PERGAMON

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Structures 81 (2003) 197–213

Computers & Structures

www.elsevier.com/locate/compstruc

Move limits definition in structural optimization with sequential linear programming. Part I: Optimization algorithm

Luciano Lamberti *, Carmine Pappalettere [1]

Dipartimento di Ingegneria Meccanica e Gestionale, Politecnico di Bari, Viale Japigia 182, 70126 Bari, Italy

Received 5 November 2001; accepted 1 November 2002

Abstract

A variety of numerical methods have been proposed in literature in purpose to deal with the complexity and non-linearity of structural optimization problems. In practical design, sequential linear programming (SLP) is very popular because of its inherent simplicity and because linear solvers (e.g. Simplex) are easily available. However, SLP performance is sensitive to the definition of proper move limits for the design variables which task itself often involves considerable heuristics. This research presents a new SLP algorithm (LESLP—linearization error sequential linear programming) that implements an advanced technique for defining the move limits. The LESLP algorithm is formulated so to overcome the traditional limitations of the SLP method. The new algorithm is successfully tested in weight minimization problems of truss structures with up to hundreds of design variables and thousands of constraints: sizing and configuration problems are considered. Optimization problems of non-truss structures are also presented.

The key-ideas of LESLP and the discussion on numerical efficiency of the new algorithm are presented in a two-part paper. The first part concerns the basics of the LESLP formulation and provides potential users with a guide to programming LESLP on computers. In a companion paper, the numerical efficiency, advantages and drawbacks of LESLP are discussed and compared to those of other SLP algorithms recently published or implemented in commercial software packages.

Keywords: SLP; Move limits; Linearization error; Trust region

1. Introduction

An optimization problem is defined by the objective function and inequality constraint functions and it is formulated as follows:

$$\begin{cases} \mathrm{Min}\, W(x_1, x_2, \ldots, x_N) \\ G_k(x_1, x_2, \ldots, x_N) \geqslant\ \leqslant 0 \\ x_j^{\mathrm{l}} \leqslant x_j \leqslant x_j^{\mathrm{u}} \end{cases} \quad \begin{cases} j = 1, \ldots, N \\ k = 1, \ldots, NC \end{cases} \tag{1}$$

where $(x_1, x_2, \ldots, x_N)$ are the N design variables; $W(x_1, x_2, \ldots, x_N)$ is the objective function; $G_k(x_1, x_2, \ldots, x_N)$ are the NC inequality constraint functions; x_j^{l} and x_j^{u} are the lower and upper bounds of the jth design variable.

Expressions (1) show that optimization problems usually have very complex and highly non-linear implicit formulations. Therefore, the sequential linear programming (SLP) method is a popular approach to deal with the complexity and non-linearity of optimization problems. Here, the original non-linear problem is replaced with a linearized problem, built using linear approximations of the objective function and constraints. Each non-linear function of the problem is linearized about a point P_o^i and replaced with its first-order Taylor series

* Corresponding author.

E-mail addresses: lamberti@dimeg.poliba.it (L. Lamberti), carpa@poliba.it (C. Pappalettere).

[1] President of the Italian Society of Stress Analysis (AIAS).

0045-7949/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
PII: S0045-7949(02)00442-X

expansion. The original non-linear problem changes into the following linearized one:

$$\begin{cases} \text{Min}\, W(\mathbf{X}_o^i) + \sum_{j=1}^{N} \frac{\partial W}{\partial x_j}\Big|_{P_o^i}(x_j - x_{o,j}^i) & \\ G_k(\mathbf{X}_o^i) + \sum_{j=1}^{N} \frac{\partial G_k}{\partial x_j}\Big|_{P_o^i}(x_j - x_{o,j}^i) \geqslant\, \leqslant 0 & \begin{cases} j = 1,\ldots,N \\ k = 1,\ldots,NC \end{cases} \\ x_j^{\mathrm{l}} \leqslant x_j \leqslant x_j^{\mathrm{u}} & \end{cases} \tag{2}$$

The $\mathbf{X}_o^i(x_{0,1}^i, x_{0,2}^i, \ldots, x_{0,N}^i)$ vector defines the linearization point P_o^i in the design space while the superscript i refers to the current optimization cycle (ith iteration).

If the optimization problem is convex the linearized constraints lie entirely outside the feasible region; hence the optimum found with the linearized sub-problem will be infeasible. However, the convergence to the solution of the original problem can be achieved after a few re-linearizations. The SLP method is hence a recursive procedure consisting of the formulation and resolution of a series of linearly approximated sub-problems, where each intermediate solution is chosen as the starting point of the subsequent sub-problem.

In practical optimization, the SLP method is very popular because of its simple theoretical foundation and because reliable linear solvers are readily available (i.e. Simplex packages) [1]. Besides, the method performs very well in convex programming problems with nearly linear objective and constraint inequality functions [2]. However, because of their inherent conceptual simplicity, the SLP techniques are not globally convergent [3]. Moreover, numerical problems like convergence to local or infeasible optima and objective function oscillations may occur if the method is not well controlled [1]. Finally, for under-constrained problems, where there are fewer active constraints than there are design variables, the method often performs poorly because the linearized domain is unbounded [4].

In view of these limitations and of its simplicity, SLP is often considered a “poor” method by many theoreticians. For instance, Arora objected that the SLP method should not be used as a black box for solving optimization problems because there is no descent function; moreover, there are considerable heuristics involved in the formulation of the approximate sub-problems [5]. However, it is generally acknowledged that “well coded” SLP algorithms may outperform very sophisticated optimization methods. An SLP algorithm is said to be well coded if it includes tools for enhancing the numerical efficiency. In particular, the use of proper move limits is seen as the best way to make the SLP method reliable and efficient. The move limits are additional side constraints that define a region of the design space where the solution of the linearized sub-problem will lie. This region is located in the neighbourhood of the linearization point and is called the move limit domain. The move limit domain is built in purpose to have a reasonable accuracy of the linear approximation also when design variables are considerably perturbed within the optimization process. Care should be taken in the definition of move limits: they must not be too large in order to avoid oscillations in numerical solution, and not too small in order to have a reasonable convergence rate without ending trapped into local optima or infeasible designs. In addition, move limit definition should include line search procedures in order to improve the cost function in each iteration.

Since Pope introduced move limits in the early ’70s, researchers proposed a variety of techniques to define the move limits. Haftka and Gurdal [1] suggests to choose as move limit the 10–30% of the value that a design variable has at the beginning of the iteration cycle. The move limits are gradually shrunk as the design approaches the optimum and they should be reduced if the cost function does not improve or the constraints are more violated than at the previous iteration. Vanderplaats and Kodyalam [6] adopt the following criterion: the move limits are very large in the first design cycles and they are then reduced by 50% if the design violates the constraints more than at the previous iteration; anyhow the move limits are never reduced to less than 25%. John et al. [7] solve the linearized sub-problems with the Simplex algorithm. Move limits are reduced by means of the α parameter each time the design does not improve in the current iterate. The optimal step size α is determined with reanalysis techniques based on quadratic approximations of the cost function.

Schittowsky et al. [8] implemented an SLP algorithm which adjusts move limits according to trust region models. The technique uses a penalty function strategy to build the descent function. Numerical tests proved that the algorithm is suitable also for large-scale optimization problems. However, the very large number of parameters (σ, ρ_1, ρ_2) required in input is a considerable drawback.

Yu Chen [9,10] defines the move limits using the gradients of constraint equations. The move limits are evaluated each iteration or only in the first design cycle and then shrunk by means of an user supplied factor. The basic idea is to approach the constraint boundaries very quickly. Intermediate designs are accepted or rejected according to improvements in cost function or reduction of constraint violation found with inexact line search based on reanalysis techniques. Yu Chen’s techniques are very interesting because the definition of move limits does not involve any heuristic criteria and hence the user gets rid of guesswork. However, particularly in the case of optimization problems with many design variables and constraints, control parameters are needed to preserve the numerical efficiency (CPU time