Information Sciences 181 (2011) 4124–4138

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Evolutionary algorithms for optimization problems with uncertainties and hybrid indices

Dun-wei Gong, Na-na Qin *, Xiao-yan Sun

School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China

A R T I C L E I N F O

Article history:
Received 19 October 2009
Received in revised form 10 March 2011
Accepted 11 May 2011
Available online 24 May 2011

Keywords:
Evolutionary optimization
Uncertainty
Hybrid indices
Genetic algorithms
Interaction

A B S T R A C T

Many optimization problems in real-world applications contain both explicit (quantitative) and implicit (qualitative) indices that usually contain uncertain information. How to effectively incorporate uncertain information in evolutionary algorithms is one of the most important topics in information science. In this paper, we study optimization problems with both interval parameters in explicit indices and interval uncertainties in implicit indices. To incorporate uncertainty in evolutionary algorithms, we construct a mathematical uncertain model of the optimization problem considering the uncertainties of interval objectives; and then we transform the model into a precise one by employing the method of interval analysis; finally, we develop an effective and novel evolutionary optimization algorithm to solve the converted problem by combining traditional genetic algorithms and interactive genetic algorithms. The proposed algorithm consists of clustering of a large population according to the distribution of the individuals and estimation of the implicit indices of an individual based on the similarity among individuals. In our experiments, we apply the proposed algorithm to an interior layout problem, a typical optimization problem with both interval parameters in the explicit index and interval uncertainty in the implicit index. Our experimental results confirm the feasibility and efficiency of the proposed algorithm.

1. Introduction

Many optimization problems in real-world applications contain both explicit (quantitative) and implicit (qualitative) indices. Explicit indices can be expressed in the form of well-defined functions while implicit indices cannot be expressed in this way. Implicit indices can only be described through opinions or thoughts of a person which vary from person to person. We refer to their occurrence as problems with hybrid indices [5]. It is difficult to determine the exact values of the parameters in these explicit indices. Without doubt, fuzzy cognition of a person on an object will lead to an uncertain evaluation. In our study, we focus on optimization problems with uncertainties and hybrid indices. In the following paragraphs, we review some recent research on optimization with uncertainties.

Cheng et al. employed three methods to deal with optimization problems with uncertainties, i.e., stochastic programming, fuzzy programming and interval programming [4]. When an optimization problem has "stochastic" parameters, stochastic programming can be employed if, a *priori*, the probability distributions of these variables are known. In the case where an optimization problem is characterized as fuzzy, fuzzy programming can be employed if the membership functions of these fuzzy parameters are known in advance. Unfortunately, it is usually difficult to know in advance the probability

* Corresponding author. Tel.: +86 516 83995199; fax: +86 516 83995312.
E-mail addresses: dwgong@vip.163.com (D.-w. Gong), qinnana186@126.com (N.-n. Qin), xysun78@126.com (X.-y. Sun).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2011.05.011

distribution of a stochastic variable or the membership function of a fuzzy parameter in real-world applications, so the applications of stochastic and fuzzy programming are limited. Interval programming is employed for optimization problems where parameters take interval values if the range of these intervals is known beforehand. Interval programming is popular since it is often easy to obtain the range of an interval.

To tackle optimization problems with uncertain information, the common approach is to transform uncertain optimization problems into precise optimization problems. The transformed problems are then solved by traditional or intelligent optimization methods [15]. Note that such a transformation might generate a series of optimization problems with precise parameters, indicating that effective algorithms are preferred. Evolutionary algorithms (EAs), especially genetic algorithms (GAs), are stochastic optimization methods dating back to the 1960s. GAs have been successfully applied in a large number of real-world optimization problems. In the following, we mainly review the application of GAs to uncertain optimization problems.

Recently, many successes have been achieved in solving optimization problems with uncertain parameters in explicit indices and/or constraints. Jin et al. provided a survey of EAs on optimization problems with stochastic parameters [17]. Cheng et al. and Jiang employed a master–slave parallel genetic algorithm to solve a two-stage optimization problem with precise parameters transformed from an optimization problem with interval parameters [4,15]. Taking both the upper and the lower limits, or both the span and the mid-point of an interval as different objectives, Zheng et al. and Jiang et al. converted their original optimization problems to a multi-objective optimization problem with a number of precise parameters [16,29]. Then, Limbourg et al. proposed an approach to determine the dominant relation of different individuals based on their interval dominance when solving a multi-objective optimization problem with interval parameters using EAs [22]. Soares et al. developed a robust interval multi-objective evolutionary algorithm based on the worst scenario of an optimization problem, which combines analytical interval techniques to deal with uncertainties in a deterministic way with a multi-objective evolutionary algorithm [26]. Fieldsend et al. presented an approach to compare different individuals based on probabilistic dominance when employing EAs to deal with a multi-objective optimization problem with stochastic parameters [6]. Martinez et al. described a tracking controller for the dynamics of a mobile unicycle robot by integrating a kinematic and a torque controller based on a type-2 fuzzy logic theory and GAs [24]. In addition, Toscano et al. utilized Kharitonov's theorem and an evolutionary algorithm to design a robust static output feed back controller with minimal quadratic cost in the context of multiple parametric uncertainties [27].

However, these studies are unsuitable to solve optimization problems with implicit indices, which are difficult to be expressed with well-defined functions, required when the fitness of an individual is calculated in a GA.

Interactive genetic algorithms (IGAs), which incorporate the intelligent evaluation of users into traditional evolution mechanisms, are effective methods of dealing with these problems. In IGAs, users assign the fitness of an individual according to their cognition or preferences. For uncertainties in explicit indices of an optimization problem are different from the ones in implicit indices, the result from a fuzzy and gradual cognition by users is of an uncertain nature. Obviously, the fitness of an individual should be expressed with an uncertain value.

Previous methods often employed precise values to express the fitness of an individual, which cannot reflect the cognition of the user on the evaluated object, and thus restrict the effectiveness of IGAs. We proposed to represent the fitness of an individual using an interval and choose a dominant individual based on the probabilistic dominance of the interval, which effectively reflects the uncertain cognition of the user [7]. Incorporating the fuzzy cognition of the user, we further adopted a fuzzy number described with Gaussian membership to express the fitness of an individual, and compared different individuals based on the fuzzy cut set and the probabilistic dominance of the interval [8]. Apart from these uncertainties, arising from cognition of the user, stochastic uncertainty also exists during optimizing implicit indices. Therefore, by incorporating an error theory into the fuzzy fitness of an individual, we further expressed the fitness of an individual with both a fuzzy number described with Gaussian membership and a stochastic variable, and compared different individuals based on the confidence level, the fuzzy cut set and the probabilistic dominance of the interval [28]. In addition, Guo et al. proposed an approach to express the fitness of an individual with a grey number, an uncertain number with fixed lower and upper bounds but unknown distribution [9].

These approaches not only describe the uncertainties in IGAs, but also alleviate user fatigue as a result of reducing the psychological burden of the user in evaluation, hence improving the performance of IGAs.

Although it is very difficult to solve an optimization problem with hybrid indices and precise parameters in explicit indices, a number of approaches have been proposed during the last decade by using both traditional GAs and IGAs. Zhou et al. scaled the values of various indices obtained in different ways, calculated the fitness of an individual by weighing them, and applied the proposed algorithm in fashion designs [30]. Brintrup et al. studied an optimization problem with multi-dimensional implicit indices, evaluated an individual in the performance of these implicit indices by combining IGAs with fuzzy systems, and then formulated a multi-objective optimization problem together with the explicit indices and solved it using NSGA-II. This algorithm is named a multi-objective IGA, and applied in a floor-planning problem [1]. They further compared the performance of a sequential IGA with a multi-objective IGA, where the former first optimizes the implicit indices by employing IGAs for some generations, and then optimizes the explicit indices by employing traditional GAs based on these optimal solutions obtained by IGAs. These authors confirmed that the performance of the multi-objective IGA is better than that of the sequential IGA in optimizing the floor-planning problem [2]. They also solved an ergonomic chair design problem with qualitative and quantitative criteria. Different design criteria are treated as optimization objectives and solutions are iteratively improved through the cooperative efforts of the computer and the user [3]. Kamalian et al. first focus on optimization problems with multiple explicit indices. After evolving the population for some generations by employing traditional multi-objective GAs, the user then evaluates the optimal solutions according to implicit indices and their offspring are then generated based on the evaluation of the user [19]. Kamalian et al. also launched a proposal for a number of individuals to be evaluated by the user in every some generations and to have their Pareto ranks changed based on these evaluations [20]. Recently, we solved an optimization problem with multi-dimensional hybrid indices using IGAs. The entire optimization process is divided into several phases according to different degrees of importance of these indices, where only the indices with the same degree of importance are optimized in each phase. Under this strategy, a complex optimization problem is converted to several simple sequential optimization problems, effectively alleviating user fatigue [10].

These successes provide feasible ways to solve optimization problems with both hybrid indices and precise parameters in explicit indices. However, there are still two obvious disadvantages: one is that both the size of the population, i.e., the number of individuals in the population, and the number of generations are required to be small in IGAs to reduce the involvement with the user, which restricts the performance of IGAs in exploration; the other is that it is difficult to guarantee simultaneous optimization of two kinds of indices since they are iteratively optimized in different phases.

Previous achievements are suitable for solving the following optimization problems: problems with uncertain parameters in explicit indices, problems with implicit indices, and problems with hybrid indices and precise parameters in explicit indices. However, there have been few evolutionary theories and methods available for problems with both uncertain parameters in explicit indices and uncertainties in implicit indices due to the fact that these problems are very complicated, containing a large number of different types of indices with various kinds and degrees of uncertainties.

The objective of our study is to solve an optimization problem with both interval parameters in explicit indices and interval uncertainty in implicit indices by employing a novel evolutionary optimization algorithm. First, we construct a mathematical uncertainty model for this problem, then transform the model into a precise one by employing the method of interval analysis. Finally, we present a novel evolutionary optimization algorithm to effectively solve the converted problem by combining traditional GAs with IGAs. Therefore, the key techniques contain the following three aspects, i.e., modeling the optimization problem with both interval parameters in explicit indices and interval uncertainties in implicit indices, transforming the uncertain model into a precise one, and proposing efficient evolutionary optimization strategies.

This paper is organized as follows. Section 2 describes some basic knowledge about interval analysis. The mathematical uncertainty model for the optimization problem with both interval parameters in explicit indices and interval uncertainties in implicit indices, the transformation process and results from the uncertain model into the precise one are studied in Section 3. Section 4 presents an efficient evolutionary optimization algorithm in detail, including clustering of a large population, i.e., a population consisted of many individuals, and estimation of the implicit indices of an individual. An application of our method in an interior layout problem is provided in Section 5, and finally, Section 6 draws conclusions and suggests possible opportunities for future research.

2. Basic knowledge about interval analysis

2.1. Interval

$A = [\underline{a}, \bar{a}]$ is called an interval, where both $\underline{a}$ and $\bar{a}$ are real values, they are the lower and the upper limits of A, respectively. If $\underline{a}$ is equal to $\bar{a}$, A regresses to a point and is called a point interval [11]. An interval can be uniquely expressed with both limits. In addition, an interval can also be expressed with its mid-point and radius [25], i.e.,:

$$A = [m(A) - w(A), m(A) + w(A)] = m(A) + [-1, 1]w(A), \tag{1}$$

where $m(A)$ and $w(A)$ are the mid-point and the radius of A, respectively. Their expressions are as follows:

$$\begin{aligned} m(A) &= \frac{\underline{a} + \bar{a}}{2}, \\ w(A) &= \frac{\bar{a} - \underline{a}}{2}. \end{aligned} \tag{2}$$

2.2. Interval dominance

When a method is employed to solve an optimization problem with interval parameters, it is inevitable to compare the performance of different potential solutions, equivalent to comparing the objective(s) of these solutions, i.e., the dominant relation(s) of these intervals. For two intervals, whether one dominates the other or not is determined by the optimization problem. For purpose of illustration, an optimization problem with only one interval objective will be considered in this subsection, where the interval objective is denoted as $f(x)$.

For a minimization problem with only one interval objective, i.e., $\min f(x)$, we expect that both the mid-point and the radius of the objective of an optimal solution are as small as possible, because a small mid-point of the objective means that the average performance of the optimal solution is good; in addition, and a small radius of the objective means that the uncertainty of the optimal solution is low. For two optimal solutions x_i and x_j with their objective being $f(x_i)$ and $f(x_j)$, x_i dominates x_j, equivalent to $f(x_i)$ interval dominating $f(x_j)$, if and only if both the mid-point and the radius of $f(x_i)$ are smaller than or equal to those of $f(x_j)$ [18], denoted as $x_j \leqslant_{mw} x_i$, where $\leqslant_{mw}$ means that the dominance is based on the comparison of the mid-point and the radius of the interval. We further assume that x_i strictly dominates x_j, if and only if x_i is not equal to x_j and x_i dominates x_j, and denote this case as $x_j <_{mw} x_i$. This case can be described as follows:

$$\begin{aligned} &x_j \leqslant_{mw} x_i \iff m(f(x_i)) \leqslant m(f(x_j)) \cap w(f(x_i)) \leqslant w(f(x_j)),\\ &x_j <_{mw} x_i \iff x_j \leqslant_{mw} x_i \cap x_j \neq x_i. \end{aligned} \tag{3}$$

The maximization problem, i.e., max$f(x)$, is different from a minimization problem with only one interval objective, where we expect the mid-point of the objective of an optimal solution to be as large as possible. Simultaneously, the radius of the objective of the optimal solution needs to be as small as possible because a large mid-point of the objective implies that the average performance of the optimal solution is good, and a small radius of the objective means that the uncertainty of the optimal solution is low. In this situation, the relation of x_i dominating x_j, as well, the condition that x_i strictly dominates x_j can be described as follows, given that we use a method similar to that described earlier.

$$\begin{aligned} &x_j \leqslant_{mw} x_i \iff m(f(x_i)) \geqslant m(f(x_j)) \cap w(f(x_i)) \leqslant w(f(x_j)),\\ &x_j <_{mw} x_i \iff x_j \leqslant_{mw} x_i \cap x_j \neq x_i. \end{aligned} \tag{4}$$

In fact, it is unnecessary that the mid-point and the radius of the objective are employed to compare different potential solutions. We can also use the upper and the lower limits of the objective, the lower limit and the mid-point of the objective and other combinations when comparing different potential solutions. It is easy to understand that different combinations of the characteristics of the objective emphasize various aspects of the objective, reflecting different manners of a decision-maker.

We use the mid-point and the radius of each objective to compare the performance of different solutions, so that both the average performance and the degree of uncertainty of each objective can be emphasized, i.e., we expect that the average performance of each objective of an optimal solution is as good as possible, while at the same time, the degree of uncertainty of each objective of the optimal solution is the lowest.

3. Modeling optimization problem with both interval parameters in explicit indices and interval uncertainties in implicit indices

Consider the following optimization problem:

$$\begin{aligned} &\min : f_1(x,a), f_2(x,a), \ldots, f_p(x,a),\\ &\max : f_{p+1}(x), f_{p+2}(x), \ldots, f_{p+q}(x), \end{aligned} \tag{5}$$

$$\begin{aligned} &\text{s.t. } x \in \mathbf{S} \subseteq R^n,\\ &a = (a_1, a_2, \ldots, a_k)^{\mathrm{T}}, \quad a_l = [\underline{a_l}, \overline{a_l}], \quad l = 1, 2, \ldots, k, \end{aligned}$$

where x is an n-dimensional decision variable belonging to the domain S, $f_i(x,a)$, $i = 1,2,\ldots,p$ the ith explicit index with vector parameter a, $f_j(x)$, $j = p + 1, p + 2, \ldots, p + q$ the value of the $(j-p)$th implicit index, an interval reflecting the evaluation of the user on x, and a the vector parameter with k intervals in which a_l is the l-th component with $\underline{a_l}$ and $\overline{a_l}$ the lower and the upper limits, respectively.

If p is equal to zero, then Eq. (5) is an optimization problem with interval uncertainties in implicit indices; if q is equal to zero, this equation has interval parameters in explicit indices; if both p and q are larger than zero, this equation describes the condition of both interval parameters in explicit indices and interval uncertainties in implicit indices. Since the first two scenarios are special cases of the third, the solution of the third case is also applicable to the first two cases. We consider therefore only the last case scenario.

3.1. Transformation of explicit indices with interval parameters

Jiang et al. considered a minimization problem with interval parameters in explicit indices [18]. It is clear that the objective of the problem is also an interval. Since the mid-point and the radius are two important characteristics of an interval, they transformed the minimization of the original objective into minimizing both its mid-point and radius by employing interval analysis, a bi-objective minimization problem with precise parameters.

Without loss of generality, we consider the ith explicit index $f_i(x,a)$ in Eq. (5) and employ the same method of dealing with an interval objective as that of Jiang et al. After transformation, the minimization of $f_i(x,a)$ turns into minimizing both $m(f_i(x,a))$ and $w(f_i(x,a))$ with only precise parameters.

When considering different ranges of two objectives, it is necessary to normalize their values into the range of [0,1] by $\frac{m(f_i(x,a)) - \min\, m(f_i(x,a))}{\max\, m(f_i(x,a)) - \min\, m(f_i(x,a))}$ and $\frac{w(f_i(x,a)) - \min\, w(f_i(x,a))}{\max\, w(f_i(x,a)) - \min\, w(f_i(x,a))}$, where min$m(f_i(x,a))$ and max$m(f_i(x,a))$ are the minimum and maximum values of the mid-point of $f_i(x,a)$ in some generation, respectively, and min $w(f_i(x,a))$ and max$w(f_i(x,a))$ are the minimum and maximum values of the radius of $f_i(x,a)$ in some generation, respectively. It is clear that these values change along with the optimization process.

A decision-maker may have different preferences on these two objectives in real-world applications. We accumulate these two normalized objectives into one by assigning different weights to the two objectives. Let β be the weight, reflecting the preference of the decision-maker on $m(f_i(x,a))$ whose value is between zero and one, set by the decision-maker in advance. Then $1-\beta$ is the weight which reflects the preference of the decision-maker on $w(f_i(x,a))$. By incorporating the weights, the minimization of $f_i(x,a)$ is transformed into the following minimization problem with precise parameters:

$$\min F_i(x) = \beta\frac{m(f_i(x,a)) - \min m(f_i(x,a))}{\max m(f_i(x,a)) - \min m(f_i(x,a))} + (1-\beta)\frac{w(f_i(x,a)) - \min w(f_i(x,a))}{\max w(f_i(x,a)) - \min w(f_i(x,a))},\quad \text{s.t. } x \in \mathbf{S} \subseteq R^n. \tag{6}$$

3.2. Transformation of implicit indices with interval uncertainties

Although several means can be employed to express uncertain cognition of the user on the object, e.g., an interval, a fuzzy value or a stochastic variable, we use intervals to express the values of these implicit indices. For the $(j-p)$th implicit index $f_j(x)$, $j = p+1, p+2, \ldots, p+q$, the user should assign the values of $m(f_j(x))$ and $w(f_j(x))$ when he/she evaluates individual x, where $m(f_j(x))$ reflects the average performance of individual x to meet the $(j-p)$th implicit index and $w(f_j(x))$ reflects the degree of uncertainty of the evaluation of the user, where both of them are precise values.

At least one of the following four cases will occur when the user evaluates the performance of individual x to meet the $(j-p)$th implicit index $f_j(x)$: the first one is that the user, with great confidence, is satisfied with individual x, indicating that it is the optimal solution with little uncertainty after decoding, and will assign x a larger value of $m(f_j(x))$ and a small value of $w(f_j(x))$; the second case is one in which the user, with only a small amount of confidence, is satisfied with individual x, indicating that it is the optimal solution with a great deal of uncertainty after decoding, and will assign x a larger value of $m(f_j(x))$ as well as of $w(f_j(x))$; in the third case the user, with great confidence, is unsatisfied with individual x, indicating that it is not an optimal solution with only a small amount of uncertainty after decoding, and will assign x with both a small value of $m(f_j(x))$ and a small value of $w(f_j(x))$; the last case shows that the user is unsatisfied with individual x with only a small amount of confidence, indicating that it is not an optimal solution with large uncertainty after decoding, and will assign x a small value of $m(f_j(x))$ but a large value of $w(f_j(x))$. Obviously, what we want is the optimal solution with little uncertainty, which is equivalent to maximizing $m(f_j(x))$ as well as minimizing $w(f_j(x))$.

By employing the same method of dealing with an interval objective as that in SubSection 3.1, we can transform the original optimization problem of maximizing $f_j(x)$ into the following single objective optimization problem:

$$\max \quad F_j(x) = \gamma\frac{m(f_j(x)) - \min m(f_j(x))}{\max m(f_j(x)) - \min m(f_j(x))} - (1-\gamma)\frac{w(f_j(x)) - \min w(f_j(x))}{\max w(f_j(x)) - \min w(f_j(x))},\quad \text{s.t.} \quad x \in \mathbf{S} \subseteq R^n. \tag{7}$$

As described in the previous subsection, $\min m(f_j(x))$ and $\max m(f_j(x))$ are the minimum and maximum values of $m(f_j(x))$ assigned by the user in some generation, $\min w(f_j(x))$ and $\max w(f_j(x))$ are the minimum and maximum degrees of uncertainty assigned by the user. These values change along with the optimization process. Similar to β, γ is the weight which reflects the preference of the decision-maker on $m(f_j(x))$ with a value between zero and one set, a *priori*, by the decision-maker, with $1-\gamma$ reflecting the weight of the preference of the decision-maker on $w(f_j(x))$. In contrast to Eq. (6) where the two terms are added, in Eq. (7) the second term is subtracted as a result of maximizing $m(f_j(x))$ as well as minimizing $w(f_j(x))$.

After dealing with these uncertainties, we can transform the original optimization problem expressed with Eq. (5) into the following optimization problem with precise parameters in explicit indices and precise implicit indices:

$$\begin{aligned} &\min F_1(x), F_2(x), \ldots, F_p(x),\\ &\max F_{p+1}(x), F_{p+2}(x), \ldots, F_{p+q}(x),\\ &\text{s.t. } x \in \mathbf{S} \subseteq R^n. \end{aligned} \tag{8}$$

On the one hand, Eq. (8) shows that each objective is expressed with either a function with precise parameters or a precise value, and infers that Eq. (8) can be solved by employing previous optimization algorithms for an optimization problem with precise indices, i.e., the reason for the transformation of the original indices. On the other hand, the number of objectives in Eq. (8) is equal to that in Eq. (5), but their meanings have changed. Therefore, it is necessary to calculate the values of these objectives using Eq. (5) when the optimal solutions of Eq. (8) have been obtained.

4. Evolutionary optimization with precise indices

To overcome the deficiencies of previous evolutionary algorithms with precise indices, we present an efficient evolutionary algorithm for this problem by combining traditional GAs with IGAs to improve the performance of IGAs in exploration, alleviating user fatigue, as well as simultaneously optimizing explicit and implicit indices. The proposed algorithm works as follows: first, a large population is used for evolution, where the number of population clusters is determined according to the similarity of individuals, changing along with evolution; on this basis the population is divided into several clusters using the K-means clustering method [23]. The computer calculates the values of all explicit indices of each individual, while the user only evaluates all implicit indices of each cluster center and expresses the evaluation results using mid-points and degrees of uncertainty. The values of all implicit indices of the remaining individuals are obtained by the similarity-based estimation according to those of their cluster centers; in addition, the Pareto dominance is employed to compare the performance of different individuals.

There are two key problems to be solved, i.e., the determination of the number of clusters of a large population and the similarity-based estimation of the implicit indices of an individual.

4.1. Number of clusters of large population

We determine the number of clusters of a large population according to the distribution of individuals. At the initial phase of the evolution, the degree of similarity among individuals is small and the population is diverse. Therefore the number of clusters should be large. Along with evolution, the degree of similarity among individuals increases, while the diversity of the population decreases, so does the number of clusters. Based on this analysis, we propose the following method of determining the number of clusters of a large population.

Assume that the genotype of an individual contains M gene meaning units [14] and denote N as the size of population $x(t)$ in the tth generation. In addition, assume that $x(t)$ is divided into $N_c(t)$ clusters, where $N_c(t)$ is smaller than or equal to $N_{\max}$, and $N_{\max}$ the largest number of individuals displayed by the human–computer interface. Denote the ith individual of $x(t)$ as $x_i(t)(i = 1,2,\ldots,N)$ with genotype being $x_{i1}x_{i2}\ldots,x_{iM}$, the degree of similarity between two individuals $x_i(t)$ and $x_j(t)$, denoted as $\alpha(x_i(t),x_j(t))$, can be expressed as [12]:

$$\alpha(x_i(t),x_j(t)) = \frac{1}{M}\sum_{m=1}^{M}\alpha_m(x_i(t),x_j(t)), \tag{9}$$

where

$$\alpha_m(x_i(t),x_j(t)) = \begin{cases} 1, & x_{im} = x_{jm}, \\ 0, & x_{im} \neq x_{jm}. \end{cases}$$

By denoting $A(x(t))$ as the degree of similarity of $x(t)$, we have:

$$A(x(t)) = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N}\alpha(x_i(t),x_j(t)). \tag{10}$$

Eq. (10) shows that the value of $A(x(t))$ is between zero and one. It should be observed as well that the more dissimilar the individuals in $x(t)$, the closer the value of $A(x(t))$ approaches zero, and therefore the larger the value of $N_c(t)$. If any individuals in the population are different from others, the value of $A(x(t))$ will be zero, in which case, the value of $N_c(t)$ is the largest, i.e., $N_{\max}$. In contrast, the more similar the individuals in $x(t)$, the closer the value of $A(x(t))$ approaches one, hence the smaller the value of $N_c(t)$. If all individuals in the population are the same, the value of $A(x(t))$ will be one. In this case, the value of $N_c(t)$ is the smallest, i.e., one. Therefore a candidate equation for calculating $N_c(t)$, reflecting this relation, is as follows:

$$N_c(t) = \lceil (A(x(t)) + N_{\max}(1 - A(x(t))))e^{-\frac{t}{T}} \rceil, \tag{11}$$

where T is the largest number of generations, $\lceil\cdot\rceil$ the upper integer function, and $e^{-\frac{t}{T}}$ is employed to adjust the number of clusters secondarily.

If the number of clusters is known a *priori*, the K-means clustering algorithm can be employed. For the large population employed in our study, we can determine the number of clusters according to Eq. (11). Therefore the K-means clustering algorithm can be employed to cluster this large population. The K-means clustering algorithm has some known deficiencies, e.g., it suffers from being very sensitive to the initial random selection of points. There are other more efficient clustering algorithms, such as the furthest point first clustering algorithm [13]. The performance of the proposed algorithm will be improved if we choose other more efficient clustering algorithms. But this is beyond the scope of our work, and we expect to investigate this topic in the future.

4.2. Similarity-based estimation of implicit indices of individuals

The user only evaluates all implicit indices of $N_c(t)$ cluster centers in order to alleviate fatigue, while the computer calculates the values of all explicit indices of each individual. Therefore, all N individuals have the values of all explicit indices in $x(t)$, but only $N_c(t)$ cluster centers have the values of all implicit indices, while the remaining $N - N_c(t)$ individuals do not. In order to compare the performance of different individuals, it is necessary to estimate the values of all implicit indices of the remaining individuals according to the values of their cluster centers. The comparison in performance between different individuals is based on their precise fitness, transformed from the uncertain values in our study. Therefore, what to be estimated are the precise values of all implicit indices of the remaining individuals.

It is worth noting that what to be estimated are not some intervals that reflect the implicit indices of the remaining individuals, but some precise values transformed from these intervals. Therefore "precise values" are the opposite of "intervals", and obtained by estimation based on the values of the implicit indices of these cluster centers.

It is easy to see that the greater the similarity between the estimated individual and its cluster center, the closer the estimated values of the implicit indices approach the values of the implicit indices of its cluster center, and the more precise these estimates, and vice versa. Based on this, we propose the following method to estimate the implicit indices of an individual.

Denote the ith cluster of $x(t)$ as $\{c_i(t)\}$ with $c_i(t)$, $i = 1,2,\ldots,N_c(t)$ as its center, which is exactly the individual evaluated by the user. The precise values of the implicit indices of $c_i(t)$ are $F_{p+1}(c_i(t)), F_{p+2}(c_i(t)),\ldots,F_{p+q}(c_i(t))$. Assume that an individual $x_l(t)$, $x_l(t) \neq c_i(t)$, whose implicit indices are to be estimated, belongs to $\{c_i(t)\}$, where the estimations of the implicit indices are denoted as $\widehat{F}_{p+1}(x_l(t)), \widehat{F}_{p+2}(x_l(t)), \ldots, \widehat{F}_{p+q}(x_l(t))$. According to the degree of similarity $\alpha(x_l(t),c_i(t))$ between $x_l(t)$ and $c_i(t)$, the estimation of the j-th implicit index of $x_l(t)$ can be expressed as follows:

$$\widehat{F}_j(x_l(t)) = F_j(c_i(t))e^{-(1-\alpha(x_l(t),c_i(t)))}, \tag{12}$$

where j is equal to $p+1, p+2,\ldots,p+q$, and $F_j(c_i(t))$ is calculated by Eq. (7).

Via this process, the precise values of all explicit and implicit indices of an individual can be obtained. Based on these values, we can further compare the performance of different individuals by using the Pareto dominance, employ elitist strategy, and produce offspring by utilizing such traditional genetic operations as crossover and mutation.

4.3. Steps of algorithms

The steps of our algorithm are as follows:

Step 1 Set the values of evolutionary control parameters in the algorithm and initialize a population.
Step 2 Determine the number of clusters in the population according to the approach given in subSection 4.1, and then divide the population into $N_c(t)$ clusters by using the K-means clustering method.
Step 3 Decode each individual and evaluate all implicit indices of all cluster centers by the user. Estimate the values of all implicit indices of the remaining individuals in each cluster according to the approach presented in subSection 4.2, and calculate the values of all explicit indices of each individual by computer. Form the vector of fitness of an individual based on these values.
Step 4 Judge whether the termination criteria are met or not, if yes, go to Step 6.
Step 5 Compare the performance of different individuals by using the Pareto dominance, and perform genetic operations on the population to generate offspring. Go to Step 2.
Step 6 Stop the evolution and output the optimal solutions.

These steps show that our work is reflected in Steps 2 and 3. Once the number of clusters has been determined, it is easy to divide the entire population in some generation into several clusters by employing the K-means clustering method. The number of clusters is determined according to the similarity of individuals and the evolution process in Step 2. The purpose of estimating the implicit indices of a large number of non-center individuals is to decrease the number of individuals evaluated by the user, hence alleviating user fatigue. In order to guarantee the precision of these estimates, a similarity-based estimation strategy is employed in Step 3.

The pseudo-code of our algorithm is shown as Fig. 1.

Note that to improve the performance in solving complex optimization problems in real-world applications, our work can be incorporated in other efficient evolutionary strategies, e.g., adaptive crossover probability and mutation probability [7], niche technique and crowding strategy [21].

5. Application in interior layout problem

5.1. Problem description

Consider the interior layout design of a general flat (Fig. 2). The flat is composed of three bedrooms, a sitting room, a kitchen, a toilet and one aisle. Since the width W and the length L of the flat are given a *priori*, the area of the flat can be easily calculated. The problem is to assign each part with an appropriate width and length in order to minimize the total cost of the flat and to meet the aesthetical requirements of the user.

This is a typical optimization problem with hybrid indices that contains both one explicit index and one implicit index. The explicit index is the total cost of the flat which is the sum of the cost of all parts. The cost per unit area of each part, affected by the relationship between market supply and demand, is not a precise value but fluctuates within a certain range, making the total cost of the flat to be an objective with interval parameters, denoted as $f_1(x,c)$. The implicit index is to meet the aesthetical requirements of the user, denoted as $f_2(x)$. Since $f_2(x)$ cannot be expressed with a well-defined function and cognition of the user on the object is uncertain, $f_2(x)$ is expressed with an interval to reflect this characteristic.

```
/*input: the values of evolutionary control parameters*/
/*output: optimal solutions*/
begin
    t=1
    initialize a population
    while (not termination) do
        calculate the number of clusters N_c(t)
        divide the population by using the K-means clustering method
        decode each individual
        evaluate all implicit indices of all cluster centers by the user
        estimate the values of all implicit indices of the remaining individuals
        calculate the values of all explicit indices of each individual
        compare all individuals
        perform the following genetic operators
            selection
            crossover
            mutation
            t=t+1
    end while
    return the optimal solutions
end
```

Fig. 1. Pseudo-code of algorithm.

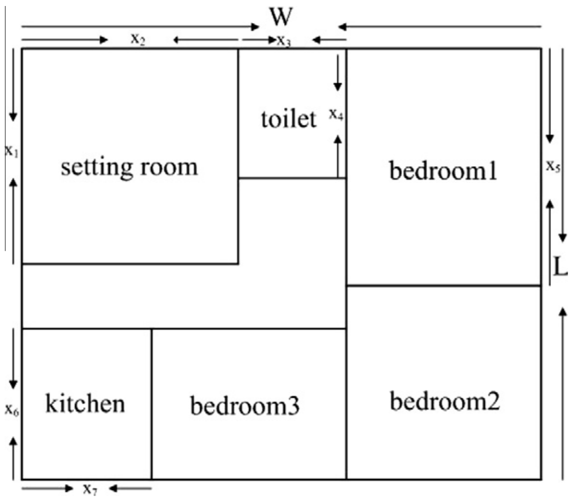

Fig. 2. Interior layout of flat.

The mathematical model of this problem is constructed as follows. The cost per unit area of different parts varies because of their special functions. Denote the cost per unit area of the sitting room, toilet, bedroom 1, kitchen, bedroom 3, bedroom 2 and aisle as $c_1, c_2, \ldots, c_7$, respectively, we have the following expression of the total cost of the flat:

$$C(x,c) = c_1x_1x_2 + c_2x_3x_4 + c_3(W - x_2 - x_3)x_5 + c_4x_6x_7 + c_5x_6(x_2 + x_3 - x_7) \\ + c_6(W - x_2 - x_3)(L - x_5) + c_7((x_2 + x_3)(L - x_1 - x_6) + (x_1 - x_4)x_3)),$$

where $x = (x_1, x_2, \ldots, x_7)^{\mathrm{T}}$, $c = (c_1, c_2, \ldots, c_7)^{\mathrm{T}}$, $c_i = [\underline{c_i}, \overline{c_i}]$, $i = 1, 2, \ldots, 7$. For convenient to illustration, we denote $f_1(x, c) = C(x, c)$.

If we denote x as the decision variable of the interior layout, the problem can be formulated as follows:

$$\begin{aligned}
\min \quad & f_1(x,c) = c_1x_1x_2 + c_2x_3x_4 + c_3(W - x_2 - x_3)x_5 + c_4x_6x_7 + c_5x_6(x_2 + x_3 - x_7) + c_6(W - x_2 - x_3)(L - x_5) \\
& \qquad + c_7((x_2 + x_3)(L - x_1 - x_6) + (x_1 - x_4)x_3)) \\
& \max \quad f_2(x) \\
\text{s.t.} \quad & \\
& x_1 \in \{4.0, 4.3, 4.6, 4.9, 5.2\} \\
& x_2 \in \{4.0, 4.3, 4.6, 4.9, 5.2, 5.5, 5.8, 6.1, 6.4, 6.7, 7.0, 7.3\} \\
& x_3 \in \{2.0, 2.3, 2.6, 2.9, 3.2\} \\
& x_4 \in \{2.0, 2.4, 2.8, 3.2, 3.6\} \\
& x_5 \in \{1.0, 2.0, 3.0, 4.0, 5.0\} \\
& x_6 \in \{2.6, 2.9, 3.2, 3.5, 3.8\} \\
& x_7 \in \{1.0, 2.0, 3.0, 4.0, 5.0\}.
\end{aligned} \tag{13}$$

The value of each decision variable of the problem is chosen from the corresponding set, composed of specific discrete values, because the width and the length of each part often have some discrete values in architectural standards.

Eq. (13) does not directly obtain the fitness of an individual, but needs to be converted as follows: for $f_1(x, c)$, a function with interval parameters c, we first obtain its upper and lower limits according to the values of its parameters and variables, calculate the mid-point and the radius of $f_1(x, c)$ with Eq. (2), and finally obtain the value of the explicit index with Eq. (6). For $f_2(x)$ of a cluster center, we first obtain its mid-point and degree of uncertainty based on evaluation of the user, and then calculate the value of the implicit index from Eq. (7); in addition, we obtain the values of the remaining individuals according to the similarity-based estimation strategy presented in subSection 4.2.

5.2. Parameter setting and individual coding

In this application, the width W of the flat is 12.5 m and its length, L, is 10 m. The cost per unit area of each part is an interval, with the value of c_1 set between 800 and 900. Similarly, the value of both c_2 and c_4 is set between 900 and 1100, that of c_3, c_5 and c_6 between 600 and 700, and that of c_7 between 400 and 600. In order to enhance the performance of the algorithm in exploration, N, the size of the population, is equal to 200. Given the limitation of the human–computer interface, the value of $N_{\max}$, the largest number of individuals displayed, is 12. The probabilities of crossover and mutation are 0.95 and 0.01, respectively. The largest number of generations, T, is 15. An individual, encoded with some real values and its genotype is composed of all decision variables, is denoted as $x_1x_2x_3x_4x_5x_6x_7$. Given that different variables stand for specific phenotypes, M is 7. The number of possible choices of decision variables x_1, x_2, x_3, x_4, x_5, x_6 and x_7 are 5, 12, 5, 5, 5, 5 and 5, therefore the number of all possible interior layout designs is $12 \times 5^6 = 187{,}500$. The system looks for the layout that simultaneously meets the explicit index and the implicit index from these designs.

5.3. Genetic operators

The following genetic operators, i.e., tournament selection with size two, one-point crossover and one-point mutation, are employed. The performance of different individuals is compared according to the rank and the crowding distance of these individuals (for details, see [21]). Since an individual is encoded with real values, chosen from a corresponding set, containing several real values, the one-point crossover and the one-point mutation are similar to those of binary coding, i.e., the segments of the parents after the crossover point, randomly chosen, are exchanged with each other to generate their offspring in the one-point crossover operation; the gene of the parent in the mutation point, randomly chosen, is replaced by one of its alleles to generate an offspring in the one-point mutation operation.

5.4. Human–computer interface and evolutionary process

We developed an interior layout design system using Visual Basic 6.0. Its human–computer interface, shown as Fig. 3, includes the following four parts: the first one, at the top left corner, is a sample of the interior layout in which the name of its each part is shown. The second part at the bottom left, is a parameter setting with statistical information about the evolution, including the current generation, the number of individuals evaluated by the user and his time consumption, as well as the probabilities of crossover and mutation. The third part, occupying a large space on the interface, is the phenotype of cluster centers and the values of their indices, where the value of the explicit index reflects the total cost of an individual (after normalization). The smaller the value, the lower its total cost. The mid-point and the degree of uncertainty assigned by the user are chosen from the set that only contains values from 100 to 900 by increments of 100 and a set of integers with element between zero and 100. The fourth part, found at the bottom right corner, contains three command buttons for an evolutionary population, i.e., "Initialize", "Next Generation" and "End".

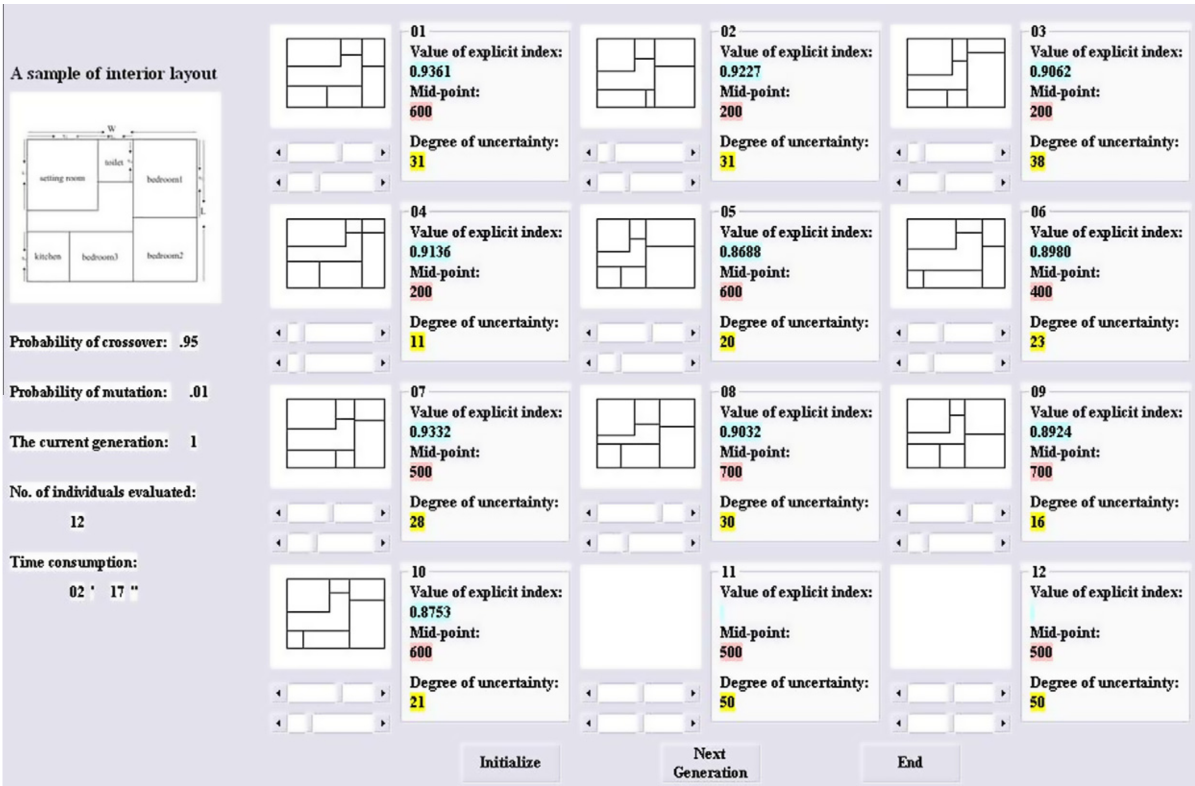

Fig. 3. Human–computer interface.

After clicking the "Initialize" button by the user, the system will randomly generate an initial population, select the number of clusters to be $N_{\max}$, and divide the initial population into $N_{\max}$ clusters by employing the *K*-means clustering method. Then each cluster center is displayed through the interface for the evaluation of the user. After the value of the implicit index of each cluster center has been assigned by the user, the values of the remaining individuals will be estimated according to Eq. (12). The value of the explicit index of each individual is automatically calculated by the system. After the evolution of $t-1$ generations and when the user clicks the "Next generation" button, the system will compare the performance of the individuals in the $(t-1)$th generation using the Pareto dominance, perform genetic operations to generate the population in the tth generation, calculate the number of its clusters, $N_c(t)$, divide the population into $N_c(t)$ clusters, and display each new cluster center to the user. The system will iterate this process until the user clicks the "End" button. At this moment, the system will stop the evolution and output the optimal solutions.

5.5. Experimental results and analysis

5.5.1. Effect of weights on performance of optimal solutions

We transform the original optimization problem with both interval parameters in explicit indices and interval uncertainties in implicit indices into a problem with both precise parameters in explicit indices and precise implicit indices by weighting the mid-point and the radius of each interval. Hence, the weights of the mid-point and the radius are important parameters, affecting the performance of the optimal solutions, i.e., the values of the explicit indices and those of the implicit indices. We only analyze the effect of different weights on the values of the explicit indices of the optimal solutions, since the user only evaluates each cluster center, while the values of the implicit indices of the remaining individuals are obtained by similarity-based estimation.

Let the value of β in Eq. (6) be 0.2, 0.4, 0.6 and 0.8, respectively. A user conducts five independent experiments for each value, records the mid-point and the radius of the explicit index of each Pareto optimal solution in each generation and then obtains their averages. Fig. 4(a) shows the curves of the mid-point and Fig. 4(b) those of the radius w.r.t. evolution.

Fig. 4 (a) shows that (1) for the same generation, the larger the value of β, the smaller the mid-point of the total cost of the Pareto optimal solution, and hence the better the average performance of the Pareto optimal solution; (2) for the same value of β, the mid-point of the total cost of the Pareto optimal solution gradually decreases along with evolution, indicating that the average performance of the Pareto optimal solution gradually improves.

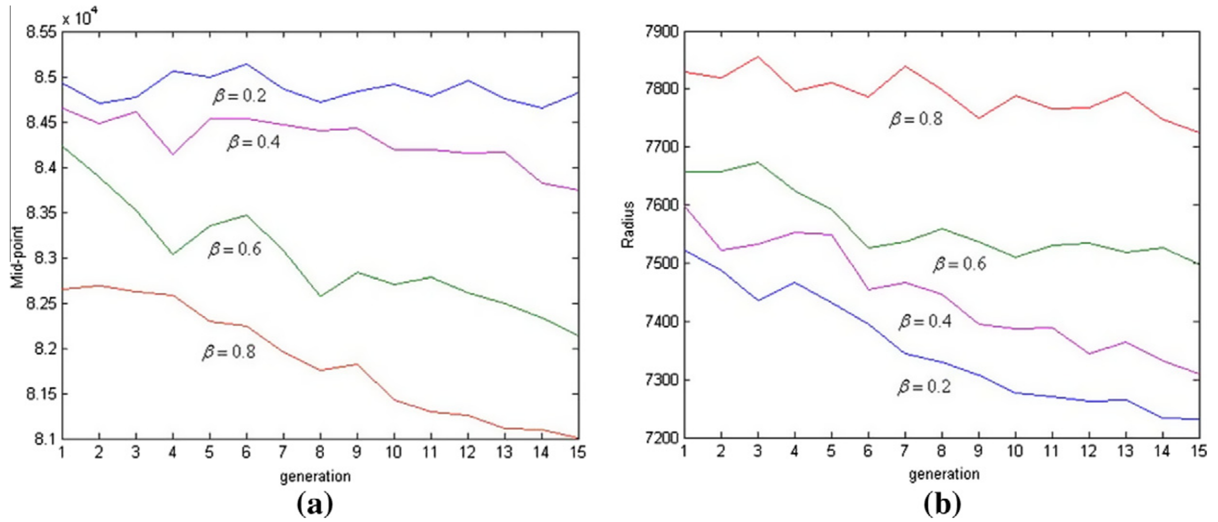

Fig. 4. Curves of mid-point and radius w.r.t. evolution: (a) depicts curves of mid-point w.r.t. evolution for different values of β and (b) curves of radius w.r.t. evolution for different values of β.

Fig. 4(b) shows that (1) for the same generation, the larger the value of β, the larger the radius of the total cost of the Pareto optimal solution, and hence the larger the degree of uncertainty of the Pareto optimal solution; (2) for the same value of β, the radius of the total cost of the Pareto optimal solution gradually decreases along with evolution, indicating that the degree of uncertainty of the Pareto optimal solution gradually decreases.

We conclude that the average performance of the optimal solution gradually improves and the degree of uncertainty of the optimal solution gradually decreases along with evolution, but both are affected by their weights.

5.5.2. Relationship between cognition of user and evolution

We now consider the rules of cognition of the user w.r.t. evolution. Let the value of γ in Eq. (6) be 0.2, 0.4, 0.6, and 0.8, respectively. A user performs five independent experiments for each value, records the mid-point and the degree of uncertainty of the evaluation of the user on each cluster center in each generation, and then obtains their averages. Fig. 5 (a) and (b) show the curves of the mid-point and the degree of uncertainty w.r.t. evolution.

Fig. 5 shows that (1) at the initial phase of the evolution, the mid-point of the implicit index of the cluster center assigned by the user is small, and the degree of uncertainty is large, indicating that the average performance of the cluster center is not good, nor the confidence of the user on the evaluation result; (2) the mid-point of the implicit index of the cluster center

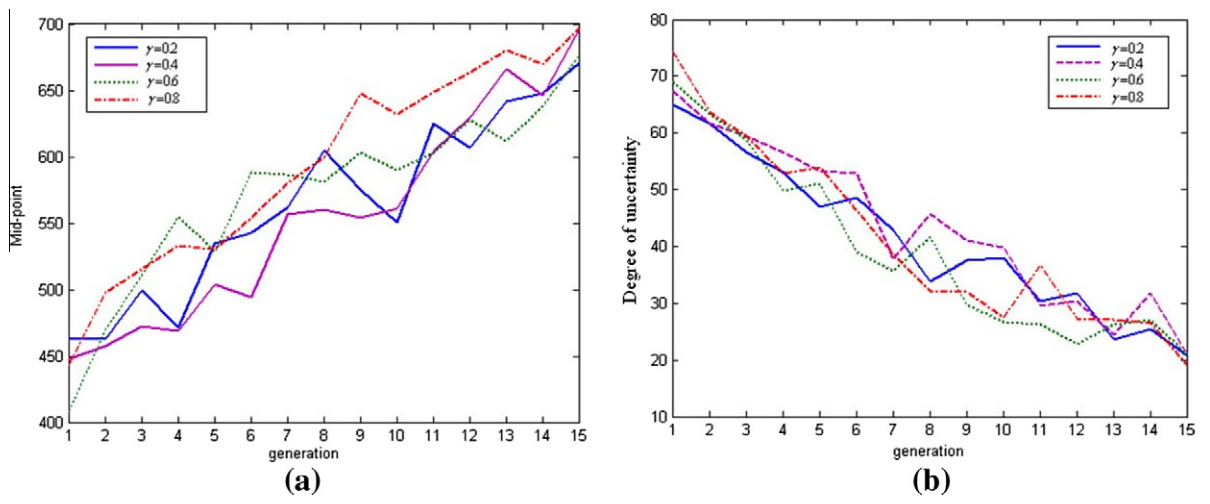

Fig. 5. Curves of mid-point and degree of uncertainty w.r.t. evolution, where (a) depicts curves of mid-point w.r.t. evolution for different values of γ and (b) depicts curves of degree of uncertainty w.r.t. evolution for different values of γ.

assigned by the user gradually increases along with evolution, while the degree of uncertainty gradually decreases, showing that the individual found by the system is improving, and the cognition of the user on the object becomes clearer.

We conclude that it is reasonable to consider the uncertainty of the evaluation results in the implicit index, and the rule of cognition of the user is, to a considerable extent, reflected by the mid-point and the degree of uncertainty of the implicit index.

5.5.3. Performance of proposed algorithm in solving optimization problem with both precise parameters in explicit indices and precise implicit indices

We validate the performance of the proposed algorithm in solving an optimization problem with both precise parameters in explicit indices and precise implicit indices by comparing it with multi-objective IGA, a typical and effective algorithm to solve this problem [1]. For multi-objective IGA, a small population is employed when two kinds of indices are optimized, the user evaluates all individuals in each generation, and the comparison among different individuals is based on the Pareto dominance. Since our algorithm is different from multi-objective IGA, we use a large population, carry out clustering according to the distribution of individuals, and use the similarity-based estimation of the implicit indices of an individual. The user only evaluates each cluster center whose number changes along with evolution.

Given the limitation of the human–computer interface, the population size of multi-objective IGA is set to 12, and the values of other parameters are the same as those in subSection 5.2. Let the values of both β and γ be 0.5. The termination criterion of the algorithm is that the number of generations approaches the maximum number of generations, set a *priori*.

Firstly, we consider the effect of the two algorithms on user fatigue and their performance in exploration. A user runs each algorithm for five times independently, records the time consumption of the user, the number of individuals evaluated, the number of individuals searched, and then obtains their averages. The results are listed in Table 1.

Table 1 shows that (1) the time consumption of the user in our algorithm is 8’ 59”, which is about half of that in multi-objective IGA. This result derives from the relatively small number of individuals evaluated by the user in our algorithm, suggesting that our algorithm can greatly alleviate user fatigue; (2) the number of individuals evaluated by the user in our algorithm is 91.6, which is about half of that in multi-objective IGA, but the number of individuals searched in our algorithm is 3000, far more than 180 in multi-objective IGA. This is due to the larger population in our algorithm, which indicates a good performance of our algorithm in exploration.

Based on these results, our algorithm not only alleviates user fatigue, but also performs well in exploration for this application.

Secondly, we illustrate the performance of our algorithm in alleviating user fatigue by considering the relationship between the number of individuals evaluated by the user and the degree of similarity of the population. We average the number of individuals evaluated by the user and the degree of similarity of the population in each generation in five independent experiments and obtain the curves w.r.t. evolution (see Fig. 6).

Fig. 6 shows that (1) the degree of similarity of the population gradually increases along with evolution, from 0.180 in the first generation to 0.638 in the fifteenth generation; (2) the number of individuals evaluated by the user gradually decreases along with evolution, from twelve in the first generation to 2.4 in the fifteenth generation. The decrease of the number of individuals evaluated by the user suggests an alleviation of user fatigue by using our algorithm. In addition, the relationship between the number of individuals evaluated by the user and the degree of similarity, shown in Fig. 6, is in complete agreement with Eq. (11).

In the third place, we consider the number and the performance of the Pareto optimal solutions obtained by the two algorithms. A user conducts five independent experiments for each algorithm, records the Pareto optimal solutions, and averages them. The results are listed in Table 2.

Table 2 shows that (1) the number of Pareto optimal solutions obtained by our algorithm is 13.8, far more than 5.2 by multi-objective IGA, as a result of a larger population used in our algorithm, indicating that our algorithm provides more choices for the user; (2) the value of the explicit index of the Pareto optimal solution obtained by our algorithm is 0.9066, smaller than 0.9444, from the use of multi-objective IGA, while the value of the implicit index of the Pareto optimal solution obtained by our algorithm is 0.3887, greater than 0.3159 from multi-objective IGA, indicating that the performance of the optimal solution obtained by our algorithm is better than that by multi-objective IGA.

We conclude that our algorithm obtain more and better optimal solutions than multi-objective IGA.

Finally, we consider the distribution of the Pareto optimal solutions obtained by the two algorithms. Fig. 7 shows the distribution of the Pareto optimal solutions in the fifteenth generation, where Fig. 7(a) shows the Pareto optimal solutions

Table 1
Time consumption, No. of individuals evaluated by user and that of individuals searched.

Algorithms	Time Consumption	No. of individuals evaluated	No. of individuals searched
Our algorithm	8′59″	91.6	3000
Multi-objective IGA	17′18″	180	180

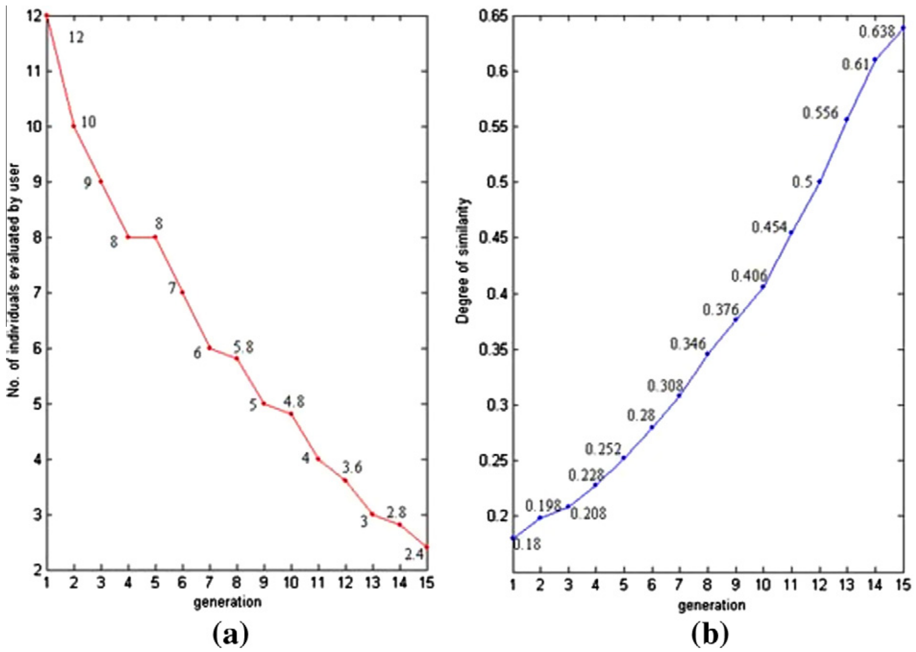

Fig. 6. Curves of No. of individuals evaluated by user and degree of similarity w.r.t. evolution; (a) depicts No. of individuals evaluated by user w.r.t. evolution and (b) depicts degree of similarity w.r.t. evolution.

Table 2
Pareto optimal solutions obtained by two different algorithms.

Algorithm	No. of optimal solutions	Value of explicit index	Value of implicit index
Our algorithm	13.8	0.9066	0.3887
Multi-objective IGA	5.2	0.9444	0.3159

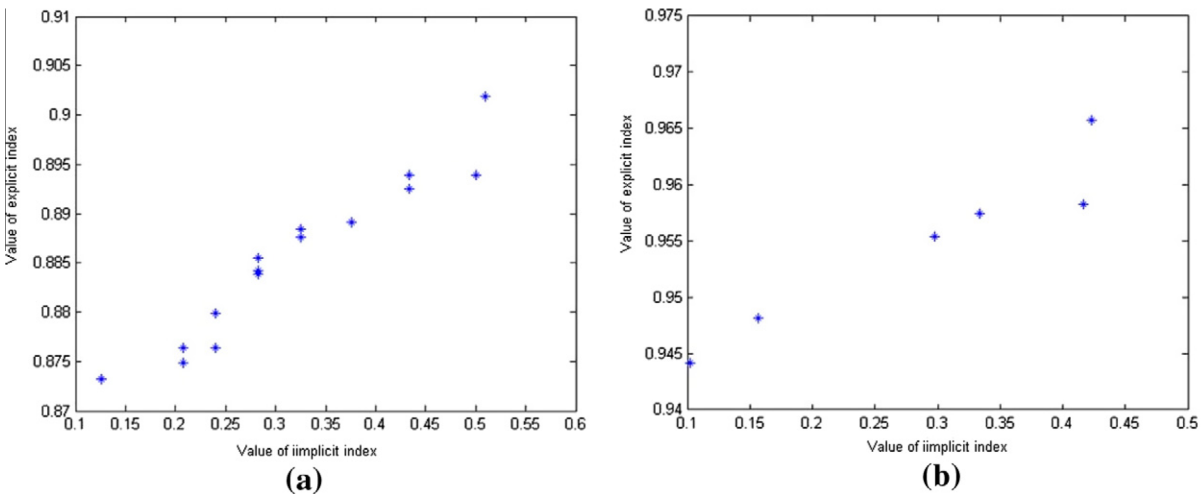

Fig. 7. Distribution of Pareto optimal solutions of our algorithm and multi-objective IGA: (a) depicts distribution of Pareto optimal solutions of our algorithm and (b) depicts distribution of Pareto optimal solutions of multi-objective IGA.

obtained by our algorithm, while Fig. 7(b) shows the solutions using multi-objective IGA. It is easy to see that the distribution of the Pareto optimal solutions obtained by our algorithm is more uniform than that from multi-objective IGA.

Based on these experimental results and analysis, our algorithm not only alleviates user fatigue, but also obtains better performance and more well-distributed Pareto optimal solutions.

6. Conclusions

Many optimization problems in real-world applications contain both explicit and implicit indices, where these indices usually contain uncertain information. The use of appropriate strategies to effectively tackle this uncertain information is very helpful in solving these optimization problems. These strategies have recently become popular research topics in information science.

Focusing on an optimization problem with both interval parameters in explicit indices and interval uncertainties in implicit indices, we propose a novel evolutionary optimization algorithm to effectively solve the problem. We first construct a mathematical uncertainty model of the optimization problem, and then transform the uncertain optimization problem into a problem with both precise parameters in explicit indices and precise implicit indices by using the method of interval analysis. To solve the transformed optimization problem effectively, we further present an approach to determine the number of clusters of a large population based on the distribution of the population, and then divide the entire population into several clusters by employing the K-means clustering method. The user only evaluates the values of the implicit indices of each cluster center, while those of the remaining individuals are obtained by the similarity-based estimation. The computer automatically calculates the values of the explicit indices of each individual. Based on the vector of the fitness of an individual, we use the Pareto dominance to compare the performance of different individuals.

In the experimental study, we apply the proposed algorithm to an interior layout problem, a typical optimization problem with both interval parameters in one explicit index and interval uncertainty in one implicit index. Our experimental results confirm that our algorithm not only enhances the performance of IGAs in exploration, hence improving the quality of the optimal solutions, but also alleviates user fatigue.

Our contributions are mainly embodied in the following three aspects. Firstly, we study the optimization problem with both interval parameters in explicit indices and interval uncertainties in implicit indices, construct its mathematical uncertainty model, and transform it into a precise one. Secondly, we present a novel evolutionary optimization algorithm to effectively solve this problem. In particular, we provide approaches to cluster a large population according to the distribution of individuals, and estimate implicit indices of an individual based on similarity. Thirdly, we apply the proposed algorithm to an interior layout problem, and validate its feasibility and efficiency empirically.

There are also some limitations in our work. For one, Eq. (9) only works for an individual assigned with a discrete code. If the code is continuous, Eq. (9) will not be applicable. The second limitation is that the values of β and γ are determined by the user according to his preference. To date, there is no appropriate method to choose these parameters. The third limitation is that various clustering methods have different effects on the performance of the proposed algorithm. In particular, the K-means clustering method has some deficiencies, e.g., it is sensitive to the initial selection of the cluster centers. To tackle these problems is our further research direction.

Acknowledgment

This work was jointly supported by National Natural Science Foundation of China, Grant No. 60775044 and Program for New Century Excellent Talents in Universities, Grant No. NCET-07-0802.

References

[1] A. Brintrup, J. Ramsden, A. Tiwari, Integrated qualitativeness in design by multi-objective optimization and interactive evolutionary computation, in: Proceedings of IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005, pp. 2154–2160.
[2] A. Brintrup, J. Ramsden, A. Tiwari, An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization, Computers in Industry 58 (2007) 279–291.
[3] A. Brintrup, J. Ramsden, A. Tiwari, Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms, IEEE Transactions on Evolutionary Computation 12 (3) (2008) 343–354.
[4] Z.Q. Cheng, L.K. Dai, Y.X. Sun, Feasibility analysis for optimization of uncertain systems with interval parameters, Acta Automatica Sinica 30 (3) (2004) 455–459.
[5] P. Funes, E. Bonabeau, J. Herve, Y. Morieux, Interactive multi-participant tour allocation, in: Proceedings of IEEE Congress on Evolutionary Computation, Portland, 2004, pp. 1699–1705.
[6] J.E. Fieldsend, R.M. Everson, Multi-objective optimisation in the presence of uncertainty, in: Proceedings of IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005, pp. 234–250.
[7] D.W. Gong, G.S. Guo, Interactive genetic algorithms with interval fitness of evolutionary individuals, dynamics of continuous, discrete and impulsive systems, Series B 14 (2007) 446–450.
[8] D.W. Gong, J. Yuan, X.Y. Sun, Interactive genetic algorithms with fuzzy individual fitness, computers in human behavior, doi:10.1016/j.chb.2010.10.012.
[9] G.S. Guo, J.F. Cui, Adaptive interactive genetic algorithms with grey for fitness of evolutionary individuals, Computer Applications 28 (10) (2008) 2525–2528.
[10] D.W. Gong, L. Tang, Phased evolutionary optimization for problems with hybrid indices, Journal of System Simulation 21 (19) (2007) 4936–4939.
[11] J.P. Guo, Y.H. Wu, Standard form of interval linear programming and its solution, System Engineering 21 (3) (2003) 79–82.

[12] D.W. Gong, J. Yuan, Large population size IGAs with individuals' fitness not assigned by user, Applied Soft Computing 11 (1) (2011) 936–945.
[13] T.F. Gonzalez, Clustering to minimize the maximum intercluster distance, Theoretical Computer Science 38 (2/3) (1985) 293–306.
[14] G.S. Hao, D.W. Gong, Interactive genetic algorithm based on landscape of satisfaction and taboos, Journal of China University of Mining and Technology 34 (2) (2005) 204–208.
[15] Z. Jiang, Optimization of uncertain systmems with interval parameters and its application in gasoline blending, Ph.D. dissertation, Zhejiang University, 2005.
[16] C. Jiang, X. Han, G.R. Liu, G.Y. Li, The optimization of the variable binder force in U-shaped forming with uncertain friction coefficient, Journal of Materials Processing Technology 182 (26) (2007) 262–267.
[17] Y.C. Jin, J. Branke, Evolutionary optimization in uncertain environments a survey, IEEE Transactions on Evolutionary Computation 9 (3) (2005) 303–317.
[18] C. Jiang, X. Han, F.J. Guan, Y.H. Li, An uncertain structural optimization method based on nonlinear interval number programming and interval analysis method, Engineering Structures 29 (2007) 3168–3177.
[19] R. Kamalian, H. Takagi, A.M. Agogino, Optimized design of MEMS by evolutionary multi-objective optimization with interactive evolutionary computation, in: Proceedings of Genetic and Evolutionary Computation Conference, Seattle, USA, 2004, pp. 1030–1041.
[20] R. Kamalian, Y. Zhang, H. Takagi, A.M. Agogino, Reduced human fatigue interactive evolutionary computation for micro-machine design, in: Proceedings of Fourth International Conference on Machine Learning and Cybernetics, Canton, China, 2005, pp. 5666–5671.
[21] D. Kalynmoy, A fast and elitist multiobjective genetic algorthm: NSGA-II, IEEE Transactions on Evolutionary Computation 2 (6) (2002) 182–197.
[22] P. Limbourg, D.E. Aponte, An optimization algorithm for imprecise multi-objective problem functions, in: Proceedings of IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005, pp. 459–466.
[23] J.Y. Lee, S.B. Cho, Sparse fitness evaluation for reducing user burden in interactive algorithm, in: Proceedings of 1999 IEEE International Fuzzy Systems Conference, 1999, pp. 998–1003.
[24] R. Martinez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Information Sciences 179 (13) (2009) 2058–2174.
[25] R.E. Moore, Methods and Applications of Interval Analysis, Prentice-Hall Inc., London, 1979.
[26] G. Soares, F. Guimaraes, C. Maia, J. Vasconcelos, L. Jaulin, Interval robust multi-objective evolutionary algorithm, in: Proceedings of IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp.1637–1643.
[27] R. Toscano, P. Lyonnet, Robust static output feedback controller synthesis using Kharitonov's theorem and evolutionary algorithms, Information Sciences 180 (10) (2010) 2023–2028.
[28] X.Y. Sun, D.W. Gong, Interactive genetic algorithms with individual's fuzzy and stochastic fitness, The Chinese Journal of Electronics 18 (4) (2009) 619–624.
[29] Y.L. Zheng, L.H. Ma, J.X. Qian, Tri-objective programming solution for uncertain parameters programming, Systems Engineering Theory and Practics 23 (5) (2003) 59–64.
[30] Y. Zhou, D.W. Gong, Y. Zhang, Evolutionary optimization methods for hybrid index optimization problems and application, Control and Decision 22 (3) (2007) 352–356.