Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

A New Genetic algorithm for Nonlinear Programmig Problems

Jiafu Tang and Dingwei Wang
P.O.135, Scool of Information Science & Engineering,Northeastern University
Shenyang, Liaoning, 110006, P.R.China. email: neu423@mail.neu.edu.cn

Abstract

A special Genetic Algorithm with mutation along the weighted gradient direction for nonlinear programming problems is proposed. It uses penalty function to construct fitness function for evaluating the solution which violated the constraints. The convergency analysis of the method are also given in this paper.

1 Genetic Algorithm with mutation along the weighted gradient direction for NLP

1.1 Canonical form of NLP

Nonlinear programming problems with n variables and m constraints may be written as the following canonical form NLP:

$$\begin{cases} Max\ f(x) = f(x_1, x_2, \dots, x_n) \\ S.t\ \ x \in Q = \left\{ x \in E_n \mid g_i(x) \le 0,\ i = 1, 2, \dots, m \right\} \end{cases} \tag{1}$$

In this paper, we assume that $f(x)$ and $g(x)$ are continuous and derivative in E_n.

1.2 Genetic Algorithm combined with penalty function

In the procedure of solution to NLP by means of penalty function method, it first transmit NLP into a unconstrained optimization problem by means of penalty function, then solve a sieries of unconstrained optimization problems with a certain penalty multiplier, in order to get the optimal solution or near optimal solution to original problem. But as the penalty multiplier tend to zero or infinite, the iteration point tend next to optimal, however, at the same time the objective function of the unconstrained optimization problem might gradually become worse. This is the computer difficulty of implementing the penalty function methods to solve NLP.

On the other hand, when we applying traditional GA to solve NLP, a scheme of coding and decoding process for optimizing variable must be needed, more over, due to the complexity and differences of constraints in actual optimization problems, there has no a general coding method for all types of optimization problems.

Based on the above analysis, in this paper, we propose a special genetic algorithm combined with penalty function to solve NLP.The basic idea may be described as follow: First randomly produce an initial population with the size of NP individuals, each individual is selected to reproduce children by mutation along the weighted gradient direction, according to the selection probability depending on its fitness function(objective function).In the process of iteration, for an individual $x_i \notin Q$, give it a less fitness function, namely, objective function, by way of penalty function, so that it may have a less chance than others to be selected as parents to reproduce children in the later generation.As the generation increases, the individual with less fitness function die out gradually, namely, the individuals $x_i \in Q$ with less objective function and individuals $x_i \notin Q$ die out gradually, and the individuals maintained in the population are the individuals with a high value of objective function. After a number of generations, the individuals' objective function value reach at the optimal or near optimal.

For an individual x, if $x \in Q$, then move along the gradient direction of objective function $\nabla f(x)$, the objective function may be improved.

For an individual x, if $x \notin Q$, it denotes that x is out of the feasible domain.Let

$$I^+ = \left\{ i \mid g_i(x) > 0, x \in E_n \right\}.$$

For $i \in I^+$,if x move along the negative gradient direction $\nabla g_i(x)$, it may satisfy $g_i(x) \le 0$.

Based on this, we construct a weighted gradient direction[1], denoted by $d(x)$, which is defined as follow:

$$d(x) = \nabla f(x) - \sum_{i=1}^{m} w_i \nabla g_i(x) \tag{2}$$

where, w_i is the weight of gradient direction.

$$w_i = \begin{cases} 0 & g_i(x) \le 0 \\ \delta_i & g_i(x) > 0 \end{cases} \tag{3}$$

0-7803-3970-8/97 $10.00 © 1997 IEEE

$$\delta_i = \frac{1}{g_{max}(x) - g_i(x) + \delta} \quad (4)$$

$$g_{max}(x) = max\left\{g_i(x), i = 1, 2, \ldots, m\right\} \quad (5)$$

where δ is a very small positive number.

Then $x_i^{(k+1)}$ is generated from $x_j^{(k)}$ by mutation along the weighted gradient direction $d(x)$ can be described as:

$$x_i^{(k+1)} = x_i^{(k)} + \beta^{(k)} d(x_i^{(k)}) \quad (6)$$

where, $\beta^{(k)}$ is a step-length of Erlang distribution random number with declining means, generated by random number generator.

From formular(5), we may find that $g_{max}(x)$ is the maximum value of violation for constraints, which reflects the information of relationship between individual x and the feasible domain Q.

If $g_{max}(x) \leq 0$, it denotes $x \in Q$; else, $g_{max}(x) > 0$ and the bigger the $g_{max}(x)$ is, the worse the performance of $x \in Q$, namely, individual x is 'far' apart from feasible domain.

Objective function $f(x)$ is calculated as

$$f(x) = \begin{cases} f(x) & g_{max}(x) \leq 0 \\ \frac{f(x)}{M + g_{max}(x)} & \text{else} \end{cases} \quad (7)$$

where, M is a preferred very big positive number.

2 Convergency Analysis of GA

In this section, we give the convergency of the proposed GA. In the GA with mutation along the weighted gradient direction, the key procedure is

$$x^{(k+1)} = x^{(k)} + \beta^{(k)} d(x^{(k)}).$$

let x^* is the optimal solution to NLP(1).

$$I = \left\{i \mid g_i(x^*) = 0, i = 1, 2, \ldots, m\right\}$$

then,

$$d(x^*) = \nabla f(x^*) \quad (8)$$

Case 1 x^* is in the feasible domain.

In this case, $I = \phi$(empty), according to the Kuhn-Tuck condition, we have $\nabla f(x^*) = 0$, in light of formuar(8), $d(x^*) = 0$.

Hence, in the process of iteration, if a iteration point rearch at the optimal point, it would be stop at that point. Then we have Theorem 1.

Theorem 1 The solution generated by the proposed Genetic Algorithm after enough generation can converge to the optimal solution,if it is in the feasible domain.

Corollary 1 If the solution x_0, generated by GA, satisfy

$$\begin{cases} g_i(x_0) < 0, & i = 1, 2, 3, \ldots, m \\ \nabla f(x_0) = 0 \end{cases}$$

then, x_0 is the optimal solution.

Case 2 x^* is at the edge of the feasible domain.

In this case,

$$I \neq \phi.$$

in light of Kuhn-Tuck condition, $\exists \lambda_i^* \geq 0$ such that

$$\nabla f(x^*) - \sum_{i \in I} \lambda_i^* \nabla g_i(x^*) = 0 \quad (9)$$

where, λ_i^* is Lagrange multiplier.

Obviously, $d(x^*) \neq 0$, namely, according to the proposed GA, in the process of iteration, a point reach at the optimal solution, but it might not be stop at the optimal solution.However, it can converge to the neighbore domain of optimal solution.

Based on the above analysis, we give some corrections to the gradient direction weight w_i as follow:

$$w_i = \begin{cases} 0 & g_i(x) < 0 \\ \lambda_i & g_i(x) = 0 \\ \delta_i & g_i(x) > 0 \end{cases} \quad (10)$$

$$\lambda_i = \begin{cases} \lambda_i^* & \text{if } \lambda_i^* \text{ exits} \\ \delta_i & \text{else} \end{cases} \quad (11)$$

where, λ_i^* is the Lagrange multiplier which satisfy formular (9). Then, we may get theorem 2 as follow:

Theorem 2 The solution generated by the corrected Genetic Algorithm can converge to the optimal solution if it is at the edge of the feasible domain.

3 Conclusion

In summary, a special Genetic Algorithm with mutation along the weighted gradient direction is proposed in this paper.

References

[1] Tang, J. and Wang, D., 1997, An interactive approach based on GA for a type of Quadratic Programming Problems with fuzzy objective and resources, *Computers & Operations Research*, 24(5),pp413-422.