Swarm Intelligence for Solving Continuous Domain Problems: Survey

Fadl Dahan [A,B] , Muneer Hamid Ahmed Al-Hammadi[C]

[A] Computer Science Department, College of Computer and Information Sciences, King Saud University, Saudi Arabia
[B] Computer Science Department, Faculty of Computing and Information Technology, Al-Turbah Branch, Taiz University, Yemen
[C] Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Saudi Arabia

Abstract

Some of swarm intelligence algorithms are mainly devolved for solving discrete optimization problems such as ACO while other algorithm mainly developed for solving continuous optimization problems such as PSO. The development of optimization problems lead the researcher to prove that metaheuristics algorithm such as swarm intelligence is able to solve the global continuous optimization problems efficiently. In this work, we are going to investigate two of the swarm intelligence algorithms (ACO and PSO) which are used for solving continuous global optimization problems and the hybridization between both.

Keywords: *Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Continuous Global Optimization.*

I. Introduction

One of the most common tasks in our world is optimal solution searching for the problems. Optimization means that we have many solutions for specific problem, and we need to get the best solution where this solution in form of maximization or minimization based on the problems and given constraints. Optimization is a challenging part in the problems solving and it has many applications in different domains.

Generally, we can divide the optimization algorithm based on the operation method into two classes deterministic and stochastic algorithm [1]. Deterministic are used when we have a clear relationship between the problems and the possible solutions. However, when this relationship is too complicated or not obvious then we try the stochastic algorithm.

In general, optimization problems has several classes according to the problem characteristics; first characteristic is types of the search space, second types of used functions, and third the based on the problems constraints [2]. In this work, we will concentrate into the optimization problems with search space and constraints characters.

We have two types of optimization problems based on the search space characteristic local and global optimization. We have to find the optimal solution for local optimization in subset of the problem space while in global optimization we have to get the optimal solution for the whole space over many local solutions [1].

Global optimization has three types of problems: discrete or combinatorial, continuous, and mixed discrete-continuous optimization problems and these three types can be classified base on the constraints into constrained and unconstrained problems [3]. Combinatorial Optimization Problems (COPs) aims to find an optimal combination for problems where the problems divided into a finite parts and the COP algorithm attempts to find the optimal combination for these parts [2]. Time scheduling, graph coloring, routing problems, etc. are COP examples.

For continuous optimization problems, the used methods need to find the solution among infinite number of solutions. Many of the real word problems can be represented as continuous optimization problems.

Nature is fully of wonders, as it contains many phenomena that enrich the minds of researchers; there are many ideas that help researchers in solve problems inspired from the nature. One of these phenomena is the social insect, which is the base of swarm intelligence algorithm. There are many algorithms inspired from the social insect proposed to solve optimization problems such as one inspired from the behavior of ants while foraging in 1992 by Dorigo PhD thesis [4] then the formalizing of ACO into a framework in [5]. Another type of social insect algorithm is PSO that is introduced by kennedy and Eberhart in 1995[6]. PSO inspired from the behavior of flock of birds while finding the food. There are other algorithms inspired from the social insect such as bee algorithm, firefly algorithm, bat algorithm, bacterial algorithm, etc.

In this work, we are going to study how the researchers improve ACO and PSO algorithm to solve continuous global optimization problems. The rest of paper is organized as follows: researches used in this survey are described in section 2. Section 3 presents briefly the background theory. Section 4 contains general view about function benchmark for continuous algorithms evaluation. Discussion of the improvement of ACO and PSO for continuous domain is described in section 5. Section 6 contains the paper conclusion.

II. RELATED WORKS

Some of the swarm intelligence algorithms mainly developed for combinatorial optimization problems such as ACO while the other mainly developed for continuous optimization problems such as PSO. However, as we know that combinatorial optimization problems are one of the optimization problems. On the other hand, we have complicated optimization problems called continuous optimization problems. In this section, we are going to explore the researches, which are used to tackle the continuous optimization problems by using ACO and PSO.

A. ACO for Continuous Optimization

The first used of ACO for continuous problems optimization is proposed in 1995 [7] called Continuous ACO (CACO). This paper is a hybridization of genetic algorithm for global search and ACO extension for local search. Then in 2000, another algorithm for continuous domain developed by Monmarché et al. [8] based on Pachycondyla Apicalis Ants (API) where finding solution is based on parallel local search by individual ant and each ant has memory instead of utilizing pheromones.

Socha [9] introduced ACO extension to tackle the continuous and mixed optimization problems. Socha presented the core idea of his new method with implementation and compared the results with ants-based algorithm used to solve the continuous problems and other metaheuristic methods used for the same domain.

Seid et al. [10] introduced a new method based on ACO for global optimization of continuous domain. The proposed method differs from other ACO based methods because it is purely pheromone-based method. The advantage of the new method is that it has a few numbers of parameters to control. The proposed method is named Continuous Ant Colony System (CACS).

Authors in [3] showed how the ACO could be extended to work for continuous domain without changing in the original concept of ACO. In their paper, the authors denoted this extension for ACO by ACOℝ and presented the core idea for the new algorithm with implementation. In this algorithm, each ant constructs a solution, which is denoted by Gaussian Probability Density Function (PDF) with embedded pheromone. Because continuous problems are new field, they compared the algorithm performance not only for other ants' method and they compared it with other metaheuristic methods used for the same domain. It is worth mentioned that because ACOℝ concept is similar to original ACO proposed by Dorigo [5], ACOℝ can be used for solving the discrete-continuous optimization problems.

Kong et al. [11] introduced a new algorithm based on ACO called Direct Ant Colony Optimization (DACO), which aims to optimize function in continuous domain. The idea of DACO, stochastic solutions is generated according to normal distribution based on pheromone association with the variables in normal distribution.

Orthogonal designed method is used in [12] which it controls the pheromone deposit in effective way. The new method termed Continuous Orthogonal Ant Colony (COAC) helps ants in searching process about the solution in collaborative and effective way.

New inspired method from ACOℝ [3] termed Hybrid Ant Colony Optimization (HACO) was introduced in [13]. HACO is a population-based method where the probability density function (PDF) is generated dynamically.

A new algorithm based on hybridization method between GA and ACO developed by Ciornei et al. [14] named Genetic Algorithm with Pachycondyla Apicalis Ants (GAAPI). In this algorithm, API [8] is combined with real coded GA to solve problems in continuous domain with simple constrained or unconstrained functions.

A new extension for ACO introduced by Hu et al. [15] named Sampling ACO (SamACO). The proposed algorithm used sampling for continuous variable to convert the discrete ACO for continuous optimization problems. To do this task there are three steps, first candidate variables are generated and selected then according to these candidates the ants construct the solution, finally, the pheromone updated.

B. PSO for Continuous Optimization

Liang et al. [16] proposed a PSO variant named Comprehensive Learning PSO (CLPSO). In this variant, the authors improved the PSO performance by increase the velocity based on historical information.

A combination between PSO and spreadsheets solver used in [17] for solving continuous constrained optimization problems named PSOLVER. In this hybridization algorithm, spreadsheets solver used as local optimizer while PSO used as global optimizer i.e. PSOLVER is a local-global optimization algorithm where spreadsheets solver improves the PSO performance.

Example based learning algorithm is introduced in [18] where the authors focus on balancing between diversity and converge speed to overcome the PSOs algorithm shortcoming. The proposed algorithm called Example-based Learning PSO (ELPSO) is inspired from the social phenomenon by the many of elite examples, which have great effect on crowd.

A hybridization of PSO and Shuffling Frog Leaping Algorithm (SFLA) for optimizing continuous problems called Grouping- Shuffling PSO (GSPSO)[19]. GSPSO combines the advantages of PSO and SFLA where PSO has flight strategy while SFLA has Grouping- Shuffling strategy.

Gompertz Binary PSO (GBPSO) introduced in [20] where the PSO binary function (sigmoid function) was replaced by Gompertz function to improve the exploitation of binary PSO.

In [21] the author presented a PSO instance called PSO-2S for continuous domain optimization. PSO-2S aims to fix standard PSO stagnation problem without losing the fast convergence advantage. For doing this, PSO depends on two types of particle main and auxiliary where main particles uses find the best auxiliary particle while auxiliaries particle use for increase diversity.

C. Hybrid ACO – PSO

Both PSO and ACO are improved to use it for continuous optimization and each one has its own advantages and disadvantages. Few researches are combined between these advantages to improve the performance of each one.

An improvement for PSO for multimodal continuous optimization by using pheromone guide technique inspired from the ants developed in [22] named Particle Swarm Ant Colony Optimization (PSACO). Ants pheromone uses to help particle in position updating while particle doing global optimization.

On the other side, an improvement for ACO for solving continuous problems using PSO developed in [23] also named Particle Swarm Ant Colony Optimization (PSACO). In this algorithm, the authors solve the problems of local minimum in ACO by using the ability of PSO particle velocity to minimize the convergence time.

III. BACKGROUND THEORY

Generally, optimization problems can be defined mathematically as follows [24]:

$$\text{For given function } f: S \to \mathbb{R}, \text{we need to find } X^* \in S: \forall_{x \in s} f(X^*) \leq f(X)\ (\text{Minimization}) \text{or } f(X^*) \geq f(X)(\text{Maximization})$$

Where:

f: is the objective function.
S: is the search space.
X^*: is the optimal solution.
X: is vector of solution.

We can formally define continuous optimization problems as follows[3]:

A model $Q = (S, \Omega, f)$ of continuous optimization problems consists of:

S: search space over finite set of continuous decision.

f: objective function to be minimized or maximized.

Ω: a set of constraints among the variables.

The search space defines as:

$$\text{Set of continuous variables } X_i, i = 1, \ldots, n \text{ with values } v_i \in D_i \subseteq \mathbb{R}$$

In the above model, we aim to find solution $s \in S$ this solution should satisfy the constraints in Ω where in this case the problems are called constrained. However, if Ω is empty, we call it unconstrained. This solution can be a global optimum denoted by $s^* \in S\ iff: f(s^*) \leq f(s)(\text{Minimization}) \forall s \in S$

ACO is one of the social insect inspired algorithm appeared in 1992 by Dorigo PhD thesis [4]. Generally, ACO is inspired from the foraging behaviors of ants and it introduced for solving discrete optimization problems. The ACO framework shown in Figure 2[15].

While using ACO we have *m* ants and each one placed on different solution randomly at the beginning. The movement for next step for each ant determined according to the pheromone and visibility value using eq.(1).

$$P_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta}{\sum_{r \in allowed_k} \tau_{ir}^\alpha(t)\eta_{ir}^\beta} & if\ j \in allowed_k \\ 0, & otherwise \end{cases} \quad (1)$$

Where:

$\tau_{ij}(t) =$ pheromone intensity on (i, j).

$allowed_k$: is the set of remaining step.

α and β: pheromone vs. visibility balancing parameters.

$$\eta_{ij} = \frac{1}{cost_{ij}}$$

After selecting the next movement the ants should update the pheromone value using eq.(2).

Figure 2: ACO Framework

$$\tau_{ij}(t+n) = \rho.\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k \quad (2)$$

Where:

ρ: evaporation between 0 and 1.

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & , if\ (i,j) \in tour\ done\ by\ ant\ k \\ 0\ , & otherwise \end{cases}$$

L_k: ant k tour length.

Q: is constant.

PSO one of the populations based metaheuristic algorithm and population called particles. Particles achieve searching task by using knowledge sharing with each other where each agent updates its velocity toward the best solution based on its own experience (called *pbest*) and its neighbor experience (called *gbest*) [6]. Figure 1 shows the searching mechanism for PSO [25].

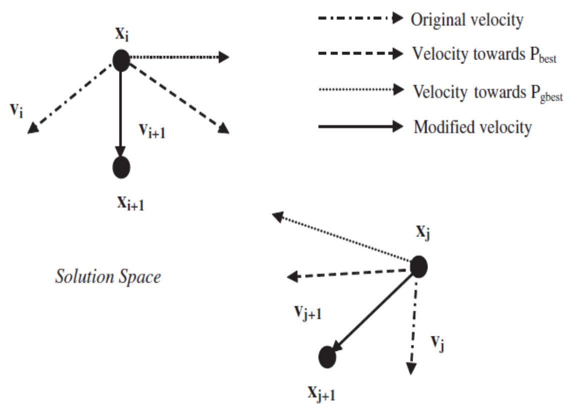

Figure 1: PSO Searching Mechanism.

The formal that governs the velocity vector is:

$$v_{t+1}^{i} = v_t^i + c_1 r_1 (pbest_i - x_t^i) + c_2 r_2 (gbest_t - x_t^i)$$

Where:

v_t^i: $velocity\ of\ particle\ i\ at\ iteration\ t,$

c_1, c_2: $weighting\ factor,$

r_1, r_2: $random\ number\ between$ [0,1],

x_t^i: $the\ current\ position\ of\ particle\ i\ at\ iteration\ t,$

$pbest_t$: $pbest\ of\ particle\ i,$

$gbest_t$: $gbest\ of\ particle\ j.$

While the formal that governs the position vector is:

$$x_{t+1}^{i} = x_t^i + v_{t+1}^i$$

Where:

x_t^i: $the\ current\ position\ of\ particle\ i\ at\ iteration\ t,$

v_{t+1}^i: $velocity\ of\ particle\ i\ at\ iteration\ t+1.$

IV. CONTINUOUS OPTIMIZATION BENCHMARK

Researches in continuous domain need to compare their algorithms results with other methods' results fairly. To do such task, the researchers simulate their algorithms on test function designed especially for continuous optimization problems called benchmark. However, the researchers need to take into account the test problems criteria such as initialization space, stop criterion , searching area etc. [26] and use the same test problems and the same criteria with compared algorithm.

Benchmarks provide great benefit for researchers because it helps them in [26]:

- Researchers don't need to implement benchmarks because they are usually implemented in many programming language and the researchers can easily link to one of them.
- No need for reprograming other algorithms to compare with which use the same benchmark they need only to comply with the same condition and criteria used by the.
- It is easy to compare the results because their presentation can be standardized.

Benchmarks for continuous optimization problems have several classes based on its properties [21, 27]:

- Single modal: the main characteristic of these problems are that they have slow convergence time for global solution and they are multidimensional problems. Examples of this type *(Sphere function, Axis parallel hyper-ellipsoid function, Rotated hyper-ellipsoid* function etc.).

Multimodal: the difficulty of this type of testing problems increases while the number of dimension and the local extreme increasing. It uses for testing the optimizer intelligence and quality. According to [25] this type of problems can be two-dimensional and multidimensional also small number of local minimum and huge number. Examples of this test functions *(Ackley's function, Langermann's function, Rastrigin's function, Rosenbrock's function etc.).*

V. DISCUSSION

In this section, we are going to discuss generally some aspects according to the ACO and PSO improvement to work with continuous domain, enhancing its performance, and the hybridization between both.

A. ACO Improvement

We know that the ACO mainly proposed for discrete optimization problems so for adapting ACO for continuous domain the researchers have to improve the ACO algorithm. According to the literature, we can summarize this improvement into following steps:

- Construct the Solutions

Solutions in traditional ACO constructed incrementally because the discrete space of the problems where each ant starts with empty or random solutions based on the problems then in each step the new solutions that are constructed randomly using probability are added or altered also based on the problems.

On the other hand, in continuous problems the domain changing so instead of using discrete probability distributions where in each step the ants choose one solution from the search space so we need to use continuous probability distribution [9] where the nest started randomly on the search space and the swarm of ants construct the solutions on this continuous probability distribution.

Table 1 contains the ACO algorithms mentioned in the literature with the continuous probability function, which are used.

TABLE I: Continuous ACO with probability function

Algorithm	Probability Function
CACO	Probability Density Function
API	Uniform Distribution
CACS	Pseudo-Random-Proportional
ACO_R	Gaussian Kernel
DACO	Probability Density Function
COAC	Probability Density Function
HACO	Gaussian Kernel
GAAPI	Uniform Distribution
SamACO	Probability Density Function

- Pheromone Update

Next step, how to update the pheromone in continuous problems which are the difference between the ACO methods. Pheromone represents the quality of the solutions in the construction step then ants update this pheromone to modify probability distribution towards best solutions. Pheromone updating traditionally has to actions: (a) reinforcement which leads to best solutions *(positive feedback)*, and (b) evaporation which means probability decreasing *(negative feedback).*

Positive feedback is done if additional probability distribution is added to the solutions.

Negative feedback in continuous domain has flexible way to update method. Her we will mention some examples according to the literature. First, negative updating opposite to positive updating where adding and removing is done in the same manner [12, 13]. Second, like the evaporation in canonical ACO where there are an evaporation weight [2, 3].

Also From the literature, we can classify the ant-based algorithms, which are used for solving continuous optimization problems into three types:

Pheromone based: in this type, we have points in the search space and ants deposit the pheromone on these points, which each one considers a complete solution. Generally, this type of algorithm is hybridization between ACO with other algorithm because we need to maintain the diversity. CACO algorithm [7] is an example for this type of ant-based for continuous problems where it is combined ACO with GA. COAC algorithm [12] also another example where it is a combination of ACO with orthogonal design method.

Without pheromone update: in this type the algorithms replace pheromone with other form such as memory in API algorithm [8] where each ant marks the place and uses memory to remember it. Also, API reused in [14] combined with GA.

ACO framework based: the algorithms in this type are developed as an extension for original ACO framework where each ant constructs the solution and uses pheromone to motivate searching. Socha algorithm [9] is an example of this type where he extend original ACO to tackle the continuous and mixed optimization problems. ACO_R [3] is an improvement of algorithm in [9]. DACO [11] also follows the ACO framework. SamACO [15] instead of using PDF sampling method which used in [3] the authors used candidate sampling and chose the solution from candidate variables.

B. PSO Improvement

Mainly PSO introduced for continuous optimization problems. However, according to the literature PSO suffers from being stuck into local optima because it is fast convergence. The main goal of researches on this area is to improve the PSO performance by avoiding this drawback with keeping the convergence speed. To solve this drawback the proposed algorithms in the literature have two types of solutions as follows:

- *Population management:* the researchers suggest to manage the population like using two population first one is temporary to initialize the first position of main population [21] or after *k* iteration the elite example used as initialization [18].
- *Hybridization:* this type uses other methods to help PSO such as SFLA in [19] or spreadsheets solver [17] to refine the PSO solutions and help PSO to avoid local minimum.

C. ACO and PSO improvement

To our best knowledge, no more researches in this part and we only have two researches in the literature. In this hybridization, the researchers aim to mix between the advantages of both to get better results for solving continuous optimization problems.

The advantage of ACO is that it works well in the local optimization problems while PSO works well in the global optimization problems [22].

In this type of researches, they used PSO to find the global best solution then turn to ACO by using special pheromone-guide mechanism to search in the global solution neighbors about better solution [22]. The contrary, in [23] they used the PSO to optimize the ACO solution.

VI. CONCLUSION

In this paper, we investigated a set of improvement for two of swarm intelligence algorithms to work with continuous global optimization problems. The swarm algorithms mentioned in this paper one of them introduced mainly for discrete problems optimization (ACO) while the other is for continuous optimization problems (PSO).

The cornerstone on ACO for continuous optimization problems are the way to represent and update the pheromone while in PSO is how to manage the population to avoid the PSO drawbacks or hybrid PSO with other optimization methods to enhance the PSO performance.

REFERENCES

[1] T. Weise, "Global optimization algorithms–theory and application," Self-Published, 2009.

[2] K. Socha, Ant Colony Optimisation for Continuous and Mixed-variable Domains: VDM Publishing, 2009.

[3] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," European Journal of Operational Research, vol. 185, pp. 1155-1173, 2008.

[4] M. Dorigo, "Optimization, learning and natural algorithms," Ph. D. Thesis, Politecnico di Milano, Italy, 1992.

[5] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, 1999.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995, pp. 1942-1948.

[7] G. Bilchev and I. C. Parmee, "The ant colony metaphor for searching continuous design spaces," in Evolutionary Computing, ed: Springer, 1995, pp. 25-39.

[8] N. Monmarché, G. Venturini, and M. Slimane, "On how Pachycondyla apicalis ants suggest a new search algorithm," Future Generation Computer Systems, vol. 16, pp. 937-946, 2000.

[9] K. Socha, "ACO for continuous and mixed-variable optimization," in Ant Colony Optimization and Swarm Intelligence, ed: Springer, 2004, pp. 25-36.

[10] H. Seid, Pourtakdoust, and N. Hadi, "An extension of ant colony system to continuous optimization problems," in Ant Colony Optimization and Swarm Intelligence, ed: Springer, 2004, pp. 294-301.

[11] M. Kong and P. Tian, "Application of ACO in continuous domain," in Advances in Natural Computation, ed: Springer, 2006, pp. 126-135.

[12] X.-M. Hu, J. Zhang, and Y. Li, "Orthogonal methods based ant colony search for solving continuous optimization problems," Journal of Computer Science and Technology, vol. 23, pp. 2-18, 2008.

[13] J. Xiao and L. Li, "A hybrid ant colony optimization for continuous domains," Expert Systems with Applications, vol. 38, pp. 11072-11077, 2011.

[14] I. Ciornei and E. Kyriakides, "Hybrid ant colony-genetic algorithm (GAAPI) for global continuous optimization," Systems, Man, and

Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 42, pp. 234-245, 2012.

[15] X.-M. Hu, J. Zhang, H.-H. Chung, Y. Li, and O. Liu, "SamACO: variable sampling ant colony optimization algorithm for continuous optimization," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 40, pp. 1555-1566, 2010.

[16] J. J. Liang, A. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," Evolutionary Computation, IEEE Transactions on, vol. 10, pp. 281-295, 2006.

[17] A. H. Kayhan, H. Ceylan, M. T. Ayvaz, and G. Gurarslan, "PSOLVER: A new hybrid particle swarm optimization algorithm for solving continuous optimization problems," Expert Systems with Applications, vol. 37, pp. 6798-6808, 2010.

[18] H. Huang, H. Qin, Z. Hao, and A. Lim, "Example-based learning particle swarm optimization for continuous optimization," Information Sciences, vol. 182, pp. 125-138, 2012.

[19] Y. Li, X. Dong, and J. Liu, "Grouping-Shuffling particle swarm optimization: an improved PSO for continuous optimization," in Advances in Swarm Intelligence, ed: Springer, 2010, pp. 86-93.

[20] P. Chauhan, M. Pant, and K. Deep, "Novel binary PSO for continuous global optimization problems," in Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011, 2012, pp. 169-176.

[21] A. El Dor, M. Clerc, and P. Siarry, "A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization," Computational Optimization and Applications, vol. 53, pp. 271-295, 2012.

[22] P. Shelokar, P. Siarry, V. Jayaraman, and B. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," Applied Mathematics and Computation, vol. 188, pp. 129-142, 2007.

[23] Y. Xiao, X. Song, and Z. Yao, "Improved Ant Colony Optimization with Particle Swarm Optimization Operator Solving Continuous Optimization Problems," in Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on, 2009, pp. 1-3.

[24] S. Boyd and L. Vandenberghe, Convex optimization: Cambridge university press, 2004.

[25] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, "Particle swarm optimization: Hybridization perspectives and experimental illustrations," Applied Mathematics and Computation, vol. 217, pp. 5208-5226, 2011.

[26] K. Oparaa and J. Arabasb, "Benchmarking Procedures for Continuous Optimization Algorithms," Journal of Telecommunications and Information Technology, pp. 73-80, 2011.

[27] M. Molga and C. Smutnicki, "Test functions for optimization needs," Test functions for optimization needs, 2005.