AN ADAPTIVE CONJUGATE GRADIENT ALGORITHM WITH CLUSTERING THE SINGULAR VALUES OF THE SEARCH DIRECTION MATRIX FOR LARGE-SCALE UNCONSTRAINED OPTIMIZATION

Neculai ANDREI[1]

Abstract. *An adaptive nonlinear conjugate gradient algorithm based on clustering the singular values of the search direction matrix and on the inexact Wolfe line search is presented. The search direction is dependent to a positive parameter. The value of this parameter is selected in such a way that the singular values of the matrix defining the search direction are clustered around 1. We prove that for general nonlinear functions and independent of the line search procedure the search direction satisfies both the sufficient descent condition and the Dai and Liao conjugacy condition. According to the value of the parameter, the algorithm uses the suggested search direction, or it triggers to the Hestenes and Stiefel direction. Under classical assumptions, for uniformly convex functions, we prove that the algorithm is globally convergent. Using a set of 800 unconstrained optimization test problems we prove that our algorithm is significantly more efficient and more robust than* CG-DESCENT *algorithm and slightly more efficient and more robust than* ADCG *algorithm. By solving five applications from the* MINPACK-*2 test problem collection, with 10^6 variables, we show that the suggested adaptive conjugate gradient algorithm is top performer versus* CG_DESCENT.

Keywords: Unconstrained optimization, Conjugate gradient algorithms, Wolfe conditions, Singular values clustering, Sufficient descent condition

1. Introduction

Let us consider the unconstrained optimization problem

$$\min\{f(x), x \in R^n\}, \tag{1.1}$$

where $f : R^n \to R$ is continuously differentiable and bounded below. For solving this problem we suggest a nonlinear conjugate gradient algorithm, where the iterates x_k, $k = 0,1,\ldots$ are generated as

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

the stepsize α_k is positive and the search directions d_k are computed as:

$$d_{k+1} = -g_{k+1} + \beta_k^N s_k, \;\; d_0 = -g_0, \tag{1.3}$$

[1]Research Institute for Informatics, Center for Advanced Modeling and Optimization, 8-10 Averescu Avenue, Bucharest 1, Romania (nandrei@ici.ro).