 1. The extended HoQ for market segment

Assuming that the OCS of the whole market is the weighted sum of the customer satisfaction of the individual market segments, the objective function of this optimization problem can be developed as follows

$$\text{OCS} = \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{k=1}^{K} \xi_t CR_{jkt} x_{jkt}$$

where x_{jkt} is equal to 1 if alternative k of TA j in market segment t, TA_{jkt}, is selected, and 0 otherwise. CR_{jkt} is defined as (1), and ξ_t is the normalized weight of the importance of market segment t ($0 \le \xi_t \le 1$ and $\sum_{t=1}^{T} \xi_t = 1$).

If the number of customers in market segment *t* is estimated based on the historical sales data of the firm and the salesmen's knowledge, ξ_t can be obtained as

$$\xi_t = {q_t}/{\sum_{t=1}^{T} q_t}$$

where q_t is the estimated number of customers in market segment t.

The problem of selecting a set of TA alternatives for each segment of the multi-segment market to maximize the OCS of the multi-segment market while not exceeding the budget available for the multi-segment market can be formulated as a multiple-choice 0-1 knapsack problem, Problem (*P*).

Problem (*P*)

$$\max \text{OCS} = \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} \xi_t CR_{jkt} x_{jkt} \tag{2}$$

$$\text{s. t.} \quad \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} C_{jkt} x_{jkt} \le B \tag{3}$$

$$\sum_{k=1}^{K} x_{jkt} = 1 \qquad \text{for all } j, t \tag{4}$$

$$x_{jkt} \in \{0,1\} \qquad \text{for all } j, k, t \tag{5}$$

In the formulation of Problem (*P*) the objective function (2) maximizes the OCS in the multi-segment market; the budget constraint (3) indicates that the capital consumption by the alternatives selected cannot exceed the multi-segment market budget available; in any market segment, the alternative selection constraint set (4) forces the problem to select one and only one alternative for each TA; and the constraint set (5) imposes the integrality of the decision variables.

2.2 Genetic Algorithm

GA approach is applied to solve problem (*P*) of determining the optimal levels of TAs in QFD under a multi-segment market to maximize the OCS of the multi-segment market. The description of each step of the proposed GA approach in this paper is as follows.

Initialization: Genetic Representation

A chromosome can be defined to have $J \times T$ genes corresponding to *J* TAs for *T* market segments. A gene takes an integer number from a set {1,2,….,*K*} which means alternatives for a TA in a market segment. The position of a gene is used to represent a TA in a market segment and the value of a gene is used to represent an alternative selected for a TA in a market segment. Then, y_{jt}, an indicator variable, is defined as follows:

$$y_{jt} = k \qquad \text{if } x_{jkt} = 1, \quad j=1,2,\ldots,J;\ k=1,2,\ldots,K;\ t=1,2,\ldots,T$$

For Problem (*P*), this permutation encoding is effective since it always meets constraint (4).

Step 1: Initial Population

The initial population with the size being fixed at a certain number is randomly generated.

Step 2: Fitness Evaluation

Each chromosome of the initial population obtained in Step 1, as well as each chromosome created by the genetic operators in Step 5 and 6, are evaluated to give some measure of their fitness. Objective function (2) in Problem (*P*) is used as the measure of the fitness of each chromosome.

Genetic operators in Step 5 and 6 used to manipulate the chromosomes often yield infeasible offspring. In this research, the penalty technique is proposed to handle infeasible offspring [29].

This technique turns a constrained problem into an unconstrained problem by penalizing infeasible solutions, in which a penalty term is applied to the objective function for any violation of the constraints. An evaluation function with a penalty term is proposed as follows:

$$\text{eval}(X^i) = f(X^i) + p(X^i) \qquad \text{i=1,2,…,population size}$$

where X^i represents chromosome i, $f(X^i)$ the objective function of Problem (*P*) and $p(X^i)$ the penalty term of X^i.

For Problem (*P*), the penalty term is expressed as follows:

$$p(X^i) = \begin{cases} 0, & if\ constraint(3) is\ satisfied \\ \alpha\left(B - \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} c_{jkt}x_{jkt}^{i}\right), & \text{otherwise} \end{cases}$$

where α is a positive penalty value.

Let $|p(X)|_{max}$ and $|f(X)|_{min}$ be the maximum of $|p(X)|$ and the minimum of $|f(X)|$ in the current population, respectively. It is also required that $|p(X)|_{max} \leq |f(X)|_{min}$ to avoid negative fitness value.

Step 3: Termination

If the population converges when either a fixed % of the chromosomes in the population have the same fitness value or the number of generations reaches a fixed number, then the GA is terminated.

Step 4: Selection

The roulette wheel selection and elitist approach are combined as a selection approach. Roulette wheel selection is a method to reproduce a new generation proportional to the fitness of each individual, and the elitist method is used to preserve the best next-generation chromosome and to overcome stochastic sampling errors.

Step 5: Crossover

After performing the selection procedure in Step 4, the population sorted by descending fitness values of individuals is bisected. From among the members of the first half of the population that have a higher degree of fitness in order than the other half of the population, the first member who has the highest degree of fitness in the first half of the population is selected, and another member is randomly selected from the whole population to be a pair. As the second pair, the second member with the second highest degree of fitness is selected from the first half of the population and another member is randomly selected from the whole population. Repeat this *N*/2 (half size of the population) times. Then, a uniform crossover is accomplished for each pair by replacing the gene values for two genes on the same position which are different from each other with random integers generated from $\{1,2,\ldots,K\}$.

Step 6: Mutation

Uniform mutation is performed for each pair in the current population. Generating $2n$ (n is the size of a child chromosome) random numbers in [0, 1], we select genes for the mutation if the random number randomly assigned to each of them is less than the rate of mutation which is set to $1/2n$. The value for the selected gene is replaced by a random integer generated from $\{1,2,\ldots,K\}$. Then go to Step 3.

3. An Illustrative Example

An example extended from [4, 10] is used to illustrate the application of the proposed GA approach in this research.

The problem for the application is to determine in the two market segments the optimum levels of the TAs of a washing machine according to the CRs. Five CRs for the two market segments are identified as being the biggest concern of washing machine customers. They include “thorough washing”, “quiet washing”, “thorough rinsing”, “less damage to clothes” and “short washing time”. Five TAs, which are "washing quality (%)", "noise level (dB)", "washing time (min)", "rinsing quality (%)", and "clothing damage rate (%)", are also identified from the engineer's point of view of washing machine design. The relationship between CRs and TAs as well as the relative importance of CRs for market segment 1 and 2 are shown in the HoQ template in Table 2 and

Table 3, respectively. Since we are not focusing on the information about competitive analysis and the interrelationship between TAs, these are not shown in Table 2 and Table 3. In this example, each TA has three alternatives.

Table 2. The HoQ for Market Segment 1

		Washing Quality (%)		**Noise Level (dB)**		**Washing Time (min)**		**Rinsing Quality (%)**		**Clothes Damage Rate (%)**	
		Level	**Satisfaction level**	**Level**	**Satisfaction level**	**Level**	**Satisfaction level**	**Level**	**Satisfaction level**	**Level**	**Satisfaction level**
Thorough washing	0.313	0.3125		0		0.0625		0.3125		0.3125	
		90	0.65	45	0	30	0.8	95	1	0.5	0.8
		95	0.85	50	0	35	0.9	90	0.7	0.7	0.9
		98	1	60	0	40	1	80	0.4	1	1
Quiet Washing	0.25	0.3		0.5		0.1		0.1		0	
		90	1	45	1	30	1	95	0.85	0.5	0
		95	0.8	50	0.7	35	0.9	90	0.9	0.7	0
		98	0.7	60	0.4	40	0.6	80	1	1	0
Thorough rinsing	0.188	0.3		0		0.1		0.5		0.1	
		90	0.5	45	0	30	1	95	1	0.5	1
		95	0.9	50	0	35	0.6	90	0.8	0.7	0.9
		98	1	60	0	40	0.5	80	0.4	1	0.8
Less damage to clothes	0.125	0.231		0.077		0.077		0.231		0.384	
		90	1	45	1	30	1	95	1	0.5	1
		95	0.8	50	0.9	35	0.9	90	0.6	0.7	0.8
		98	0.7	60	0.9	40	0.8	80	0.5	1	0.5
Short washing time	0.125	0.714		0		0.143		0.143		0	
		90	0.7	45	0	30	1	95	0.6	0.5	0
		95	0.9	50	0	35	0.8	90	0.8	0.7	0
		98	1	60	0	40	0.6	80	1	1	0

Table 3. The HoQ for Market Segment 2

		Washing Quality (%)		**Noise Level (dB)**		**Washing Time (min)**		**Rinsing Quality (%)**		**Clothes Damage Rate (%)**	
		Level	**Satisfaction level**	**Level**	**Satisfaction level**	**Level**	**Satisfaction level**	**Level**	**Satisfaction level**	**Level**	**Satisfaction level**
Thorough washing	0.3265	0.2875		0		0.1712		0.258		0.2833	
		92	0.7	54	0	39	1	81	0.7	1	1
		94	0.8	50	0	36	0.9	83	0.8	0.8	0.8
		96	1	46	0	33	0.8	85	1	0.6	0.6
Quiet Washing	0.0067	0		1		0		0		0	
		92	1	54	0.5	39	0	81	0	1	0
		94	0.85	50	0.7	36	0	83	0	0.8	0
		96	0.75	46	1	33	0	85	0	0.6	0
Thorough rinsing	0.2237	0.285		0		0.1828		0.2849		0.2738	
		92	0.6	54	0	39	1	81	0.5	1	0.8
		94	0.8	50	0	36	0.9	83	0.7	0.8	0.9
		96	1	46	0	33	0.8	85	1	0.6	1
Less damage to clothes	0.4156	0.2688		0		0.1495		0.2688		0.3129	
		92	1	54	0	39	0.6	81	0.5	1	0.5
		94	0.9	50	0	36	0.8	83	0.6	0.8	0.7
		96	0.7	46	0	33	1	85	1	0.6	1
Short washing time	0.0275	0.2152		0		0.3119		0.2654		0.2165	
		92	0.8	54	0	39	0.5	81	1	1	0.7
		94	0.9	50	0	36	0.7	83	0.8	0.8	0.8
		96	1	46	0	33	1	85	0.6	0.6	1

We also need the TA alternatives-related cost information and the total budget for the two market segments. The accumulative customer satisfaction achieved through each TA alternative is also required for calculation. The total budget is assumed to be 24. All this information is listed in Table 4 and Table 5. It is also assumed that the number of customers in the two market segments, q_1 and q_2, respectively was estimated at 12,000 and 9,000. These data are used to show the importance of the two market segments.

Table 4. Cost and Customer Satisfaction Level for Market Segment 1

Washing Quality (%)			Noise Level (dB)			Washing Time (min)			Rinsing Quality (%)			Clothes Damage Rate (%)		
Level (%)	Cost	Satisfac-tion	Level (dB)	Cost	Satisfac-tion	Level (min)	Cost	Satisfac-tion	Level (%)	Cost	Satisfac-tion	Level (%)	Cost	Satisfac-tion
90	3	0.4342	45	5	0.2143	30	4	0.5362	95	3	0.5220	0.5	4	0.3219
95	4	0.4844	50	3	0.1643	35	2	0.4754	90	2	0.4397	0.7	2	0.3148
98	5	0.5077	60	2	0.1214	40	1	0.4183	80	1	0.3645	1	1	0.3005

Table 5. Cost and customer satisfaction level for market segment 2

Washing Quality (%)			Noise Level (dB)			Washing Time (min)			Rinsing Quality (%)			Clothes Damage Rate (%)		
Level (%)	Cost	Satisfac-tion	Level (dB)	Cost	Satisfac-tion	Level (min)	Cost	Satisfac-tion	Level (%)	Cost	Satisfac-tion	Level (%)	Cost	Satisfac-tion
92	3	0.3459	54	3	0.0014	39	1	0.3486	81	1	0.2467	1	1	0.3139
94	4	0.3620	50	4	0.0020	36	2	0.3630	83	2	0.2954	0.8	2	0.3323
96	5	0.3744	46	5	0.0029	33	3	0.3785	85	4	0.4210	0.6	3	0.3697

The proposed GA in this research was applied to solve this problem. All procedures have been coded using MATLAB and run on a PC with an Intel Core i3-2100 CPU (3.10 GHz) processor and 4 GB RAM.

A chromosome can be defined to have ten genes corresponding to the five TAs for the two market segments. A gene takes a random integer number generated from a set {1, 2, 3} which means alternatives for each TA in each market segment. An indicator variable, y_{jt} is defined as follows:

$$y_{jt} = k \qquad \text{if } x_{jkt} = 1, j=1,2,\ldots,5; k=1,2,3; t=1,2$$

Fig. 2 shows an example of this genetic representation method which stands for a solution of $x_{111} = 1$, $x_{231} = 1$, $x_{331} = 1$, $x_{411} = 1$, $x_{531} = 1$, $x_{112} = 1$, $x_{222} = 1$, $x_{332} = 1$, $x_{412} = 1$, $x_{522} = 1$.

$y_{11}=1$	$y_{21}=3$	$y_{31}=3$	$y_{41}=1$	$y_{51}=3$	$y_{12}=1$	$y_{22}=2$	$y_{32}=3$	$y_{42}=1$	$y_{52}=2$

Figure 2. Example of permutation encoding

The genetic system environment for this example was set as follows: Population size was 30, The GA was terminated after 200 generations or when 100% of the chromosomes in the population have the same fitness value, uniform crossover and mutation were adopted for generic operators. In mutation, the value of the selected gene with the mutation rate (1/20) was replaced by a random integer from {1, 2, 3}. A total of 10 runs of the algorithm was made with different random number seeds. The number of generations and whether the GA finds the optimal solution or not when the GA was terminated were measured and identified, respectively. The computational results are shown in Table 6. The GA found the optimal solutions for all 10 runs within 115 generations. The optimal solution to the example is summarized in Table 7. The OCS for this example is 3.3346. More computational experiments with comparisons of problem size, GA parameter setting, execution times, and frequency of obtaining optimal solutions are performed in the next section.

Table 6. Computational results for the application

Runs	1	2	3	4	5	6	7	8	9	10
The number of generations at termination of GA	115	87	102	95	66	91	107	74	52	73
Optimal or not	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Table 7. Summarization of results

Segments	Technical attributes	Alternatives	Customer satisfaction level	Cost
1	Washing quality (%)	95%	0.4844	4
	Noise level (dB)	60dB	0.1214	2
	Washing time (min)	35min	0.4754	2
	Rinsing quality (%)	95%	0.5220	3
	Clothes damage rate (%)	1%	0.3005	1
2	Washing quality (%)	92%	0.3459	3
	Noise level (dB)	54dB	0.0014	3
	Washing time (min)	39min	0.3486	1
	Rinsing quality (%)	85%	0.4210	4
	Clothes damage rate (%)	1%	0.3139	1

4. Computational Experiments

To test the computational performance of the proposed GA methodology, all of the developed procedures were computerized and run for a range of multi-segment market QFD planning scenarios. All procedures were coded using MATLAB and executed on a PC with an Intel Core i3-2100 CPU (3.10 GHz) processor and 8 GB RAM.

The following combinations were considered to examine the behavior of the proposed algorithm with regard to both problem size and budget availability,

i) number of market segments $\in$ {10, 15}
ii) number of TAs $\in$ {15, 30}
iii) number of alternatives for each TA $\in$ {5, 10}
iv) budget availability factor $\in$ {30%, 50%}.

A fractional factorial design has been used to plan the experiments. In this experiment, there are four factors, each of which has two levels: (1) number of market segments; (2) number of TAs; (3) number of all alternatives for each TAs; and (4) budget availability factor. Table 8 shows the orthogonal array OA (8,4,2,3) —in this notation, 8 is the number of the runs; 4 is the number of the factors; 2 is the number of the levels; and 3 is the strength, which is the number of columns where an equal number of times is guaranteed to see all possibilities. The per-level combination factors are converted into problem types. The array rows represent the conditions for experimentation. The columns of the orthogonal array correspond to the different variables or factors which are being analyzed for their effects. The entries in the array specify the levels at which the factors are to be applied.

Each combination is referred to as a problem type. The budget availability factor is defined as a percent value θ such that the budget for a multi-segment market is set equal to $B=\theta A$, A being the average cost for all alternatives of all TAs in the multi-segment market. The objective function coefficients and the costs for all alternatives of all TA in each combination are generated from the discrete uniform generator U (0.001, 0.550) and U (2, 5), respectively in reference to the example introduced to section 3 in this paper. Ten problems were generated for each combination, giving a total of 80 problems for this experimentation.

Table 8. OA (8,4,2,3)

0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

In this experimentation, the proposed GA was run once for each of 10 problems for each combination, giving a total of 80 problems. The population size is 100. Each run terminated when either 105 non-duplicate chromosomes were generated or 90% of chromosomes in the population had the same fitness value. The percentage gap, the CPU time and the number of generations which the GA takes to first reach the termination rule for 10 problems for each combination, as well as the number of instances (out of 10) for which, if known, the GA finds the optimal solution values were measured as shown in Table 9.

The quality of solutions is measured by a gap which is defined as truncated hundredths of one percent of the difference between the value of a solution obtained by the GA and the value of an optimal solution compared to the value of an optimal solution:

$$\text{Gap} = \frac{\text{The value of an optimal solution} - \text{The value of a solution obtained by GA}}{\text{The value of an optimal solution}} \times 100(\%)$$

The results for all the 80 problems in Table 9 relate to using 'bintprog', a solver in the MATLAB Optimization Toolbox to solve problems to optimality. It is shown in Table 9 that in all the problem types, the average percentage gap produced by the GA is within 0.01681 (%) as well as the number of problem types in which the number of instances (out of 10) for each problem type which the GA finds the optimal solution is greater than or equal to 5 is 5. Especially, the number of these instances for the two problem types with 10 segments, 15 TAs, 5 levels, and 30% budget level, and 15 segments, 15 TAs, 5 levels, and 50% budget level is 9.

In addition to the quality of the solution obtained from the GA, the computational performance was measured by computation times in seconds and the number of generations in the case that 90% of chromosomes in a population have the same fitness value as shown in Table 9. In the range of this experiment, computation times appear to increase with increasing the problem size which is exponentially proportional to the number of segments, TAs, and levels. Finally, the budget availability does not seem to have a significant influence on the computational results. Fig. 3 shows the variation of computation time with problem size.

Summarily, the results obtained by the GA indicate that the GA is very effective for the various instances of each problem type.

Table 9. Computational results

Problem Types				Measures			
Segment (t)	TA (j)	The number of alternatives for TA (k)	Budget θ (%)	CPU time (second)	The number of generations in the case that 90% of chromosomes in a population have the same fitness value.	optimal/non-optimal	Gap (%)
10	15	5	30	88.3688	1546	optimal	0
				92.148	1584	optimal	0
				82.7337	1451	optimal	0
				80.7036	1405	non-optimal	0.00035
				83.0783	1484	optimal	0
				74.1593	1291	optimal	0
				85.2835	1495	optimal	0
				77.9642	1349	optimal	0
				76.7367	1333	optimal	0
				77.6225	1345	optimal	0
average				81.88	1428	9 (the number of instances the GA finds optimal solution)	0.00035
10	15	10	50	224.4602	3993	optimal	0
				253.754	4496	optimal	0
				228.3355	4014	non-optimal	0.00053
				247.7869	4346	non-optimal	0.00173
				246.2015	4395	optimal	0
				229.6935	4085	non-optimal	0.00464
				224.3462	3578	non-optimal	0.0062
				224.1264	3685	optimal	0
				254.1955	4097	optimal	0
				276.0705	4453	non-optimal	0
average				240.90	4114	5 (the number of instances the GA finds optimal solution)	0.00133

10	30	5	50	330.2298	-	non-optimal	0.00072
				320.8412	-	optimal	0
				325.6341	3019	optimal	0
				319.083	2952	optimal	0
				305.5215	2788	non-optimal	0.00273
				307.6115	2871	non-optimal	0.00094
				288.891	2689	optimal	0
				286.509	2607	non-optimal	0.01138
				309.4933	2839	optimal	0
				321.7561	2954	optimal	0
average				311.56	2840	6 (the number of instances the GA finds optimal solution)	0.00160
10	30	10	30	942.6181	8402	non-optimal	0.01516
				837.1006	7419	non-optimal	0.00246
				944.9026	8399	non-optimal	0.00242
				911.9529	8158	optimal	0
				1035.70	9179	non-optimal	0.00786
				966.3991	8583	non-optimal	0.00758
				920.5992	8231	optimal	0
				961.863	8588	non-optimal	0.00422
				1045.20	9257	non-optimal	0.00067
				878.8308	7806	non-optimal	0.0307
average				944.52	8402	2 (the number of instances the GA finds optimal solution)	0.00710
15	15	5	50	201.5277	2253	optimal	0
				154.7709	1870	optimal	0
				162.0191	1789	optimal	0
				183.6365	2153	optimal	0
				185.7094	2181	optimal	0
				167.7273	2015	non-optimal	0.00266
				169.6421	2128	optimal	0
				194.2244	2307	optimal	0
				188.7768	2275	optimal	0
				208.0155	2515	optimal	0
average				181.60	2149	9 (the number of instances the GA finds optimal solution)	0.00027
15	15	10	30	461.5897	5546	non-optimal	0.0023
				449.1294	5416	non-optimal	0.01096
				543.2074	6542	optimal	0
				579.5907	7062	optimal	0
				448.4979	5461	non-optimal	0.12259
				535.3183	6487	non-optimal	0.01698
				537.2442	6662	non-optimal	0.00062
				519.1879	6362	non-optimal	0.00125
				567.8557	6942	non-optimal	0.01312
				515.1616	6317	optimal	0
average				515.68	6280	3 (the number of instances the GA finds optimal solution)	0.01681
15	30	5	30	610.2242	4173	optimal	0
				610.7684	4226	optimal	0
				601.3839	4201	optimal	0
				548.7335	3725	non-optimal	0.02031
				584.882	4043	optimal	0
				575.0002	3986	non-optimal	0.00202
				617.867	4367	optimal	0
				564.4559	3855	non-optimal	0.01905
				628.1748	4325	non-optimal	0.01
				611.4201	4291	optimal	0
average				595.29	4119	6 (the number of instances the GA finds optimal solution)	0.00514

15	30	10	50	1906	12833	non-optimal	0.00715
				1910.2	12880	non-optimal	0.01592
				1788.9	12080	non-optimal	0.00102
				1951.4	13179	non-optimal	0.00935
				1800.1	12066	non-optimal	0.0139
				1876.7	12518	optimal	0
				1837.5	12529	optimal	0
				2053.7	12727	non-optimal	0.00222
				1852.5	12426	non-optimal	0.00125
				1909.4	13010	non-optimal	0.00169
average				1888.64	12625	2 (the number of instances the GA finds optimal solution)	0.00525

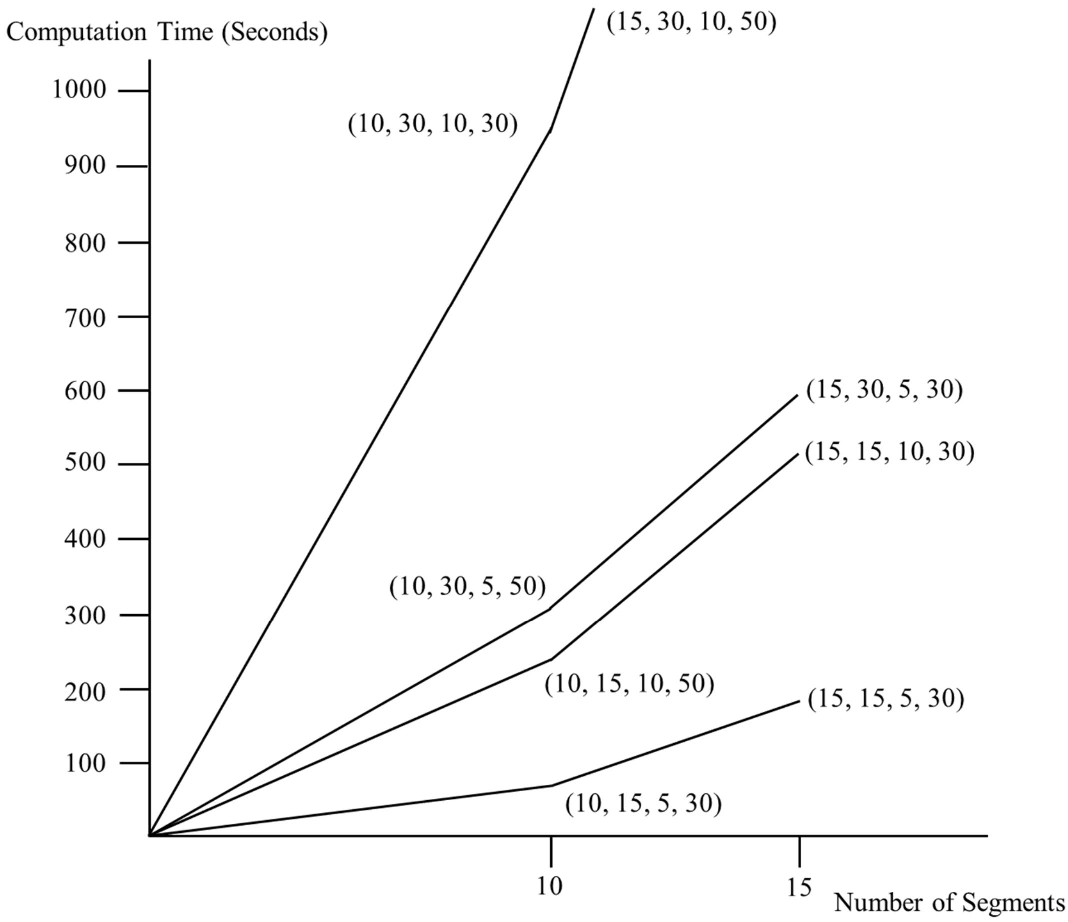

Figure 3. Computation Time vs Number of Segments

5. Conclusions

GA approach was developed to solve the problem of selecting the optimal set of TA alternatives with discrete values for each segment of the multi-segment market to maximize the OCS of the multi-segment market while not exceeding the budget available for the multi-segment market.

To explain the application of the proposed GA approach in this study, a simple example of determining the optimal levels of five TAs of a washing machine according to the five CRs in two market segments was introduced. A total of 10 runs of the algorithm was made with different random number seeds. The GA found the optimal solutions for all 10 runs within 115 generations.

To examine the behavior of the proposed algorithm as a function of both the size of the problem and the availability of the budget, a fractional factorial design was applied to plan the experiments. In this experiment the following combinations were run: (i) number of market segments $\in$ {10, 15}; (ii) number of TAs $\in$ {15, 30}; (iii) number of alternatives for each TA $\in$ {5, 10}; (iv) budget availability factor $\in$ {30%, 50%}.

It was measured that the average percentage gap obtained by the GA is within 0.017 (%) as well as the number of problem types in which the number of instances (out of 10) for each problem type which the GA finds the optimal solution is greater than or equal to 5 is 5.

Computation times seem to increase with increasing the problem size which is exponentially proportional to the number of segments, TAs, and levels. The budget availability does not seem to have a significant influence on the computational results. The computational results obtained by the GA indicate that the GA is very effective for instances of each problem type, judging by the small percentage gaps and the reasonable computation times.

After developing and launching products for each segment in a multi-segment market targeted by a company, the products for each segment must be supplemented according to the change of customer requirements over time. As such, deciding how companies should complement their products over a planning period horizon for each market segment is a critical issue in the face of an ever-changing global competition situation. As a future research theme, it seems to have considerable value in the research field.

Acknowledgments: This work was supported by the Dong-A University research fund.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Y. Akao, Quality Function Deployment: Integrating Customer Requirements into Product Design, Productivity Press, Cambridge, 1990.

[2] J. R. Hauser and D. Clausing, “The house of quality,” Harvard Business Review, vol. 66, no. 3, pp. 63-73, May. 1988.

[3] K. J. Kim, H. Moskowitz, and A. Dhingra, “Fuzzy multicriteria models for quality function deployment,” European Journal of Operational Research, vol. 121, no. 3, pp. 504-518, Mar. 2000, doi: https://doi.org/10.1016/S0377-2217(99)00048-X.

[4] X. Lai, M. Xie, and K. C. Tan, “Dynamic Programming for QFD Optimization,” Quality and Reliability Engineering, vol. 21, no. 8, pp. 769-780, Dec. 2005, doi: https://doi.org/10.1002/qre.685.

[5] E. K. Delice and Z. Güngör, “A mixed integer goal programming model for discrete values of design requirements in QFD,” International Journal of Production Research, vol. 49, no. 10, pp. 2941-2957, May. 2011, doi: https://doi.org/10.1080/00207541003720343.

[6] J. Tang, R. Y. K. Fung, B. Xu, and D. Wang, “A new approach to quality function deployment planning with financial consideration,” Computers and Operations Research, vol. 29, no. 11, pp. 1447-1463, Sep. 2002, doi: https://doi.org/10.1016/s0305-0548(01)00041-7.

[7] G. L. Urban and J. R. Hauser, Design and marketing of new products, Prentice Hall Press, Englewood Cliffs, 1993.

[8] R. B. Kazemzadeh, M. Behzadian, M. Aghdasi, and A. Albadvi, “Integration of marketing research techniques into house of quality and product family design,” International Journal of Advanced Manufacturing Technology, vol. 41, pp. 1019-1033, Apr. 2009, doi: https://doi.org/10.1007/s00170-008-1533-2.

[9] F. Franceschini and S. Rossetto, “Quality function deployment: How to improve its use,” Total Quality Management, vol. 9, no. 6, pp. 491-500, Aug. 1998, doi: https://doi.org/10.1080/0954412988424.

[10] J. Yoo, “Dynamic programming approach for determining optimal levels of technical attributes in QFD under multi-segment market,” Journal of the Society of Korea Industrial and Systems Engineering, vol. 38, no. 2, pp. 120-128, Jun. 2015, doi: https://doi.org/10.11627/jkise.2015.38.2.120.

[11] L. K. Chan and M. L. Wu, “Quality function deployment: A literature review,” European Journal of Operational Research, vol. 143, no. 3, pp. 463-497, Dec. 2002, doi: https://doi.org/10.1016/s0377-2217(02)00178-9.

[12] J. A. Carnevalli and P. C. Miguel, “Review, analysis and classification of the literature on QFD-Types of research, difficulties and benefits,” International Journal of Production Economics, vol. 114, pp. 737-754, Aug. 2008, doi: https://doi.org/10.1016/j.ijpe.2008.03.006.

[13] J. R. Sharma, A. M. Rawani, and M. Barahate, “Quality function deployment: a comprehensive literature review,” International Journal of Data Analysis Techniques and Strategies, vol. 1, no. 1, pp. 78-103, Aug. 2008, doi: https://doi.org/10.1504/ijdats.2008.020024.

[14] M. Abdolshah and M. Moradi, “Fuzzy Quality Function Deployment: An Analytical Literature Review,” Journal of Industrial Engineering, vol. 2013, pp. 1-11, May. 2013, doi: https://doi.org/10.1155/2013/682532.

[15] S. A. Oke, "Manufacturing quality function deployment: literature review and future trends," Engineering Journal, vol. 17, no. 3, pp. 79-103, Jul. 2013, doi: https://doi.org/10.4186/ej.2013.17.3.79.

[16] K. Sivasamy, C. Arumugam, S. Devadasan, R. Murugesh, and V. Thilak, "Advanced models of quality function deployment: a literature review," Quality & Quantity, vol. 50, no. 3, pp. 1399-1414, May. 2016, doi: https://doi.org/10.1007/s11135-015-0212-2.

[17] G. S. Wasserman, "On how to prioritize design requirements during the QFD planning process," IIE Transactions, vol. 25, no. 3, pp. 59-65, May. 1993, doi: https://doi.org/10.1080/07408179308964291.

[18] H. Moskowitz and K. J. Kim, "QFD optimizer: A novice friendly quality function deployment decision support system for optimizing product designs," Computers & Industrial Engineering, vol. 32, no. 3, pp. 641-655, Jul. 1997, doi: https://doi.org/10.1016/s0360-8352(96)00309-9.

[19] R. G. Askin and D. W. Dawson, "Maximizing customer satisfaction by optimal specification of engineering characteristics," IIE Transactions, vol. 32, no. 1, pp. 9-20, Jan. 2000, doi: https://doi.org/10.1080/07408170008963875.

[20] S. Y. Sohn and I. S. Choi, "Fuzzy QFD for supply chain management with reliability consideration," Reliability Engineering and System Safety, vol. 72, no. 3, pp. 327-334, Jun. 2001, doi: https://doi.org/10.1016/s0951-8320(01)00022-9.

[21] X. Lai, M. Xie, and K. C. Tan, "QFD optimization using linear physical programming," Engineering Optimization, vol. 38, no. 5, pp. 593-607, Jul. 2006, doi: https://doi.org/10.1080/03052150500448059.

[22] W. C. Ko and L. H. Chen, "An approach of new product planning using quality function deployment and fuzzy linear programming model," International Journal of Production Research, vol. 52, no. 6, pp. 1728-1743, Mar. 2014, doi: https://doi.org/10.1080/00207543.2013.848479.

[23] T. Park and K. Kim, "Determination of an optimal set of design requirements using house of quality," Journal of Operations Management, vol. 16, no. 5, pp. 569-581, Oct. 1998, doi: https://doi.org/10.1016/s0272-6963(97)00029-6.

[24] Y. Reich and E. Levy, "Managing product design quality under resource constraints," International Journal of Production Research, vol. 42, no. 13, pp. 2555-2572, Jul. 2004, doi: https://doi.org/10.1080/0020754042000208394.

[25] J. H. Shin, H. B. Jun, D. Kiritsis, and P. Xirouchakis, "A decision support method for product conceptual design considering product lifecycle factors and resource constraints," International Journal of Advanced Manufacturing Technology, vol. 52, no. 9-12, pp. 865-886, Feb. 2011, doi: https://doi.org/10.1007/s00170-010-2798-9.

[26] Z. Sener and E. E. Karsak, "A decision model for setting target levels in quality function deployment using nonlinear programming-based fuzzy regression and optimization," International Journal of Advanced Manufacturing Technology, vol. 48, no. 9-12, pp. 1173-1184, Jun. 2010, doi: https://doi.org/10.1007/s00170-009-2330-2.

[27] X. G. Luo, C. K. Kwong, and J. F. Tang, "Determining optimal levels of engineering characteristics in quality function deployment under multi-segment market," Computers & industrial engineering, vol. 59, no. 1, pp. 126-135, Aug. 2010, doi: https://doi.org/10.1016/j.cie.2010.03.009.

[28] X. Ji, Q. Gao, F. Yin, and H. Guo, "An efficient imperialist competitive algorithm for solving the QFD decision problem," Mathematical Problems in Engineering, Article ID 2601561, p. 13, 2016, doi: https://doi.org/10.1155/2016/2601561.

[29] M. Gen and R. Cheng, Genetic algorithms & engineering optimization, John Wiley & Sons Inc, New York, 2000.