Perception-based Evolutionary Optimization:

Outline of a Novel Approach to Optimization and Problem Solving

Alireza Rowhanimanesh   Mohammad-R. Akbarzadeh-T

Cognitive Computing Lab, Center for Applied Research on Soft Computing and Intelligent Systems (CARSIS), Ferdowsi University of Mashhad, Mashhad, Iran

Emails: rowhanimanesh@ieee.org, akbarzadeh@ieee.org

***Abstract*—Human perception and processing of information is granular and multi-resolution instead of numerical and precise. Due to this multi-resolution perception-based computing, human mind can quickly evaluate (calculate) the fitness of a large subspace of the search space. Indeed, this characteristic enables human to simplify and solve very complex problems. In contrast, evolutionary optimization (EO) as one of the most applied artificial problem solvers is based on computing with numbers since a chromosome is a single point of the search space and fitness function calculation is numerical. Hence, EO is blind towards the optimization landscape and this blindness inhibits its performance when the search space is very large and complex. Inspired by human perception based reasoning, a novel approach to optimization and problem solving is proposed here. Perception-based evolutionary optimization (PEO) is fundamentally based on computing with words. In PEO, chromosomes and fitness function calculation are perception-based (granular) instead of numerical and thus PEO works with granules (subspaces) rather than single points. Also, search is performed in a multi-resolution manner.**

***Keywords*—computing with words, multi-resolution perception-based optimization, perception-based chromosome, perception-based calculation of fitness function, perception-based evolutionary optimization (PEO).**

I. INTRODUCTION

Evolutionary optimization (EO) like the other existing optimization and problem solving methodologies is based on computing with numbers. In EO, a chromosome is a single point in the search space and fitness function calculation is numerical. As a result, if EO wants to calculate the fitness of a region of the search space, it must measure the fitness of many single points separately. In other words, EO is blind toward the optimization landscape and this blindness extremely inhibits its performance when the search space is very large and complex. Indeed, this blindness is fundamentally due to computing with numbers. In contrast, human mind works with perceptions since they are generally granular and multi-resolution instead of numerical and precise. One of the crucial characteristics of human mind is the concept of multi-resolution perception-based computing. The concepts of perception-based computing and multi-resolution computing have been separately and frequently described by Zadeh [1-9] and Albus and Meystel [10], respectively. Using this multi-resolution perception-based computing, human can easily calculate the fitness of a granule of solutions by only one look and simplify and solve very complex problems. In our related papers [11-12], we proposed a novel approach to rational exploration which is inspired by human-like rational search. In the mentioned approach, using computing with words and perceptions (CWP) which is Zadeh's brilliant work [1-9], first the search space is reduced as much as possible before the search begins. Then, conventional numerical EO begins to explore in the reduced search space. For many optimization problems, this methodology can notably decrease the size of the search space as well as complexity. Indeed, our proposed approach in [11-12] opens the door for applying CWP to optimization and problem solving.

As the next step of our research, in this paper, we are directly inspired by human cognition and propose perception-based evolutionary optimization (PEO) which is radically based on CWP instead of computing with numbers. The heart of PEO is the concept of multi-resolution perception-based problem solving which is one of the key features of human mind. In contrast to EO, in PEO, the chromosomes and fitness function calculation are perception-based and granular instead of numerical. Similar to human mind, PEO can calculate the fitness of a subspace of the search space just in one fitness calculation.

Someone may consider PEO as a method for uncertainty handling in optimization such as the work of Deb et al. in [13-14] that proposed an evolutionary framework for reliability-based optimization in which uncertainties in design variables and problem parameters are considered in optimization and a reliable optimal solution is achieved. The main differences between PEO and other works like the one of Deb [13-14] are as follows. First, PEO is inherently granular and perception-based even for certain problems where there is no uncertainty in decision variables and parameters. This means that PEO benefits granular and multi-resolution computing for improving the performance of search not for uncertainty handling. However, PEO has enough potential to be considered as an approach for uncertainty handling in optimization. But in this paper, reliability-based optimization and uncertainty handling are not discussed. Second, PEO is based on computing with words rather than conventional mathematics and thus linguistic computing plays a key role in PEO. Chromosome (solution) in PEO is a linguistic word instead of a mathematical point. This is an advantage of PEO in comparison with the most of the existing optimization methods that are based on conventional mathematics. Because of this linguistic computing, human can understand what is done by PEO and moreover human and machine can cooperate in solving complex optimization problems.

II. PERCEPTION-BASED EVOLUTIONARY OPTIMIZATION (PEO): AN OUTLINE

A. The Concept of Perception-based Chromosome

Consider that $f(x_1, x_2, ..., x_n)$ is the fitness (objective) function and the goal of optimization is finding the global minimum of f. In conventional (numerical) EO, a chromosome (individual) is a single point of the search

978-1-4244-6588-0/10/$25.00 ©2010 IEEE

space like $x^* = [x_1^*, x_2^*, \ldots, x_n^*]$ and the fitness function evaluation is calculating the value of f at x^*. In contrast to conventional EO, in PEO, the chromosome and fitness function calculation are perception-based and granular. To clarify the concept of perception-based chromosome, consider a single-variable fitness function $y = f(x)$ which is represented in Fig. 1 (a).

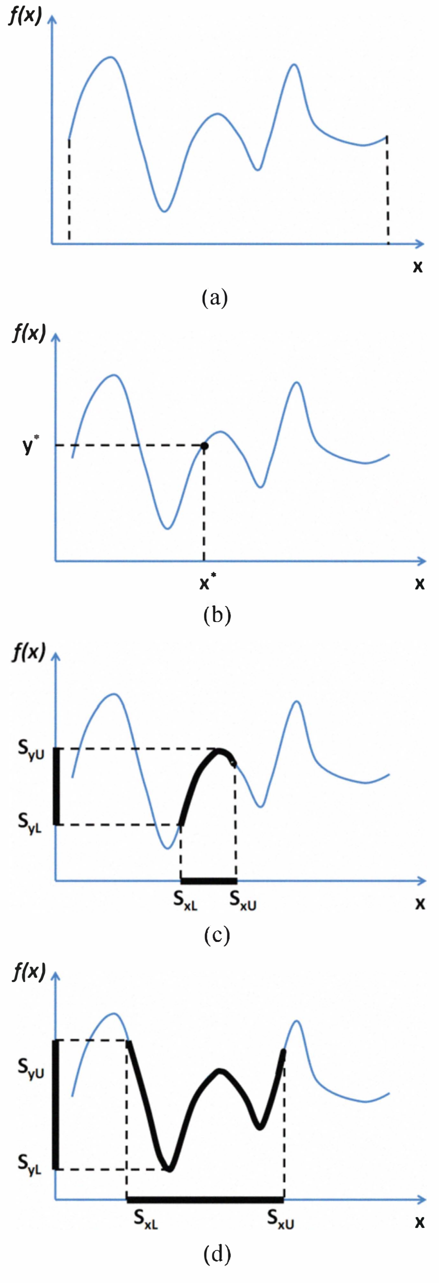

Figure 1. Numerical (b) vs. perception-based (c-d) chromosome and fitness function calculation. In this figure, the support sets of granules are just represented.

Suppose that we want to calculate the value of fitness function at *about* x^*. Fig. 1 shows three fitness calculations with different degrees of resolution. Fig. 1 (b) represents a numerical calculation of f since x is scalar x^* and the result is scalar y^* which is equal to $f(x^*)$. In fact, this is the method which is applied by conventional (numerical) EO for fitness function calculation and has the highest degree of resolution since the values of x and y are precise numbers. Now, let us consider perception-based chromosome and perception-based calculation of fitness function which are the pivotal components of PEO. A perception-based chromosome is a possibilistic constraint on the search space where possibilistic constraint is a specific and much applied mode of Zadeh's generalized constraint [8]. In other words, PEO works with the perceptions in the form of "x is A" where A is a linguistic and perception-based value such as large, about x^* or fairly low. Using CWP, such perceptions can be easily precisiated into possibilistic constraints. In this paper, a granule means a possibilistic constraint. It is clear that a crisp granule such as "x is [a b]" is a specific case of fuzzy granules. Fig. 1 (c) shows perception-based calculation of fitness function for "x is A" where $[S_{xL}\ S_{xU}]$ is the support set of fuzzy set A. Similarly, the result is "y is B" which is a possibilistic constraint on y instead of a precise number, where $[S_{yL}, S_{yU}]$ is the support set of fuzzy set B. This implies that using a perception-based (granular) chromosome, in each fitness calculation, $f(x)$ can be described in a subspace of the search space instead of a single point. The details of perception-based calculation of fitness function are described later in Subsection II.C. Fig. 1 (d) represents another perception-based chromosome whose resolution is lower than Fig. 1 (c). Hence, $f(x)$ is described in a larger subspace of the search space. In this simple example, a chromosome has only one gene. Similarly, for an n-variable perception-based chromosome which has n genes, each gene can be considered as a single-variable possibilistic constraint on its own domain. The result is a perception-based chromosome which is an n-variable possibilistic constraint on the original search space. We highly recommend triangular and trapezoidal parametric membership functions for fuzzy granules since they can significantly simplify the perception-based calculation of fitness function.

Fig. 2 (a) shows the search space of $f(x)$ which has been partitioned by three identical low resolution granules. Indeed, using this low resolution perception-based partitioning, the continuous search space of $f(x)$ including infinite number of precise points can be simply converted to a very small discrete search space with only 3 perception-based members. Using perception-based calculation of fitness function for these three granules, it can be easily understood that the global minimum is in G_1. As a result, the size of the search space is reduced three times. Now, the support set of G_1 is considered as a new search space and the mentioned process is repeated for it as shown in Fig. 2 (b). Similarly, it can be understood that the global minimum is in G_{13} which is one ninth of the original search space and has higher resolution rather than G_1. It is very notable to know that G_{13} is found after at most six perception-based calculation of fitness function. As a matter of fact, this multi-resolution perception-based granulation is a novel approach to drastically decrease the complexity of problems. It is clear that for a very large and complex search space, the number of granules may be too large and we need an efficient methodology to search in the space of them. In PEO, by means of numerical genotype, this important task is allocated to conventional related. The next subsection explains the details.

B. Genotype, Phenotype and Evolution-based Operators in PEO

Assume that the first gene is corresponding to x_1 and the domain of x_1 is partitioned by k granules. Assign a unique natural number between 1 and k to each of these granules and repeat this work for other variables. Now, each granule is coded by a number. Herein, we use these codes as genotype and the corresponding granules form the phenotype. As an example, if the second gene of a chromosome is equal to 3, this gene is interpreted as "x_2 is granule 3" since granule 3 is the third granule on the domain x_2. Therefore, the genotype of PEO is numerical while its phenotype is perception-based. In other words, the interpretation function maps integer numbers (codes) to perception-based granules. Using this simple numerical genotype, the evolution-based operators of PEO such as selection, reproduction, crossover, mutation and elitism can all be those which are used for conventional real-valued or binary EO. In case of fuzzy granulation, only the fitness ranking and selection process may be a little different with conventional EO. This is discussed in the next subsection. Generally, the major differences between PEO and conventional (numerical) EO are in interpretation function, phenotype, fitness function calculation and the concept of multi-resolution perception-based optimization.

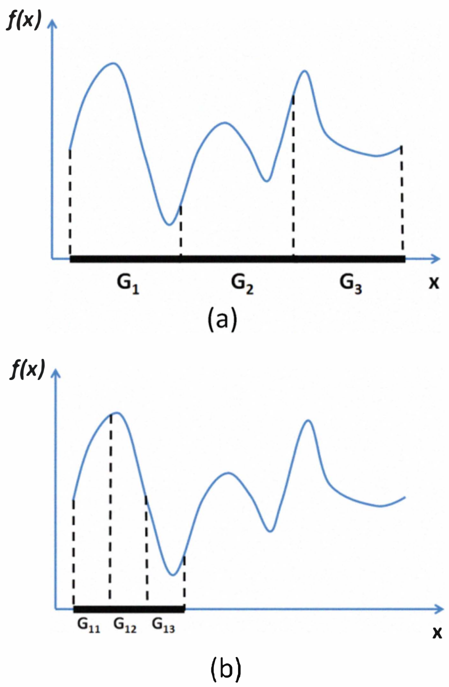

Figure 2. The concept of multi-resolution perception-based optimization. This figure only represents the support sets of granules.

C. Perception-based Calculation of Fitness Function

Due to the chromosomes (phenotype) of PEO are perception-based and granular, we radically require a different kind of computing for perception-based calculation of fitness function. Hence, PEO is fundamentally based on CWP instead of computing with numbers. Perception-based calculation of fitness function aims to compute the granular value of Y when the granular value of X is given. Fortunately, the extension principle (EP) which is one of the most crucial deduction rules of CWP can deal with this problem. In this paper, EP plays a pivotal role in perception-based calculation of fitness function and it is considered more technically below. As we have already discussed in [11, 12], we assume that f (an n-variable fitness function mentioned earlier in Subsection II.A) is explicit which means that it is represented as a mathematical formula. It is important to note that if f is not inherently explicit, it can be represented as a mathematical formula using fuzzy systems or neuro computing. According to [11, 12], any mathematical formula is composed of two fundamental components. First are single-variable functions such as e^x, $sin(x)$, x^r *and* $x^r cos(x+e^x)$. Second are the basic arithmetic operators including +, -, × and ÷. From the viewpoint of solving the extension principle, there are two types of fitness functions. We introduce these types and consider the complexity of EP for each type separately.

$$
\begin{cases} f(X_1) \ is \ A_1 \\ g(X_2) \ is \ A_2 \\ f * g(X_1, X_2) \ is \ B_1 \end{cases}
$$

$$
* : +, -, \times, \div
$$

$$
\mu_{A_1}(f(X_1)), \mu_{A_2}(g(X_2)) : known
$$

$$
\mu_{B_1}(f * g(X_1, X_2)) = ?
$$

$$
\mu_{B_1}(y) = \max_{f(X_1), g(X_2)} (\mu_{A_1}(f(X_1)) \wedge \mu_{A_2}(g(X_2)))
$$

$$
s.t. \ \ y = f * g(X_1, X_2)
$$

(a)

$$
\mu_{B_1}(y) = \max_{g(X_2)} (\mu_{A_1}(y \otimes g(X_2)) \wedge \mu_{A_2}(g(X_2)))
$$

$$
* : + \Rightarrow \otimes : -
$$

$$
* : - \Rightarrow \otimes : +
$$

$$
* : \times \Rightarrow \otimes : \div
$$

$$
* : \div \Rightarrow \otimes : \times
$$

(b)

$$
\begin{cases} f(X) \ is \ A_1 \\ g(X) \ is \ A_2 \\ f * g(X) \ is \ B_2 \end{cases}
$$

$$
* : +, -, \times, \div
$$

$$
\mu_{A_1}(f(X)), \mu_{A_2}(g(X)) : known
$$

$$
\mu_{B_2}(f * g(X)) = ?
$$

$$
\mu_{B_2}(y) = \max_{X} (\mu_{A_1}(f(X)) \wedge \mu_{A_2}(g(X)))
$$

$$
s.t. \ \ y = f * g(X)
$$

(c)

Figure 3. Zadeh's extension principle (EP). a: For Type I fitness functions, b: The simplified form of (a), c: For Type II fitness functions.

1. Type I Fitness Functions

The fitness function f is Type I if it can be represented as combination of independent terms where the combining operators are the basic arithmetic operators. If f is not Type I, it is Type II. For example $f(x_1, x_2, x_3) = f_3(f_1(x_1) \, \Delta_1 \, f_2(x_2)) \, \Delta_2 \, f_4(x_3)$ is a Type I fitness function since the operands of Δ_1 and Δ_2 are independent from each other. In contrast, $f(x_1, x_2) = f_3(f_1(x_1) \, \Delta_1 \, f_2(x_2)) \, \Delta_2 \, f_4(x_2)$ is Type II since the operands of Δ_2 are both dependant to x_2. In these examples, f_i is a single-variable function and Δ_i is a basic arithmetic operator. Fig. 3 represents Zadeh's extension principle (EP) for these two types. It should be noted that Zadeh's EP is only computationally efficient for Type I functions. For this type, using the simplified form of Fig. 3 (b), calculating $\mu_{B1}(y)$ leads to a 1-dimensional maximization which can be easily solved symbolically or numerically. Due to the lack of space, we highly recommend the readers to refer to our related papers in [11, 12] for further details and examples.

$$\begin{cases} f(X) \ is \ A_1 \\ g(X) \ is \ A_2 \\ f * g(X) \ is \ C \end{cases}$$

$$* : +, -, \times, \div$$

$$\mu_{A_1}(f(X)), \mu_{A_2}(g(X)) : known$$

$$\underline{\mu_C(f * g(X)) = ?}$$

$$\mu_C(y) = \max_{f(X), g(X)} (\mu_{A_1}(f(X)) \wedge \mu_{A_2}(g(X)))$$

$$s.t. \ \ y = f * g(X)$$

(a)

$$\mu_C(y) = \max_{g(X)} (\mu_{A_1}(y \otimes g(X)) \wedge \mu_{A_2}(g(X)))$$

$$* : + \Rightarrow \otimes : -$$

$$* : - \Rightarrow \otimes : +$$

$$* : \times \Rightarrow \otimes : \div$$

$$* : \div \Rightarrow \otimes : \times$$

(b)

Figure 4. a: The new extension principle which we proposed in [11, 12] for Type II fitness functions, b: The simplified form of (a).

2. *Type II Fitness Functions*

Fig. 3 (c) shows the Zadeh's EP for Type II fitness functions. In most cases, calculating $\mu_{B2}(y)$ leads to solving an *n*-dimensional nonlinear programming for each sample of *y*. Undoubtedly, calculating $\mu_{B2}(y)$ may be too complex and time consuming for optimization problems with large number of variables. Hence, it is clear that Zadeh's EP is not computationally efficient for evaluating Type II fitness functions since the calculation process is online and must be executed once per chromosome. In [11-12], we proposed a new EP for Type II fitness functions which is represented in Fig. 4. We also proved that the proposed EP is a conservative approximation of Zadeh's EP which implies that $\mu_{B2}(y)$ in Fig. 3 is a fuzzy subset of $\mu_C(y)$ in Fig. 4. Indeed, the proposed EP is very similar to Zadeh's EP for Type I fitness functions in Fig. 3 (a and b). Also, the computational complexity of the proposed EP is as low as Zadeh's EP for Type I fitness functions. In this paper, we use this EP for perception-based calculation of Type II fitness functions.

Regarding the EP, the result of perception-based calculation of fitness function is a fuzzy granule on *y*, $\mu_{B1}(y)$ for Type I and $\mu_C(y)$ for Type II. For simplicity, let us use $\mu(y)$ as the notation for the result which may be $\mu_{B1}(y)$ or $\mu_C(y)$. As an important point, $\mu(y)$ may need to be defuzzified for fitness ranking and selection process. Although there may exist various ideas for this deffuzification, we describe our own idea below. Let S_{yL} be the minimum of the support set of $\mu(y)$. Some examples for S_{yL} can be seen in Fig. 1. Given $\mu(y)$ for each chromosome, a better individual is a chromosome whose S_{yL} is lower. If this value is identical for several chromosomes, a chromosome which has higher membership degree (in that value) is preferred. It should be noted that fuzzy granules are very useful when we aim to incorporate human knowledge in optimization or make the optimization process human consistent [11, 12]. In case of an ordinary optimization such as the simulation examples of Section 3, crisp granulation may be preferred since it can considerably simplify the perception-based calculation of fitness function.

D. *The Concept of Multi-resolution Perception-based Optimization*

In this subsection, we introduce the novel and fundamental concept of multi-resolution perception-based optimization which is the heart of PEO. Assume that M^* is the desired value of the fitness function. Let $M = M^*$. In case that M^* is unknown, M is considered as a larger approximation of M^*, i.e. $M \geq M^*$. As described in Subsection II.A and Fig. 2, in PEO, the search space is first granulated (partitioned) into low-resolution granules. Using numerical genotype, a random initial population is created. With the aid of interpretation function, perception-based phenotypes are interpreted. Using the methodologies of Subsection II.C, the fitness function can be calculated for each individual. Remember that each individual is a granule. For an individual, if S_{yL} is larger than M, it implies that no desired solution exists in this individual (granule). Thus, a large number proportional to S_{yL} is assigned to this individual as its fitness. If S_{yL} is smaller than M, this individual is submitted to the lower level to be searched by higher resolution granules in order to find more accurate (higher resolution) solution. In this new level, the support set of the selected individual (granule) is considered as the new search space and a new PEO begins to search in this new search space. It should be noted that this new search space is significantly smaller than the original search space. The same process is repeated for this new level. If the condition ($S_{yL} \leq M$) is satisfied by a chromosome of this level, similar to the previous level, this chromosome is submitted to the lower level and this process can be repeated through multiple levels. At the last level, a numerical optimization method such as conventional EO can be employed to find a numerical solution. In this paper, we use conventional EO for the bottom level. Indeed, the size of the search space is highly decreased through the levels. Hence, as the last level, the search space of the conventional EO is small enough.

Here is a notable difference between Type I and Type II fitness functions. For Type I, if the condition $S_{yL} \leq M$ is satisfied by an individual, the desired solution certainly exists in this individual since EP can be calculated precisely. In this case, the optimization process is sequentially submitted to the lower levels and the upper levels are stopped forever. In case of Type II, the story is different. If the condition $S_{yL} \leq M$ is satisfied by an individual, it is only probable that the desired solution exists in this individual. In fact, there is not any guaranty. This uncertainty is due to the EP since it is calculated approximately not precisely for Type II fitness functions. Thus, the lower level may not find the desired solution in the selected individual since such solution may never exist there. In such case, the lower level is stopped, the best fitness of the lower level is assigned to this individual as its fitness and the optimization process is returned to the upper level.

Indeed, using this simple condition ($S_{yL} \leq M$), a large number of useless subspaces can be easily identified and not considered for high resolution search. As mentioned earlier, this is an efficient way which is used by human mind to simplify the complex problems. In PEO, top and middle levels are perception-based while the bottom level is numerical and precise. It should be noted that the number of middle levels highly depends on the problem and the type of fitness function. This topic is more explained in the simulation examples of the next section.

III. SIMULATION EXAMPLES

In this section, PEO is applied to two numerical benchmark optimization problems. For these simulation examples, PEO utilizes crisp granulation (a special case of fuzzy granulation) for top and middle levels and a conventional real-valued GA for bottom level. The performance of PEO is also compared with the same conventional real-valued GA. This conventional real GA has been used in [15] with minor changes.

It is important to mention that all of the previous sections consider fuzzy granulation. In this section, due to simplicity of the examples, crisp granulation is sufficient to solve them. More examples on perception-based calculation of mathematical functions using both crisp and fuzzy granulation can be found in our earlier works [11-12].

Table 1 represents two numerical benchmark optimization problems with various distinguishing characteristics which are discussed in [16]. It is clear that they have Type I fitness functions. For all levels of PEO, population size is 20, initial population is random, chromosomes are real-coded, crossover is single-point, elite count is 2 and mutation rate is fixed and 0.01. Only in Michalewicz's problem, the mutation rate of numerical GA is 0.05. For both problems, the generation size of the bottom level numerical GA is 100. The value of M is considered 0 and -19.63 for De Jong's and Michalewicz's problems, respectively. The Zadeh's EP is solved symbolically and due to simplicity of crisp granulation, the computational complexity of perception-based calculation is as low as the ordinary fitness calculation of numerical GA. For top and middle levels, the current search space is partitioned into 2^n crisp granules through partitioning the current domain of each variable into 2 identical crisp granules. The number of middle levels depends on parameter d since the last middle level is a level in which the length of the current domain of each variable is (equal to or) smaller than d.

Each problem was solved by PEO in 10 independent random runs. To compare PEO with numerical GA, in each run, the number of fitness calculations was calculated and then the numerical GA was executed with the same number of fitness calculations. It is important to note that this numerical GA is parametrically as same as the bottom level numerical GA. The results are represented in Table 2. For De Jong's problem, PEO could find solutions whose fitness is averagely 665 times better than the solutions of numerical GA. For Michalewicz's problem, PEO could always find a solution in the neighborhood of global minimum while numerical GA always converged to a local minimum. Indeed, these results demonstrate that the performance of PEO is significantly higher than conventional numerical EO.

TABLE 1. NUMERICAL BENCHMARK OPTIMIZATION PROBLEMS

Benchmark Problem	Fitness (Objective) Function and Search Space
De Jong 1	$F_1 = \sum_{i=1}^{n} x_i^2$
	$n = 50, -5 \leq xi \leq 5, i = 1{:}n$
Michalewicz	$F_2 = -\sum_{i=1}^{n} \sin(x_i).(\sin(\frac{i.x_i^2}{\pi}))^{2.m}, m = 10$
	$n = 20, 0 \leq xi \leq \pi, i = 1{:}n$

TABLE II. PEO IN COMPARISON WITH NUMERICAL (CONVENTIONAL) GA. Except d, all values are the mean and standard deviation (in the parentheses) of 10 independent random runs.

		De Jong's 1	*Michalewicz*
PEO	Best Fitness	0.0019 (2.6e-4)	-19.5553 (0.0493)
	d	0.0391	0.0491
Numerical GA	Best Fitness	1.2639 (0.4294)	-18.1774 (0.2100)
	Generation Size	505.7 (41.3)	288.9 (31.6)

IV. CONCLUSION

As a novel approach to optimization and problem solving, perception-based evolutionary optimization (PEO) is proposed in this paper. PEO is based on computing with words and perceptions instead of computing with numbers. The heart of PEO is the concept of multi-resolution perception-based problem solving which is directly inspired by human mind. In contrast to most of the existing optimization methods, in PEO, chromosome and fitness function calculation are perception-based rather than numerical and PEO searches through granules instead of single points. What is important to note is the basic idea of PEO that is the concept of multi-resolution perception-based problem solving. This concept plays a key role in improving the performance of optimizer especially in case of complex and large-scale problems. Thus, using this idea is not only limited to evolutionary optimization while it can be applied to other local and global optimization methodologies. Generally, PEO aims to open the door toward new approaches to perception-based optimization and problem solving but it is still in the first steps of development.

REFERENCES

[1] Zadeh, "Fuzzy Logic = Computing with Words", IEEE transactions on fuzzy systems, Vol. 4, No. 2, May 1996

[2] Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic", Fuzzy Sets and Systems 90, pp. 111-127, 1997

[3] Zadeh, "From computing with numbers to computing with words", IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, Vol. 45, No. 1, pp. 105-119, January 1999

[4] Zadeh, "A New Direction in AI, Toward a Computational Theory of Perceptions", AI magazine, Spring 2001

[5] Zadeh, "Precisiated Natural Language, Toward a Radical Enlargement of the Role of Natural Languages in Information Processing, Decision and Control", FSKD 2002, 1-3, 2002

[6] Zadeh, "Precisiated Natural Language, PNL", AI Magazine, Vol. 25, No. 3, 2004

[7] Zadeh, "Granular Computing, the Concept of Generalized Constraint-Based Computation", RSCTC 2006, pp. 12-14, 2006

[8] Zadeh, "Generalized Theory of Uncertainty (GTU), Principal Concepts and Ideas", Computational Statistics & Data Analysis 51, pp. 15 – 46, 2006

[9] Zadeh, "Toward Human Level Machine Intelligence, Is It Achievable? The Need for a Paradigm Shift", IEEE Computational Intelligence Magazine, August 2008

[10] J. S. Albus, A. M. Meystel, "Intelligent Systems: Architecture, Design, Control", Wiley, 2002

[11] A. Rowhanimanesh, M-R. Akbarzadeh-T., "Reducing the Search Space in Evolutionary Optimization Using Computing with Words and Perceptions", Proceedings of IFS2009, The 3rd Joint Congress on Intelligent and Fuzzy Systems, Yazd, Iran, July 2009

[12] A. Rowhanimanesh, M-R. Akbarzadeh-T., "Human-like Rational Exploration in Evolutionary Optimization Using Computing with Words", Proceedings of ICEE 2009, 17th Iranian Conference on Electrical Engineering, Tehran, Iran, May 12-14, 2009

[13] K. Deb, S. Gupta, A. Daum, J. Branke, A. Kumar Mall, D. Padmanabhan, "Reliability-Based Multi-objective Optimization Using Evolutionary Algorithms", IEEE Transactions on Evolutionary Computation, Volume 13, Issue 5, pp. 1054-1074, October 2009

[14] D. A. Daum, K. Deb, J. Branke, "Reliability-based optimization for multiple constraints with evolutionary algorithms", IEEE Congress on Evolutionary Computation: CEC 2007, pp. 911 – 918, 2007

[15] R. L. Haupt, S. E. Haupt, "Practical Genetic Algorithms", John Wiley & Sons, Inc., 2004

[16] M-R. Akbarzadeh-T., M. Davarynejad, N. Pariz, "Adaptive fuzzy fitness granulation for evolutionary optimization", International Journal of Approximate Reasoning, vol. 49, pp. 523-538, 2008