Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID GENETIC ALGORITHM WITH GREAT DELUGE TO SOLVE CONSTRAINED OPTIMIZATION PROBLEMS

NABEEL AL-MILLI
Financial and Business Administration and Computer Science Department
Zarqa University College
Al-Balqa' Applied University
E-mail: Nabeel.almilli@bau.edu.jo

ABSTRACT

In this paper, a new hybrid optimization algorithm based Genetic Algorithms (GAs) is proposed to solve constrained optimization engineering problems. A hybrid Genetic Algorithms (GA) and great deluge algorithm are used to solve non-linear constrained optimization problems. The algorithm works on improving the quality of the search speed of GAs by locating infeasible solutions (i.e. chromosomes) and use great deluge to return these solutions to the feasible domain of search; thus have a better guided GA search. This hybridization algorithm prevents GAs from being trapped at local minima via premature convergence. The simulation results demonstrate a good performance of the proposed approach in solving mechanical engineering systems.

Keywords: *Genetic Algorithm, Great deluge, Welded Beam Design, Pressure Vessel*

1. INTRODUCTION

Many real world problems in industrial engineering optimization are too complex such that it can be formulated as Nonlinear Programming (NLPs) problems. Moreover, no known fast algorithm is able to solve these problems since the objective functions are complicated and it has several constraint structures. Constrained optimization problems have been studied widely in the operations research and artificial intelligence literature. Operations research formulations deals with constraints as quantitative formulas, as a result, the solvers optimize the objective functions values subject to constraints.

On the other hand, artificial intelligence deals with inference-based algorithms with symbolic constraints. There have been huge research efforts on developing and testing new algorithms in the last decade to solve constrained optimization problems in artificial intelligence field, such as genetic algorithm, tabu search, simulating annealing . . . etc. Moreover, a hybridized algorithm is which inherits the advantages of two or more algorithms in one algorithm attract researchers to develop fast and robust algorithms. For instance, it is not clear there is an exact method which can solve nonlinear optimization problems. An exceptional survey about constrained optimization can be found in [3], [4], [7],[8].

In this work, we introduce a new algorithm was developed to provide a reasonable solution for complex optimization problems under constraints based Genetic Algorithms (GA). Compared to traditional search techniques, GAs can overcome many problems encountered by gradient based methods.

The aim of this algorithm is to provide a good algorithm which is able to solve nonlinear constrained optimization problems using Matlab. On doing this 1) Hybridization between GAs and great deluge is implemented; 2) An optimization toolbox using Matlab was developed.

The remainder of this paper is as follow; Section 2 presents a brief background about optimization algorithms; Section 3 provides a brief introduction about genetic algorithms; The great deluge method is illustrated in Section 5. The description of the test functions is presented in Section 6. Section 7 investigates the achieved results. Finally, the paper is concluded making comment on the effectiveness of the technique studied and potential future research approaches.

2. LITERATURE REVIEW

The use of optimization in engineering design process is increased every day as the computational capabilities of the computers are increased. Therefore the applications of numerical

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

optimization have been increased dramatically. Moreover, some of numerical optimization can run on smart phones.

Although there are huge number of fundamental backgrounds has been published about constrained optimization algorithms, there are some basic algorithms we would like to highlight it here. Since non-linear constrained optimization problems are an important class of problems with a wide range of fields such as engineering, operational research and bio-informatics.

Fletcher [2] describes the optimization process as "the science of determining the best solution". To achieve this description, a mix between science and heuristics, with different fields such as science, engineering and economics . . . etc was presented. Optimization methods can be classified into derivative and non-derivative methods, Table 1 illustrates some of the basic methods of both categories. The main difference between both categories, derivative methods easy to stuck in local optima, while non-derivative methods are likely to find a global optima. Moreover, non-derivative methods do not require any derivative of the objective functions in order to calculate the optimum. Therefore, these algorithms are known as black box algorithms. Also, there are other promising algorithms, which help to find optimal design for engineering problems, such as neural networks, Taguchi methods and response surface approximations.

In general, non-derivative methods are more suitable for general engineering design problems since it's hard to express your design problem in terms of design variables directly. However, a quite enough of engineering tools are available today to estimate the performance of the design in an early stage of the design process. For example, CAD package, FEM software, CFD solvers and Matlab.

Table 1: Optimization Methods For Classification

Derivative Methods	Non-derivative Methods
Sequential Quadratic Programming	Simulating Annealing
Direct Search	Tabu Search
Box's Complex Method	Random Search
Gradient-based method	Genetic Algorithms

3. GENETIC ALGORITHM

Since 1970, Genetic algorithms attract researchers to use it for solving hard and complex problems. Genetic algorithm idea is adopted from darwinian evolution theory Holland [6]. Genetic algorithms are a population based mechanism, which is used to explore the search space better than other searching algorithms especially for problems that are NP-hard. Genetic algorithms dose not grantee an optimal solution; however it usually gives good enough approximations in reasonable amount of time. Moreover; genetic algorithms requires no gradient information, evolves from one population to another and produces multiple optima rather than single local one. These features make genetic algorithms a well-suited algorithm for solving nonlinear constrained optimization problems.

Genetic algorithm starts by creating a set of solutions (population) of individuals. These solutions is generated either randomly or based on a construction method based on the problem nature. The fitness of each solution is evaluated based on a fitness function. Genetic algorithms consist of five main operations which are: Encoding, Evaluation, Crossover, Mutation and Decoding. These operations enable genetic algorithms to be one of the most powerful algorithm to solve hard and complex optimization problems in finding the optimal solutions Holland [6]. A simple genetic algorithm is shown in Figure 1.

Based on genetic algorithm concept, a solution vector $x \in X$ is called chromosome which represent a unique solution in the search space; each chromosome is created from discrete units called genes. Different approaches have been implemented to present a problem solution as chromosome. As a result a mapping method is required which is called encoding. Genetic algorithm deals on the encoding of a problem, not on the problem itself.

Genetic Algorithms works on a collection of chromosomes (solutions) called population. The construction of these chromosomes is depends on the nature of the problem. In general, genetic algorithms works on a population pool of solutions in order to generate new solutions. The mechanism of generating new solutions is based on two operators: crossover and mutation. The crossover works on two solutions, called parents, are combined together to produce new two solutions, called offspring. Crossover operation leads the population to converge by making the chromosomes in the population similar. The selection of parents is based on a selection mechanism based on the fitness value of

Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

chromosomes in the population pool. High quality parents usually produce a high quality offsprings. The mutation process usually applied at the gene level. The mutation rate is usually small and depends on the length of the chromosome. Mutation reintroduces genetic diversity back into the population and assists the search escape from local optima.

```
Procedure Genetic algorithm
Begin
    Initialize solution population
    Evaluate solution population
While (termination condition not met)
    Update generation counter
    Select generation counter
    Select parents for the next generation
    Reproduce or recombine parent to form
offspring
    Mutate offsprings
    Evaluate offsprings
End while
End
```

Figure 1: A Simple Genetic Algorithm

4. GREATE DELUGE ALGORITHM

The great deluge algorithm was introduced by Dueck (1993). It is a local search procedure which has certain similarities with simulated annealing. This approach is far less dependent upon parameters than simulated annealing. It needs just two parameters: the amount of computational time that the user wishes to "spend" and an estimate of the quality of solution that a user requires. Apart from accepting a move that improves the solution quality, the great deluge algorithm also accepts a worse solution if the quality of the solution is less (for the case or minimisation) than or equal to some given upper boundary value B (in the paper by Dueck it was called a "level"). In this work, the "level" is initially set to be the objective function value of the initial solution. During its run, the "level" is iteratively lowered by a constant β where β is a force decay rate (see Figure 3). The great deluge algorithm will be applied on each timetable to reduce the total penalty cost based on the calculated force value.
The pseudo code for the great deluge is presented in Figure 2.

```
Set initial solution as Sol
                              bes
Calculate the initial cost function value,
f(Sol);
Set best solution, Sol    ← Sol;
                    best
Set estimated quality of final solution,
Set number of iterations, NumOfIte;
Set initial level: level ← f(Sol);
Set decreasing rate
    β = ((f(Sol)-estimatedquality)/(NumOfIte);
Set iteration ← 0;
Set not_improving_counter ← 0;
do while (iteration < NumOfIte)
     generate a new solution called Sol*;
     Calculate f(Sol*);
     if (f(Sol*) < f(Sol   ))
                        best
        Sol ← Sol*;
        Sol    ← Sol*;
           best
        not_improving_counter ← 0;
      else
         if (f(Sol*)≤ level)
           Sol ← Sol*;
           not_improving_counter ← 0;
         else
          Increase not_improving_counter by 1;
          if (not_improving_counter ==
             not_improving_ length_GDA)
          exit;
level = level - β;
Increase iteration by 1;
end do;
```

Figure 2: The Pseudo Code For The Great Deluge Algorithm

5. PROBLEM DESCRIPTION

A. Case Study I: Design of a Pressure Vessel

The first case study is a cylindrical vessel design Kannan and Kramer [9], which is capped at both ends by hemispherical heads as shown in Figure 9. The objective function is to minimize the total cost of fabrications such as welding, material and forming. Four variables have to be optimized as following:

T_S: Which is the thickness of the shell.
T_h: which is the thickness of the head.
R: Which is the inner radius.
L: Which is the length of the cylindrical section of the vessel, not including the head.

The variables R and L are treated as continuous variables, whilst TS and Th as discrete variables which are multiples of 0.0625 inch. The problem formulation $(T_S, T_h, R, L) = (x_1, x_2, x_3, x_4)$ can be stated as follows:

Minimize
$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3 + 3.1661x12x_4 + 19.84x12x_3$
Subject to:
$g_1(X) = -x_1 + 0.0193x_3 \leq 0$
$g_2(X) = -x_2 + 0{:}00954x_3 \leq 0$
$g_3(X) = -\pi x_3 2\, x_4 - 4/3\pi x33 + 1{,}296{,}000 \leq 0$
$g_4(X) = x_4 - 240 \leq 0$

Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

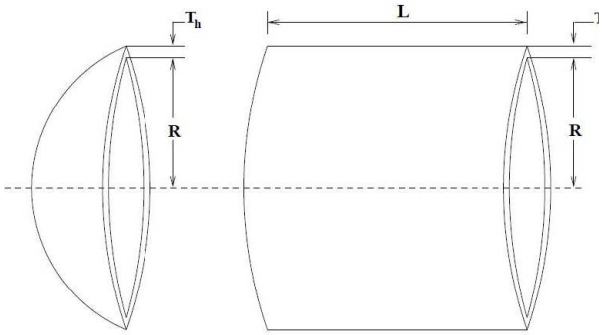

Figure 3: Diagram Of The Pressure Vessel Used As Case Study

B. Case Study II: Welded Beam Design

The second case study is welded beam design problem. The objective is to minimize the cost subject to a set of constraints on share strss (τ), bending stress in the beam (σ), buckling load on the bar (Pc), end reflection of the bean(δ), and side constraints.The problem consists of four design variables (h(x1), l(x_2), t(x_3), b(x_4))=(x_1, x_2, x_3, x_4).
Minimize f(X) = 1.10471x_1 2x_2 + 0.04811x_3x_4(14.0x_2)

Subject to:

$$g_1(X) = \tau(X) - \tau_{max} \leq 0$$
$$g_2(X) = \sigma(X) - \sigma_{max} \leq 0$$
$$g_3(X) = x_1 - x_4 \leq 0$$
$$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 0.5 \leq 0$$
$$g_5(X) = 0.125 - x_1 \leq 0$$
$$g_6(X) = \delta(X) - \delta_{max} \leq 0$$
$$g_7(X) = P - P_c \leq 0$$

where:

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$$

$$J = 2\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2\right]\}$$

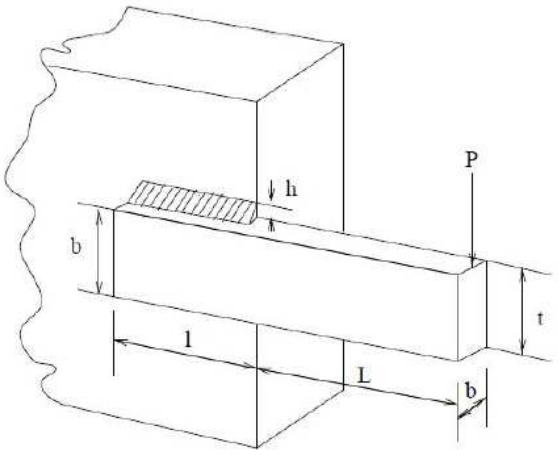

Figure 4: Diagram Of Welded Beam Design

6. SIMULATION RESULTS

The proposed algorithm was programmed using Matlab and simulations were performed on an Intel Pentium 4 2.33 GHz computer. The algorithm was tested on a standard optimization functions with 11 runs for each function.

The parameters used in the algorithm are chosen after preliminary experiments. Table 2 shows the parameters settings used in our experiments. Tables 3 and 4 provide a summary of the results obtained by our hybrid algorithm.

Table 2: Parameter Settings

Parameters	Value
Genetic Algorithm iteration	10000
Great Deluge iteration	1000
Population Size	50
Crossover Rate	0.74%
Mutation Rate	0.05 %

Table 3: Pressure Vessel Results

Run	x_1	x_2	x_3	x_4	f(x)
1	0.20540	0.347391	9.03662	0.20599	1.72486
2	0.20570	0.347011	9.03662	0.20544	1.72485
3	0.25801	0.347048	9.03659	0.20569	1.72485
4	0.20580	0.347046	9.03658	0.20571	1.72485
5	0.20561	0.34704	9.0367	0.2057	1.72486
6	0.20551	0.347039	9.03661	0.20571	1.72485
7	0.20532	0.347011	9.0366	0.2057	1.7248
8	0.20546	0.347100	9.03660	0.20571	1.7248
9	0.20481	0.34201	9.03669	0.20576	1.7248
10	0.20601	0.347022	9.03668	0.20573	1.7248
11	0.20581	0.34709	9.03661	0.20576	1.7248

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Welded Beam Design Results

Run	x_1	x_2	x_3	x_4	f(x)
1	0.58209	0.347391	0.20140	0.447043	1.8374
2	0.58604	0.347011	0.20250	0.447017	1.8335
3	0.55205	0.347048	0.24530	0.467035	1.8345
4	0.54205	0.347046	0.23352	0.457039	1.8386
5	0.58204	0.34704	0.25540	0.447069	1.8337
6	0.58209	0.347039	0.25470	0.437034	1.8358
7	0.55205	0.347011	0.27540	0.437035	1.8359
8	0.54279	0.347100	0.29950	0.427037	1.8384
9	0.52409	0.34201	0.21750	0.427033	1.8333
10	0.55509	0.347022	0.20320	0.457049	1.8324
11	0.56509	0.34709	0.20590	0.457037	1.8358

Our method is able to produce good enough results. The objective of these results is to give an indication of the variability between runs of the proposed algorithm. It is believed the quality of the solutions obtained in these experiments can be attributed to the ability of the algorithm in effective exploration of different regions of the solution space, applied to 50 different solutions, for each iteration. Our approach is able to further improve resultant solutions. However, the longer the search times, the slower the rate of improvement.

7. CONCLUSION

This paper has described the hybridization between genetic algorithms and great deluge algorithm. A set of problems have been solved using this method. The algorithm attempts to exploit the inherent advantages from genetic algorithms and great deluge algorithm. The proposed algorithm provides a balance between exploration and exploitation within the search strategy. Moreover, the results of the proposed algorithm outperform several algorithms.

REFRENCES:

[1] Vanderplaats GN. Multidiscipline Design Optimization. Colorado Springs, Vanderplaats Research and Development, Inc, 2007.

[2] R. Fletcher. Practical Methods of Optimization, A Wiley-Interscience Publications, John Wiley & Sons, Chichester, , 1987.

[3] Sandgren E. Nonlinear integer discrete programming in mechanical design. In: Proc ASME Design Technology Conference, Kissimmee, FL,USA, 1988.

[4] Deb. Kalyanmoy. GeneAS: a robust optimal design technique for mechanical component design. In: Proc Evolutionary Algorithms in Engineering Applications. Springer, Berlin Heidelberg New York 628 Int J Adv Manuf Technol (2009) 40:617-628.

[5] K.M. Ragsdelland D.T. Phillips Optimal Design of a Class of Welded Structures Using Geometric Programming. ASME Journal of Engineering for Industries, 98(3):1021-1025, Series B, 1976.

[6] Holland J. Genetic Algorithms, Scientific American, 66-72, 1992.

[7] Yeniay O. A Comparative study on optimization methods for the constrained nonlinear programming problems, Mathematical Problems in Engineering, 2 ,165-173, 2005.

[8] CAC. Coello Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41: 113-127, 2004.

[9] BK. Kannan and SN. Kramer.An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. ASME J Mech Des 116:318-320, 1994.