

Research Article

An Efficient Imperialist Competitive Algorithm for Solving the QFD Decision Problem

Xue Ji,^{1,2} Qi Gao,^{1,2} Fupeng Yin,^{1,2} and Hengdong Guo^{1,2}

¹School of Mechanical Engineering, Shandong University, Jinan, China

²Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Shandong University, Ministry of Education, Jinan, China

Correspondence should be addressed to Qi Gao; gaoqi@sdu.edu.cn

Received 11 May 2016; Accepted 5 October 2016

Academic Editor: Marco Mussetta

Copyright © 2016 Xue Ji et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is an important QFD decision problem to determine the engineering characteristics and their corresponding actual fulfillment levels. With the increasing complexity of actual engineering problems, the corresponding QFD matrixes become much huger, and the time spent on analyzing these matrixes and making decisions will be unacceptable. In this paper, a solution for efficiently solving the QFD decision problem is proposed. The QFD decision problem is reformulated as a mixed integer nonlinear programming (MINLP) model, which aims to maximize overall customer satisfaction with the consideration of the enterprises' capability, cost, and resource constraints. And then an improved algorithm G-ICA, a combination of Imperialist Competitive Algorithm (ICA) and genetic algorithm (GA), is proposed to tackle this model. The G-ICA is compared with other mature algorithms by solving 7 numerical MINLP problems and 4 adapted QFD decision problems with different scales. The results verify a satisfied global optimization performance and time performance of the G-ICA. Meanwhile, the proposed algorithm's better capabilities to guarantee decision-making accuracy and efficiency are also proved.

1. Introduction

Quality Function Deployment (QFD), as a well-known planning methodology originated in the late 1960s in Japan by Akao and Mazur [1], focuses on translating customer requirements (CRs) into a series of relevant design requirements. During this planning process, a lot of decisions should be made. One of them is to determine the engineering characteristics (ECs) to be implemented among numerous possible selections and their corresponding actual fulfillment levels, which not only directly influences the deployment of other design requirements and the final quality of the product but also influences overall customer satisfaction and the distribution of resources.

In practice, it is a complex process to analyze and organize a lot of information related to the QFD decision problem, including multifacets of customer needs, market driven factors, price/revenue streams of enterprises, and various regulation compliance. Multiple variables and complex relationships bring many difficulties for decision makers to

make decisions. Mathematical programming, an optimization technique, such as integer programming (IP), linear programming (LP), nonlinear programming (NLP), and mixed integer nonlinear programming (MINLP), has been widely used in helping people solve these difficulties. Firstly, in order to clarify those complex relationships, the problem is abstracted as a mathematical model. Then the model is calculated by an algorithm to offer a result for decision makers as a valuable reference [2–13].

However, a rapidly changing decision environment makes people not only satisfied with obtaining a decision outcome but also want to seek for a higher decision-making efficiency, in other words, a shorter time spent on making decisions. Especially in such a society with highly developed economy, technology, and culture, customer requirements tend to be more and more diverse, product systems become gradually complex, and science technologies are continuously developing. All of these may lead to a huge and complex QFD matrix. As an important step in tackling the above QFD decision problem, the selection of algorithms influences

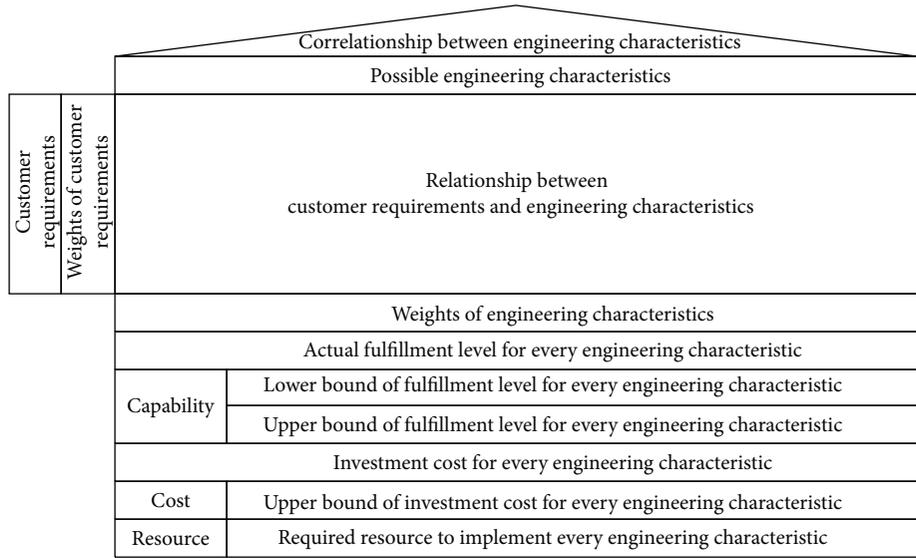


FIGURE 1: The QFD matrix.

not only the accuracy of decisions but also total decision time. And if the calculating time of the algorithm is too long, the decision efficiency and time cannot be guaranteed.

The QFD decision problem in this paper is reformulated as a MINLP model aiming to maximize overall customer satisfaction with the consideration of the enterprises' capability, cost, and resource constraints. Actually, many decision problems in various application fields, such as chemical engineering [14, 15], gas/water distribution networks optimal design [16, 17], and engineering design [18], can be formulated as MINLP models. MINLP is a NP-hard problem and the calculating algorithms to MINLPs have been developed since 1980s, which can be broadly divided into two main classes: deterministic algorithms and stochastic algorithms. Deterministic algorithms, such as branch and bound method, outer approximation method, and extended cutting plane method, can provide accurate solutions for problems in a continuous space but require an additional effort of analysis (such as the gradient analysis of objective function and constraints), which may considerably add to the work [19, 20]. Considering the existence of integer variables in QFD decision problems and easy-operation of engineers, stochastic algorithms, which are more capable of handling NP-hard problems and need not much professional mathematical knowledge, will be focused on in this paper. A lot of stochastic algorithms have been proposed. An algorithm called M-SIMPSA suitable for the optimization of MINLP is presented by Cardoso et al. [19]. Mohan and Nguyen [21] developed an improved computational algorithm called RST2ANU, which was based on the original RST2U [22]. Yiqing et al. [23] improved the particle swarm optimization (PSO) algorithm for solving nonconvex NLP/MINLP problem with equality and/or inequality constraints. Schlüter et al. [24] introduced two novel extensions for ant colony optimization (ACO) framework. Deep et al. [25] proposed a real coded genetic algorithm named MI-LXPM.

In this paper, Imperialist Competitive Algorithm (ICA), one of the stochastic algorithms proposed by Atashpaz-Gargari and Lucas [26], which simulates the social-political process of imperialism and imperialist competition, is applied to solve MINLP models. The key features of ICA are its fast convergent rate to reach global optimum, which has been proved in dealing with various optimization problems, such as structure design [27, 28], distribution network optimization [29, 30], and process planning and scheduling [31]. The advantages of ICA are beneficial to improvement of decision efficiency. In this QFD decision problem, considering the existence of integer variables in this problem, an improvement by combining genetic algorithm (GA) is carried out because the original ICA is only applied to real variables. The improved ICA, called G-IGA in the following, is compared with other mature algorithms to MINLPs, including RST2U and RST2ANU mentioned above, by calculating 7 simple numerical MINLP problems and 4 different scale QFD decision problems, and the results verify the advantages of G-ICA.

The remainder of the paper is organized as follows: the QFD decision problem is defined in Section 2. The G-ICA to resolve the problem is introduced in Section 3. Some numerical experiments and case studies are shown in Section 4, and Section 5 summarizes the paper.

2. Problem Definition

The QFD matrix is just as in Figure 1. Before the decision problem based on QFD is defined, some assumptions should be made in advance:

- (i) The QFD matrix for a certain product, including m CRs, n possible ECs, weight vectors, relationship, and correlation matrixes, is already established.
- (ii) Data about capability (lower and upper bounds of enterprise fulfillment levels for every EC), cost (maximum total budget and upper bound of investment

cost for every EC), and resource (maximum total resources and required resource to implement every EC) is offered by the enterprise.

- (iii) The fulfillment level is a percentage from 0 to 100%. For every EC, actual fulfillment level is only influenced by the corresponding investment cost, whose value is from 0 to 10 with no units. And the enterprise can determine the influence relationship based on long-term practical experience all by itself.
- (iv) There are no natural units or maximum values for customer satisfaction. It is the result of synthesizing all the actual fulfillment levels of engineering characteristics to be implemented.

Then the mathematical model is expressed as follows:

$$\max \quad CS = \sum_{j=1}^n N_j \cdot \rho_j \cdot g_j \quad (1)$$

$$\text{subject to} \quad g_{L_j} \leq g_j \leq g_{U_j} \quad (2)$$

$$0 \leq c_j \leq c_{U_j} \quad (3)$$

$$\sum_{j=1}^n N_j \cdot c_j \leq B \quad (4)$$

$$\sum_{j=1}^n N_j \cdot p_j \leq P \quad (5)$$

$$N_j \in \{0, 1\} \quad (6)$$

$$g_j = g_j(c_j) \quad (7)$$

in which the decision variables and parameters are defined as follows:

CS: the overall customer satisfaction

j : index for possible engineering characteristics, $j = 1, 2, \dots, n$

N_j : if engineering characteristic j is selected, N_j is taken as 1; otherwise, N_j is taken as 0

ρ_j : the final importance weight of the engineering characteristic j

g_j : the actual fulfillment level for engineering characteristic j , $g_j \in [0, 1]$

g_{L_j} : the lower bound of enterprise fulfillment level of engineering characteristic j , $g_{L_j} \in [0, 1]$

g_{U_j} : the upper bound of enterprise fulfillment level of engineering characteristic j , $g_{U_j} \in [0, 1]$

c_j : the investment cost for engineering characteristic j , $c_j \in [0, 10]$

c_{U_j} : the upper bound of investment cost of engineering characteristic j , $c_{U_j} \in [0, 10]$

p_j : the required resource corresponding to engineering characteristic j

B : the maximum total budget

P : the maximum total resources.

2.1. Objective Function and Decision Variables. The objective function in this model is to maximize the overall customer satisfaction CS, just as in formula (1). Whether the engineering characteristics are necessary can be viewed as 0 or 1 integer variables. If engineering characteristic j is selected, N_j is taken as 1. Otherwise, N_j is taken as 0. Another decision variable g_j , the actual fulfillment level for engineering characteristic j , is directly influenced by c_j , the corresponding investment cost. The detailed relationship expressions will be introduced in next section. Then these fulfillment levels of selected engineering characteristics will be synthesized according to their weights, to get the overall customer satisfaction. A traditional calculation method to get the weight of engineering characteristic j is as follows:

$$\rho_j'' = \sum_{i=1}^m w_i \cdot R_{ij}, \quad (8)$$

where ρ_j'' is the initial importance weight of the engineering characteristic j , w_i is the relative importance weight of the requirements i , and R_{ij} is the relationship between the customer requirement i and the engineering characteristic j , $R_{ij} \in \{\pm 1, \pm 3, \pm 5\}$.

Moreover, the correlation between different engineering characteristics is taken into consideration as in the following:

$$\rho_j = \sum_{k=1}^n \rho_j'' \cdot W_{jk} \quad (9)$$

with W_{jk} denoting the correlation between customer requirement j and k , $W_{jk} \in \{\pm 1, \pm 3, \pm 5\}$.

2.2. Relationship between Investment Cost and Actual Fulfillment Level. In reality, in formula (7), the relationship between investment cost and actual fulfillment levels is too complex to describe quantitatively. In order to simplify the problem, the relationship was generally viewed as linear in some past research [6, 32]. Here, the relationship is updated by a new expression, which can represent linear, concave, convex, or mixed relationship type.

Considering the current known conditions firstly

- (i) the independent variable c_j ranges from 0 to 10;
- (ii) the dependent variable g_j ranges from 0 to 100%;
- (iii) when c_j is taken as 0, g_j is taken as 0. When c_j is taken as 10, g_j is taken as 100%.

A preliminary expression is suggested with the help of an auxiliary function f as follows:

$$g_j(c_j) = \frac{f(c_j) - f(0)}{f(10) - f(0)}. \quad (10)$$

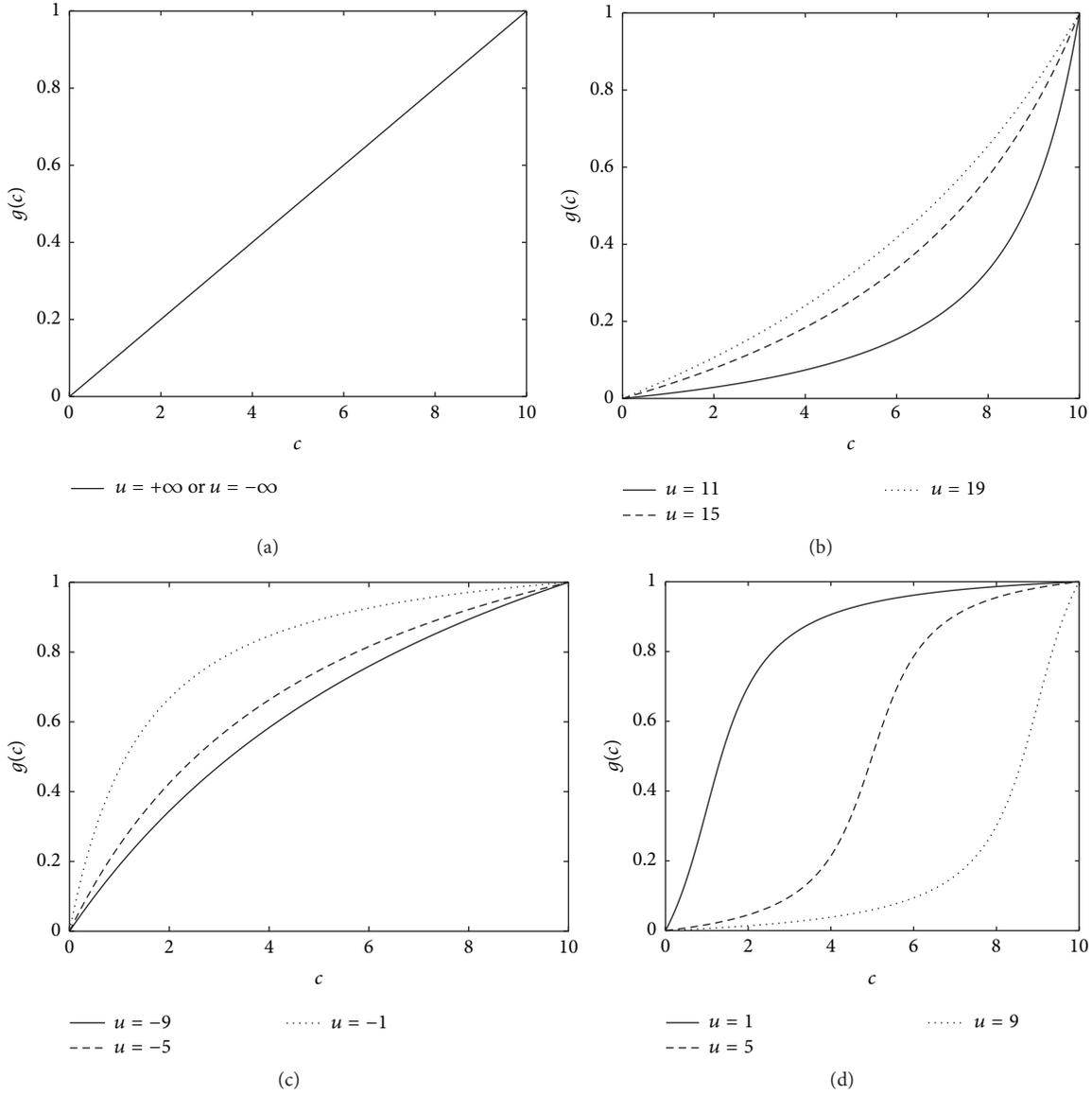


FIGURE 2: The relationship between the actual fulfillment level and the corresponding investment cost. (a) $u = +\infty$ or $-\infty$, (b) $u \geq 10$, (c) $u \leq 0$, and (d) $0 < u < 10$.

Here, a tangent function is chosen as the auxiliary function f . Moreover, in order to express more relationship types in the domain of definition with only one formula, a shape parameter u is introduced. The final expression is as follows:

$$g_j(c_j) = \frac{\arctan(c_j - u_j) + \arctan(u_j)}{\arctan(10 - u_j) + \arctan(u_j)}. \quad (11)$$

The shape parameter u varies with different engineering characteristics. Actually, u can be understood as a special investment cost value. At this point, the fulfillment level will get the fastest growing; that is, the slope of $g(c)$ is maximum.

When u goes to $+\infty$ or $-\infty$, $g(c)$ is approximately linear, just as in Figure 2(a), which is an ideal situation.

When $u \geq 10$, $g(c)$ is a concave function, just as in Figure 2(b), which represents the case that the actual fulfillment level grows faster and faster with the increase of investment cost.

When $u \leq 0$, $g(c)$ is a convex function, just as in Figure 2(c), which represents the case that the actual fulfillment level grows slower and slower with the increase of investment cost.

When $0 < u < 10$, $g(c)$ is concave at the beginning and convex afterward, just as in Figure 2(d). The investment cost value at the inflection point is equal to u . The fulfillment level will increase sharply around this point.

2.3. Capability, Cost, and Resource Constraints. The fulfillment level for every engineering characteristic is limited by actual capability of the enterprise. g_{L_j} and g_{U_j} are

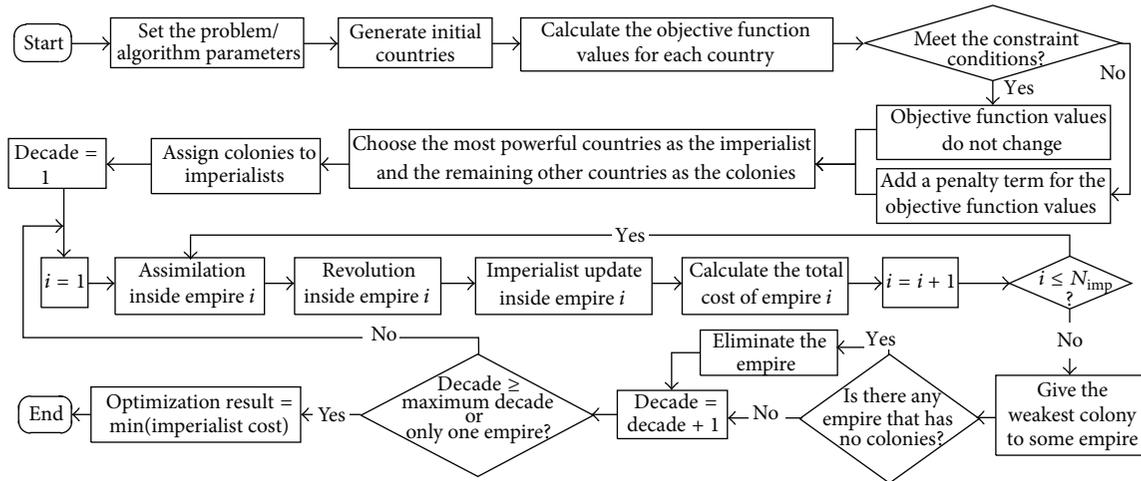


FIGURE 3: G-ICA algorithm flow chart.

corresponding to the lowest and the highest levels achieved by the enterprise. The actual fulfillment level g_j should be within this range, just as in formula (2).

Formula (3) and formula (4) are the cost constraints. The investment cost c_j of engineering characteristic j should not surpass the upper bound c_{Uj} , and the total investment cost should not exceed the total budget limit B .

Resource for every engineering characteristic is assumed to be known. The total resource should not exceed the resource limit for product development P as in formula (5).

3. Solving Algorithm

The detailed G-ICA algorithm steps are shown in Figure 3.

Step 1. A hybrid real-binary concatenated coding method is applied for the encoding and decoding of a solution.

Step 2. A penalty function method is used as the constraints handling method.

Step 3 (empires construction). The initial countries are generated and their corresponding objective function values are calculated. According to the power value of each country, which is a normalized expression of the objective function value, the most powerful countries are chosen as imperialists and the remained other countries are chosen as colonies. After the colonies are assigned to every imperialist randomly, the empires are constructed.

Step 4 (assimilation). Assimilation occurs inside every empire. According to the basic ICA, imperialists try to assimilate their colonies and make them similar to themselves, which aims to make solutions represented by colonies tending to solutions represented by imperialists. Here, the assimilation of real variables is the same as the original ICA algorithm.

But for integer variables, a certain modification has to be conducted. A multipoint crossover operator and a multipoint mutation operator are introduced, which refers to GA.

The crossover probability is P_{cr} , whose value is generally larger than 0.6. The mutation probability is P_{mu} , which is generally taken less than 0.1.

Step 5 (revolution). In each decade, revolution operators are performed on some of the colonies in each empire. There are many ways to realize revolution, such as swap, insertion, reversion, and perturbation. In this paper, some colonies are replaced by newly generated colonies. The population percentage that revolts is shown by P_{re} .

Step 6 (imperialist update). After a series of operators inside every empire, the cost of every imperialist should be updated if there exists any colony whose cost is lower than the imperialist's. In this case, the colony will become the imperialist in the current empire and vice versa.

Step 7. Calculate the total cost of each empire.

Step 8 (empires competition). The weakest colony of all empires will be selected and assigned to some empire. The lucky empire is chosen randomly; that is to say, the strongest empire does not always gain the initiative.

Step 9 (elimination of the powerless empire). When an empire loses all of the colonies, the empire will be eliminated from the population.

Step 10 (stop criterion). The stopping criterion is considered when there is only one empire that governs all of the colonies or the decades reached the limitations.

4. Case Studies and Discussions

4.1. Numerical Examples and Discussions. The G-ICA is used to solve 7 MINLP problems taken from different literatures. They are listed in the Appendix. The number of unknown decision variables in these problems varies from 2 to 10. More salient features of the test problems are listed in Table 1.

TABLE 1: Salient features of the test problems.

Problem	Number of variables			Objective function is nonlinear or not	Total	Bounds	Number of constraints			
	Total	Real	Integer				Inequality		Equality	
							Nonlinear	Linear	Nonlinear	Linear
1	2	1	1	Nonlinear	3	2	1	0	0	0
2	2	1	1	Linear	4	2	1	1	0	0
3	3	2	1	Nonlinear	6	3	1	2	0	0
4	3	2	1	Linear	4	2	0	1	1	0
5	4	2	2	Nonlinear	7	4	0	2	1	0
6	7	3	4	Nonlinear	16	7	4	5	0	0
7	10	5	5	Nonlinear	6	5	0	0	0	1

TABLE 2: Results obtained by using the following algorithm.

Problem		G-ICA	GA	$(\mu + \lambda)$ -ES	(μ, λ) -ES	SSA	RST2U	RST2ANU
1	p	100	90	100	100	100	100	100
	t	0.0268	0.5294	0.2467	5.7895	0.0998	0.0325	0.0301
	FEs	162	8360	1667	38200	1227	372	392
2	p	90	100	60	100	90	100	100
	t	0.0413	0.0627	0.5872	0.4972	2.4193	0.1660	0.0767
	FEs	1016.2	1514.6	19915	2857	16250	2113	1132
3	p	42	100	1	20	7	2	66
	t	6.0803	18.0082	9.9711	36.9382	55.5372	29.325	26.8132
	FEs	303480	1000263	330090	339173	376063	319153	303552
4	p	65	1	5	1	23	26	29
	t	5.4450	70.4521	230.7150	266.5970	30.3212	19.0057	20.0926
	FEs	260078	1992365	1852329	2281264	372594	253928	267426
5	p	59	—	51	—	39	35	39
	t	55.4304	—	156.1188	—	453.7673	615.7494	629.7191
	FEs	5000049	—	1061945	—	5135906	5247020	5132926
6	p	37	—	9	1	1	89	89
	t	5.3581	—	144.6187	187.3324	38.2674	12.4025	12.5823
	FEs	450103	—	807780	1007993	500240	100100	91705
7	p	24	—	—	—	—	3	3
	t	36.7097	—	—	—	—	301.2364	297.8793
	FEs	5000259	—	—	—	—	5045722	5012452

Index: p —the percentage of success, t —average time in seconds used by the algorithm in achieving the optimal solution in the case of the successful runs, and FEs—the average number of function evaluations in the case of successful runs.

Performance of G-ICA is compared with GA, $(\mu + \lambda)$ -ES, (μ, λ) -ES, and standard simulated annealing algorithm (SSA). Another two improved algorithms for MINLPs, RST2U, and RST2ANU algorithm, mentioned in Section 1, also take part in the comparison. All the algorithms were programmed by MATLAB R2014b, and these test problems are done on a personal computer equipped with Intel Core i7-5500U CPU, 2.4 GHz processor, 4 G RAM under Windows 8.1 operating system. The results obtained are given in Table 2.

Each problem is executed 100 runs with all the seven algorithms. Each run is initiated using a different set of initial population. A run is considered a success if achieved value of the objective function is within 1% of the known optimal value (in case the optimal value of the objective is zero, a run is considered success if the achieved absolute value of the

objective function is less than 0.01). For each problem, the percentage of success (obtained as the ratio of the number of successful runs to total number of runs), the average number of function evaluations (FEs) in the case of successful runs, and the average time in seconds used by the algorithm in achieving the optimal solution in the case of the successful runs are also listed.

From the results listed in Table 2, it can be seen that, for most problems, the G-ICA can get an optimal solution in a much shorter time. For Problem 1, most algorithms can get 100% success rate, but G-ICA's running time is the shortest and FEs is the smallest. For Problems 4, 5, and 7, on the premise of similar FEs, G-ICA can get a higher success rate in a much shorter time. For Problem 2, when the maximum number of function evaluations (MAXFEs) is 1000, G-ICA

gets 90% success rate, which is lower than RST2U and RST2ANU. But when the MAXFEs reaches 1500, G-ICA gets 100% success rate, whose running time is still shorter than RST2U and RST2ANU. The same situation occurs in solving Problem 3.

An interesting phenomenon is found in solving Problem 6. The global optimal solution of Problem 6 given in the Appendix is $(x_1, x_2, x_3, y_1, y_2, y_3, y_4, f) = (0.2, 1.280624, 1.954483, 1, 0, 0, 1, 3.557463)$. And the 37 successful runs of G-ICA are all around the global optimal solution. But RST2U and RST2ANU find another local optimal solution in searching process, $(x_1, x_2, x_3, y_1, y_2, y_3, y_4, f) = (0.2, 0.8, 1.9079, 1, 1, 0, 1, 3.5796)$. In the 89 successful runs of RST2ANU, there are about 70% points around the local optimal solution, and the following are around the global optimal solution. In our calculating method of percentage of success, all these points are viewed as successful. So the data in Table 2 shows that G-ICA is far behind in the percentage of success index. This may be an evidence to prove that G-ICA can guarantee global optimal.

In fact, the number of initial populations has a great influence on the performances of algorithms, which has been proposed by Engelbrecht [33]. In this paper, each run for each algorithm is initiated using a different set of initial population, and every index's average value is taken as reference, in order to compare these algorithms fairly. And for G-ICA, both the number of initial countries N_{pop} and number of imperialists N_{imp} , that is, the number of groups which the initial countries are divided into, are set differently. Here, an experiment is done for Problem 2 to research how these parameters influence the algorithm's performances. The iteration stop criteria of G-ICA is to reach the maximum FEs. The results are shown in Figures 4 and 5.

For the percentage of success, it can be seen from Figures 4(a) and 5(a) that more FEs provide higher percentage of success. But the parameters N_{pop} and N_{imp} 's influences to this index are hard to describe, because the relationship between N_{pop} (N_{imp}) and the percentage of success is wavy.

From the view of index FEs, the performance has become deteriorative with the increase of N_{pop} , just as in Figure 4(b), because the average FEs increase. But the average FEs will decrease when N_{imp} becomes larger, just as in Figure 5(b).

The parameters' influences to the average running time are shown clearly in the figures. On the premise of the same maximum FEs, the larger N_{pop} is, the fewer iteration times are, so the average time is shorter, just as in Figure 4(c). But the larger N_{imp} is, the more times of empires competitions are, so the average time is longer, just as Figure 5(c). In conclusion, if you want to pursue a shorter running time, you may set a larger N_{pop} and a smaller N_{imp} .

4.2. Case of the Washing Machine and Discussions. The data of this case is adapted from a washing machine case [13, 32, 34]. The QFD of a washing machine is as in Figure 6.

The importance weight matrix of engineering characteristics is as follows:

$$\omega = [5 \ 4 \ 3 \ 2 \ 2]^T. \quad (12)$$

The relationship matrix between requirements and engineering characteristics is as follows:

$$R = \begin{bmatrix} 5 & 0 & 1 & 5 & 5 \\ 3 & 5 & 1 & 1 & 0 \\ 3 & 0 & 1 & 5 & 1 \\ 3 & 1 & 1 & 3 & 5 \\ 5 & 0 & 5 & 1 & 0 \end{bmatrix}. \quad (13)$$

The correlation matrix between engineering characteristics is as follows:

$$W = \begin{bmatrix} 1 & 0 & 3 & 3 & 9 \\ 0 & 1 & 0 & 0 & 0 \\ 3 & 0 & 1 & 3 & 3 \\ 3 & 0 & 3 & 1 & 1 \\ 9 & 0 & 3 & 1 & 1 \end{bmatrix}. \quad (14)$$

The initial importance weight of engineering characteristic j is calculated by formula (8). And the result is as follows:

$$\rho'' = [62 \ 22 \ 24 \ 52 \ 38]^T. \quad (15)$$

The final importance weight of engineering characteristic j is calculated by formula (9). And the result is as follows:

$$\rho = W \cdot \rho'' = [632 \ 22 \ 480 \ 348 \ 720]^T. \quad (16)$$

Here, the corresponding shape parameters are -600000 , 200000 , -1.3 , and -0.5 , 2 .

The object function of the optimization problem is as the following formula:

$$\begin{aligned} \max \quad CS &= 632 \cdot N_1 \cdot g_1 + 22 \cdot N_2 \cdot g_2 + 480 \cdot N_3 \cdot g_3 \\ &+ 348 \cdot N_4 \cdot g_4 + 720 \cdot N_5 \cdot g_5 \end{aligned}$$

$$\begin{aligned} \text{Subject to} \quad &0.1 \leq g_1 \leq 0.5, \\ &0.1 \leq g_2 \leq 0.4, \\ &0.2 \leq g_3 \leq 0.9, \\ &0.2 \leq g_4 \leq 0.9, \\ &0.2 \leq g_5 \leq 0.7, \\ &0 \leq c_1 \leq 2, \\ &0 \leq c_2 \leq 3, \\ &0 \leq c_3 \leq 6, \\ &0 \leq c_4 \leq 5, \\ &0 \leq c_5 \leq 4, \end{aligned}$$

$$N_1 \cdot c_1 + N_2 \cdot c_2 + N_3 \cdot c_3 + N_4 \cdot c_4 + N_5 \cdot c_5$$

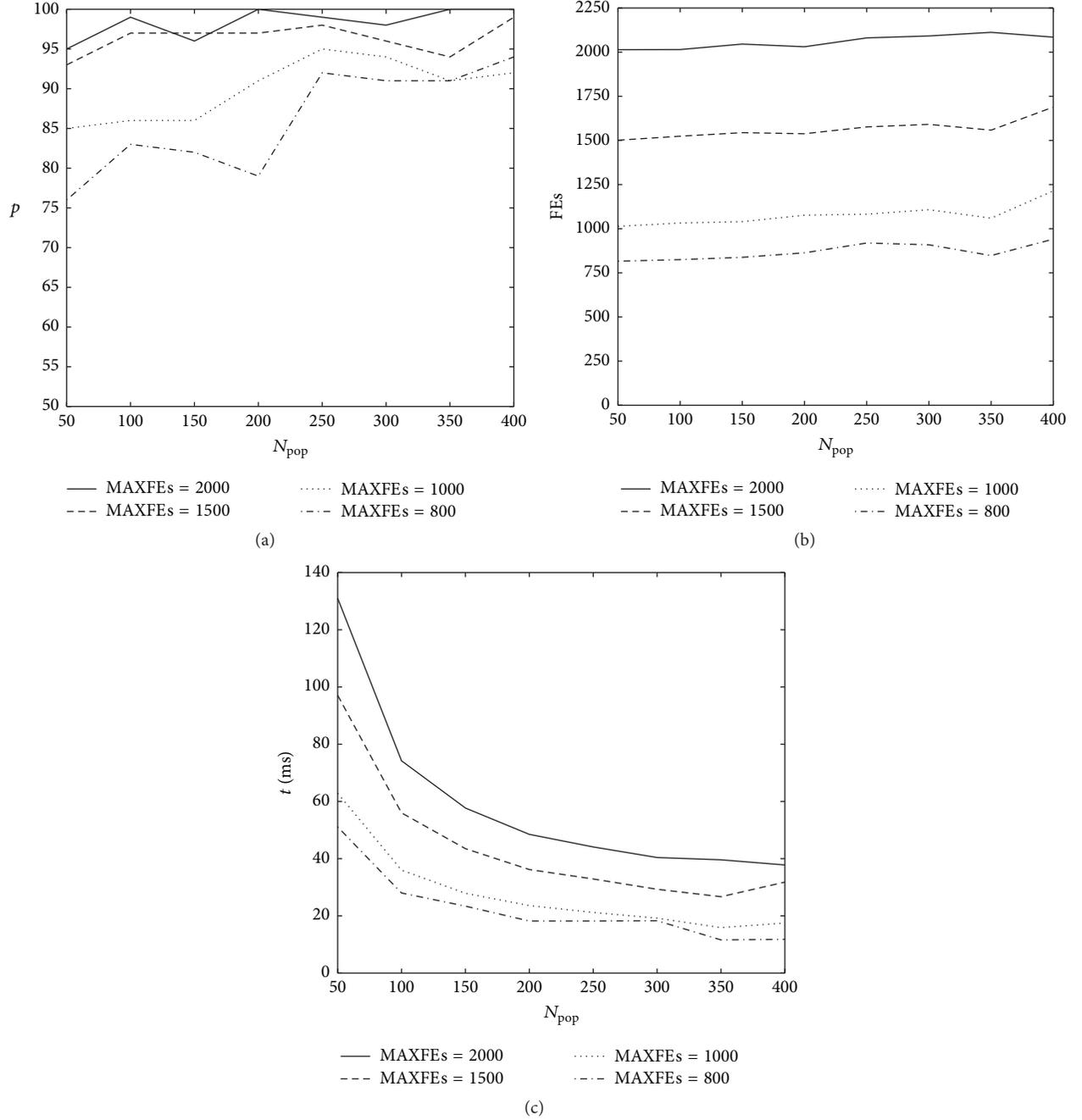


FIGURE 4: The influence of N_{pop} on percentage of success (a), average FEs of successful runs (b), and average time of the successful runs (c) for Problem 2.

$$\leq 15,$$

$$4N_1 + 2N_2 + 4N_3 + 4N_4 + 4N_5 \leq 17,$$

$$g_1 = \frac{\arctan(c_1 + 600000) + \arctan(-600000)}{\arctan(10 + 600000) + \arctan(-600000)},$$

$$g_2 = \frac{\arctan(c_2 - 200000) + \arctan(200000)}{\arctan(10 - 200000) + \arctan(200000)},$$

$$g_3 = \frac{\arctan(c_3 + 1.3) + \arctan(-1.3)}{\arctan(10 + 1.3) + \arctan(-1.3)},$$

$$g_4 = \frac{\arctan(c_4 + 0.5) + \arctan(-0.5)}{\arctan(10 + 0.5) + \arctan(-0.5)},$$

$$g_5 = \frac{\arctan(c_5 - 2) + \arctan(2)}{\arctan(10 - 2) + \arctan(2)}.$$

(17)

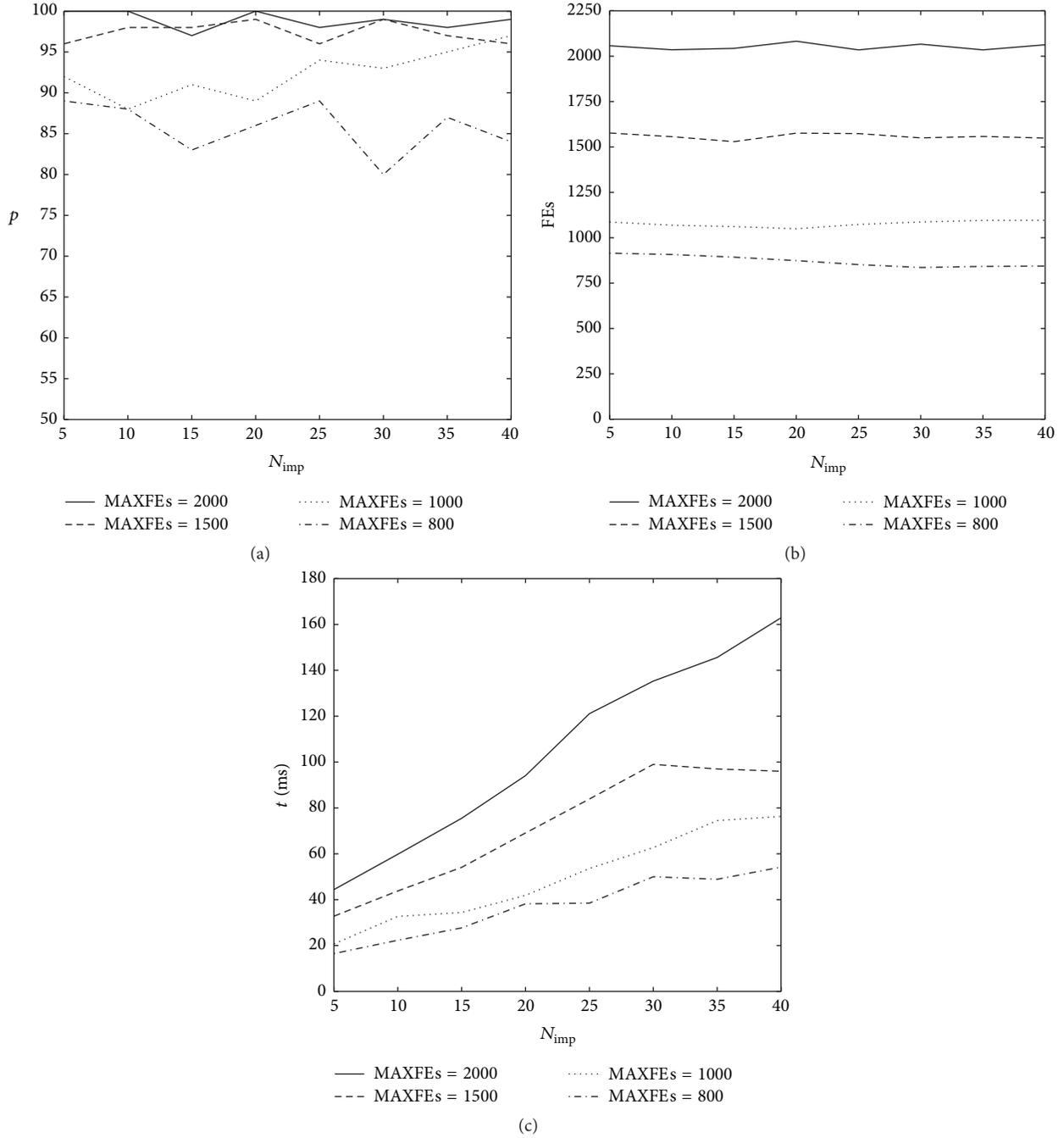


FIGURE 5: The influence of N_{imp} on percentage of success (a), average FEs of the successful runs (b), and average time of successful runs (c) for Problem 2.

The model is solved using G-ICA and RST2ANU in the same computer and every algorithm runs 10 times. The results are as the first row in Table 3. On the premise of getting similar solutions, the average FEs of RST2ANU are much smaller than that of G-ICA, but G-ICA's average time is still shorter. All of these two algorithms' solutions are clustered together. And one solution is taken as the decision reference:

$$\begin{aligned}
 & (c_1, c_2, c_3, c_4, c_5, N_1, N_2, N_3, N_4, N_5) \\
 & = (1.99, 2.31, 5.54, 4.53, 2.80, 1, 0, 1, 1, 1).
 \end{aligned}
 \tag{18}$$

The solution means that, considering the limited resource in this design phase, the engineering characteristic “noise level” can be ignored for the time being. The investment costs of the other four engineering characteristics are 1.99, 5.54, 4.53, and 2.80, respectively. And the overall customer satisfaction is 1375.4.

Beyond that, another three cases adapted from the above case are also implemented. 10, 15, and 25 possible engineering characteristics are assumed in cases 2, 3, and 4. The other problem parameters are generated randomly. Cases 2 and 3 are implemented 10 times, but case 4 is only implemented 1

in solving MINLP problems. Finally, 4 cases of washing machine with different scales are solved by G-ICA and RST2ANU. For every case, both algorithms' optimal results are clustered around the same point and can be as references for designers, to determine engineering characteristics and actual fulfillment levels. Meanwhile, the results show the G-ICA's better runtime performance than RST2ANU again.

Appendix

Problem 1. The problem is taken from [35] and also given in [25].

$$\begin{aligned} \min \quad & f(x, y) = -y + 2x - \ln\left(\frac{x}{2}\right) \\ \text{s.t.} \quad & -x - \ln\left(\frac{x}{2}\right) + y \leq 0 \\ & 0.5 \leq x \leq 1.5 \\ & y \in \{0, 1\}. \end{aligned} \quad (\text{A.1})$$

The global optimum is $(x, y, f) = (1.375, 1, 2.124)$.

Problem 2. The problem is taken from [36] and also given in [25, 35].

$$\begin{aligned} \min \quad & f(x, y) = 2x + y \\ \text{s.t.} \quad & 1.25 - x^2 - y \leq 0 \\ & x + y - 1.6 \leq 0 \\ & 0 \leq x \leq 1.6 \\ & y \in \{0, 1\}. \end{aligned} \quad (\text{A.2})$$

The global optimum is $(x, y, f) = (0.5, 1, 2)$.

Problem 3. The problem is taken from [35] and also given in [25].

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{y}) = -0.7y + 5(x_1 - 0.5)^2 + 0.8 \\ \text{s.t.} \quad & -e^{x_1 - 0.2} - x_2 \leq 0 \\ & x_2 + 1.1y + 1 \leq 0 \\ & x_1 - 1.2y - 0.2 \leq 0 \\ & 0.2 \leq x_1 \leq 1 \\ & -2.22554 \leq x_2 \leq -1 \\ & y \in \{0, 1\}. \end{aligned} \quad (\text{A.3})$$

The global optimum is $(x_1, x_2, y, f) = (0.94194, -2.1, 1, 1.07654)$.

Problem 4. The problem is taken from [36].

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{y}) = -y + 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 - 2e^{-x_2} = 0 \\ & -x_1 + x_2 + y \leq 0 \\ & 0.5 \leq x_1 \leq 1.4 \\ & y \in \{0, 1\}. \end{aligned} \quad (\text{A.4})$$

The global optimum is $(x_1, x_2, y, f) = (1.375, 0.375, 1, 2.124)$.

Problem 5. The problem is taken from [21].

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{y}) = x_1^{0.6} + x_2^{0.6} - 6x_1 - 4y_1 + 3y_2 \\ \text{s.t.} \quad & x_2 - 3x_1 - 3y_1 = 0 \\ & x_1 + 2y_1 - 4 \leq 0 \\ & x_2 + 2y_2 - 4 \leq 0 \\ & 0 \leq x_1 \leq 3 \\ & x_2 \geq 0 \\ & y_1, y_2 \in \{0, 1\}. \end{aligned} \quad (\text{A.5})$$

The global optimum is $(x_1, x_2, y_1, y_2, f) = (4/3, 4, 0, 0, -4.5142)$.

Problem 6. The problem is taken from [25].

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{y}) \\ & = (y_1 - 1)^2 + (y_2 - 1)^2 + (y_3 - 1)^2 \\ & \quad - \ln(y_4 + 1) + (x_1 - 1)^2 + (x_2 - 2)^2 \\ & \quad + (x_3 - 3)^2 \\ \text{s.t.} \quad & y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \leq 5 \\ & y_3^2 + x_1^2 + x_2^2 + x_3^2 \leq 5.5 \\ & y_1 + x_1 \leq 1.2 \\ & y_2 + x_2 \leq 1.8 \\ & y_3 + x_3 \leq 2.5 \\ & y_4 + x_1 \leq 1.2 \\ & y_2^2 + x_2^2 \leq 1.64 \\ & y_3^2 + x_3^2 \leq 4.25 \end{aligned}$$

$$\begin{aligned}
y_2^2 + x_3^2 &\leq 4.64 \\
x_1 &\geq 0 \\
x_2 &\geq 0 \\
x_3 &\geq 0 \\
y_1, y_2, y_3, y_4 &\in \{0, 1\}.
\end{aligned}
\tag{A.6}$$

The global optimum is $(x_1, x_2, x_3, y_1, y_2, y_3, y_4, f) = (0.2, 1.280624, 1.954483, 1, 0, 0, 1, 3.557463)$.

Problem 7. The problem is taken from [25].

$$\begin{aligned}
\max \quad & f(\mathbf{x}) \\
&= \sum_{i=1}^9 x_i x_{i+1} + \sum_{i=1}^8 x_i x_{i+2} + x_1 x_9 + x_1 x_{10} \\
&\quad + x_2 x_{10} + x_1 x_5 + x_4 x_7 \\
\text{s.t.} \quad & \sum_{i=1}^{10} x_i = 1 \\
& x_1, x_2, x_3, x_8, x_9 \in \{0, 1\}.
\end{aligned}
\tag{A.7}$$

The optimal solution as given in the source is $X = (0, 0, 0, 0.25, 0.25, 0.25, 0.25, 0, 0, 0)$, with $f_{\max} = 0.375$. We have considered this problem as a mixed integer optimization problem where x_1, x_2, x_3, x_8, x_9 are supposed to be integer and the remaining variables can have real values.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 51375277, Shandong Province Natural Science Foundation of China under Grant no. ZR2012GM015, and the National High-Tech Research and Development projects (863) under Grant no. 2013AA040204.

References

- [1] Y. Akao and G. H. Mazur, "The leading edge in QFD: past, present and future," *International Journal of Quality and Reliability Management*, vol. 20, no. 1, pp. 20–35, 2003.
- [2] G. S. Wasserman, "On how to prioritize design requirements during the QFD planning process," *IIE Transactions*, vol. 25, no. 3, pp. 59–65, 1993.
- [3] T. Park and K.-J. Kim, "Determination of an optimal set of design requirements using house of quality," *Journal of Operations Management*, vol. 16, no. 5, pp. 569–581, 1998.
- [4] R. Y. K. Fung, J. Tang, Y. Tu, and D. Wang, "Product design resources optimization using a non-linear fuzzy quality function deployment model," *International Journal of Production Research*, vol. 40, no. 3, pp. 585–599, 2002.
- [5] Y. Reich and E. Levy, "Managing product design quality under resource constraints," *International Journal of Production Research*, vol. 42, no. 13, pp. 2555–2572, 2004.
- [6] X. Lai, M. Xie, and K. C. Tan, "Optimizing product design using the Kano model and QFD," in *Proceedings of the IEEE International Engineering Management Conference (IEMC '04)*, pp. 1085–1089, Singapore, October 2004.
- [7] X. Lai, M. Xie, and K. C. Tan, "Dynamic programming for QFD optimization," *Quality and Reliability Engineering International*, vol. 21, no. 8, pp. 769–780, 2005.
- [8] E. K. Delice and Z. Güngör, "A new mixed integer linear programming model for product development using quality function deployment," *Computers and Industrial Engineering*, vol. 57, no. 3, pp. 906–912, 2009.
- [9] 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, 2011.
- [10] E. K. Delice and Z. Güngör, "Determining design requirements in QFD using fuzzy mixed-integer goal programming: application of a decision support system," *International Journal of Production Research*, vol. 51, no. 21, pp. 6378–6396, 2013.
- [11] 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, 2014.
- [12] Y. Miao, Y. Liu, and Y. Chen, "Determination of target values of engineering characteristics in QFD using uncertain programming," *Journal of Uncertainty Analysis and Applications*, vol. 3, no. 1, pp. 1–20, 2015.
- [13] 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, 2010.
- [14] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, "Global optimization of MINLP problems in process synthesis and design," *Computers and Chemical Engineering*, vol. 21, supplement, pp. S445–S450, 1997.
- [15] M. Yuceer and R. Berber, "A heuristic approach for solution of MINLP problem for production scheduling in preemptive mode," *Chemical Engineering and Processing: Process Intensification*, vol. 44, no. 9, pp. 933–940, 2005.
- [16] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth, "On the optimal design of water distribution networks: a practical MINLP approach," *Optimization and Engineering*, vol. 13, no. 2, pp. 219–246, 2012.
- [17] P. F. Lyons, N. S. Wade, T. Jiang et al., "Design and analysis of electrical energy storage demonstration projects on UK distribution networks," *Applied Energy*, vol. 137, pp. 677–691, 2015.
- [18] J. Zhang, P. Liu, Z. Zhou, L. Ma, Z. Li, and W. Ni, "A mixed-integer nonlinear programming approach to the optimal design of heat network in a polygeneration energy system," *Applied Energy*, vol. 114, pp. 146–154, 2014.
- [19] M. F. Cardoso, R. L. Salcedo, D. S. Azevedo, and D. Barbosa, "A simulated annealing approach to the solution of minlp problems," *Computers & Chemical Engineering*, vol. 21, no. 12, pp. 1349–1364, 1997.
- [20] N. Siddique and H. Adeli, "Nature inspired computing: an overview and some future directions," *Cognitive Computation*, vol. 7, no. 6, pp. 706–714, 2015.

- [21] C. Mohan and H. T. Nguyen, "A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems," *Computational Optimization and Applications*, vol. 14, no. 1, pp. 103–132, 1999.
- [22] W. L. Price, "Global optimization by controlled random search," *Journal of Optimization Theory and Applications*, vol. 40, no. 3, pp. 333–348, 1983.
- [23] L. Yiqing, Y. Xigang, and L. Yongjian, "An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints," *Computers and Chemical Engineering*, vol. 31, no. 3, pp. 153–162, 2007.
- [24] M. Schlüter, J. A. Egea, and J. R. Banga, "Extended ant colony optimization for non-convex mixed integer nonlinear programming," *Computers & Operations Research*, vol. 36, no. 7, pp. 2217–2229, 2009.
- [25] K. Deep, K. P. Singh, M. L. Kansal, and C. Mohan, "A real coded genetic algorithm for solving integer and mixed integer optimization problems," *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 505–518, 2009.
- [26] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 4661–4667, Singapore, 2007.
- [27] B. Abdi, H. Mozafari, A. Ayob, and R. Kohandel, "Imperialist competitive algorithm and its application in optimization of laminated composite structures," *European Journal of Scientific Research*, vol. 55, no. 2, pp. 174–187, 2011.
- [28] S. Talatahari, A. Kaveh, and R. Sheikholeslami, "Chaotic imperialist competitive algorithm for optimum design of truss structures," *Structural and Multidisciplinary Optimization*, vol. 46, no. 3, pp. 355–367, 2012.
- [29] A. Soroudi and M. Ehsan, "Imperialist competition algorithm for distributed generation connections," *IET Generation, Transmission and Distribution*, vol. 6, no. 1, pp. 21–29, 2012.
- [30] M. H. Moradi, A. Zeinalzadeh, Y. Mohammadi, and M. Abedini, "An efficient hybrid method for solving the optimal siting and sizing problem of DG and shunt capacitor banks simultaneously based on imperialist competitive algorithm and genetic algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 54, pp. 101–111, 2014.
- [31] K. Lian, C. Zhang, L. Gao, and X. Li, "Integrated process planning and scheduling using an imperialist competitive algorithm," *International Journal of Production Research*, vol. 50, no. 15, pp. 4326–4343, 2012.
- [32] 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, 2011.
- [33] A. P. Engelbrecht, "Fitness function evaluations: a fair stopping condition?" in *Proceedings of the IEEE Symposium on Swarm Intelligence (SIS '14)*, pp. 181–188, Orlando, Fla, USA, December 2014.
- [34] H. Yamashina, T. Ito, and H. Kawada, "Innovative product development process by integrating QFD and TRIZ," *International Journal of Production Research*, vol. 40, no. 5, pp. 1031–1050, 2002.
- [35] L. Costa and P. Oliveira, "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems," *Computers and Chemical Engineering*, vol. 25, no. 2-3, pp. 257–266, 2001.
- [36] G. R. Kocis and I. E. Grossmann, "Global optimization of non-convex mixed-integer nonlinear programming (MINLP) problems in process synthesis," *Industrial and Engineering Chemistry Research*, vol. 27, no. 8, pp. 1407–1421, 1988.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

