Computers & Fluids 46 (2011) 298–305

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Multi-objective shape optimization using ant colony coupled computational fluid dynamics solver

G.N. Sashi Kumar *, A.K. Mahendra, G. Gouthaman

Machine Dynamics Division, Bhabha Atomic Research Centre, Trombay, Mumbai, India

ARTICLE INFO

Article history:
Received 30 April 2010
Received in revised form 22 December 2010
Accepted 12 January 2011
Available online 18 January 2011

Keywords:
Multi-objective optimization
Ant colony optimization
MACO
Shape optimization
CFD
Elitism
Parametric optimization
R-NSGA-II

ABSTRACT

An adaptation of a parametric ant colony optimization (ACO) to multi-objective optimization (MOO) is presented in this paper. In this algorithm (here onwards called MACO) the concept of MOO is achieved using the reference point (or goal vector) optimization strategy by applying scalarization. This method translates the multi-objective optimization problem to a single objective optimization problem. The ranking is done using ϵ-dominance with modified L_p metric strategy. The minimization of the maximum distance from the goal vector drives the solution close to the goal vector. A few validation test cases with multi-objectives have been demonstrated. MACO was found to out perform R-NSGA-II for the test cases considered. This algorithm was then integrated with a meshless computational fluid dynamics (CFD) solver to perform aerodynamic shape optimization of an airfoil. The algorithm was successful in reaching the optimum solutions near to the goal vector on one hand. On the other hand the algorithm converged to an optimum outside the boundary specified by the user for the control variables. These make MACO a good contender for multi-objective shape optimization problems.

1. Introduction

There are a few global optimization algorithms that are based on natural processes. Genetic algorithms (GA) are based on Darwin's theory of natural genetics on one hand. On other hand ant colony algorithms are based on insect behavior usually called swarm algorithms. Simulated annealing is based on physical process. The swarm algorithms are based on collective intelligence, defined as the ability of a group to solve problems more efficiently than its individuals [5]. GA and its variants [7,14,25] have been extensively used for multi-objective optimization (MOO) problems in last few decades.

Ant colony optimization (ACO) is a meta-heuristic based global optimization technique introduced by Dorigo [8,9] and has proved itself in field of combinatorial optimization problems. Many variants of Ant colony algorithms have been reported in recent few years for combinatorial multi-objective optimization problems [19]. One of the approaches uses multiple ant colonies with exchange of information between them [17]. Cardoso et al. [5] have used a single ant colony with cost vector which was associated to multi-level pheromone trails to solve a multi-objective network ant colony optimization (MONACO) problem. Another variant, the Max-Min ant system [19] was applied in a production process to minimize lead time as well as work required. Garcia-Martinez et al. [11] proposed a taxonomy for ACO algorithms along with an empirical analysis for bi-criteria TSP problem. The automatically configured algorithm by López-Ibáñez and Stützle [12] is reported to outperform the MOACO. Alaya et al. [2] proposed a generic algorithm based on ACO to solve MOO problems. Angus [4] extends the ACO algorithm with a crowding population replacement scheme to increase the search efficiency.

Abbaspour et al. [1] has extended ACO to parametric optimization using the route of inverse modeling and named it as ACO-IM. The authors have earlier demonstrated shape optimization capability of ACO-IM when coupled with computational fluid dynamics (CFD) solver using a single objective function [20,22,23]. Authors earlier work on aerodynamic shape optimization using GA [21] using a single objective shows that ACO-IM out performs GA in certain situations [20]. The advantage of ACO-IM method is that, it facilitates the movement of domain space of the variables. Hence the initial domain for the variables defined by the user does not need to contain the final optima [22]. Fainekos and Giannakoglou [10] have demonstrated airfoil optimization using extended ACO (EACO). This algorithm optimizes the path between pairs of control variables using a local criterion over and above the global pheromone based criterion. It is not always possible to define a local criterion for an optimization problem. The movement of domain space of variables is restricted in EACO. Also the best accuracy of final solution is bounded by the user-defined limits and

* Corresponding author.
E-mail address: gnsk@barc.gov.in (G.N. Sashi Kumar).

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.01.016

discretization done for each control variables. Discretization is performed once at the beginning and remains same throughout the optimization process. A control variable cannot take values between these discretized values. In contrast, in ACO-IM the discretization is done in all iterations with new bounds for the variables. This process improves the accuracy of the solution as optimization progresses. Definitely for a general optimization problem (like the problems dealt in this paper) ACO-IM is superior to EACO. In this paper the concept of multi-objective optimization using ACO-IM (MACO) is proposed. It is then implemented on a problem of aerodynamic shape optimization with multiple objectives.

Next section discusses the concept of MOO in reference to ACO-IM. Section 3 demonstrates the test Cases 1 and 2. Section 4 elaborates the integration of meshless CFD solver with MACO along with results and discussion. Section 5 details conclusion along with future scope.

2. MOO for ACO-IM

ACO-IM algorithm works with single objective function and in the process of convergence shrinks the control space to a single small vector space. To extend such a method to MOO using reference point based methodology is the most appropriate. The concept of scalarization in reference point strategy reduces the multi-dimensional design space to single dimension [24]- fit for ACO-IM.

In many engineering problems the designer has an idea of the preferred zone of operation. In such situations one need not require the details of complete Pareto front but needs to know the best optimum (lying on the Pareto front) in the preferred zone. Also in reference point based methodology, interest is in obtaining a solution near to the goal vector and ACO-IM shrinks the control space to a region that is close to the goal vector. ACO-IM is unlike GA where a cluster of solutions is available at each generation. So, in order to extend ACO-IM to MACO the basic ACO-IM requires no changes, yet improvement on the values of N and T were made to have a more reliable algorithm. The major initiatives have been taken to select appropriate scalarization function and ranking procedure so as to obtain the global optimum in minimum number of function evaluations (FE).

MOO problem can be described as follows. Let $f(\vec{x}) : D \Rightarrow \mathbb{R}$ be a continuous and bounded function where D is the search space of $\vec{x}$. Find the vector of decision variables $\vec{x}^* = [x_1^*\ x_2^* \ldots x_n^*]^T;\ \vec{x}^* \in X \subset \mathbb{R}^n$ such that the objective functions $f_i(\vec{x}) \in F \subset \mathbb{R}^m,\ i = 1, 2, \ldots, m$ are simultaneously minimized.

Based on the preferences of decision maker (DM) the MOO problem gets classified as (a) apriori method (b) a posteriori method (c) progressive method and if there are no preferences then (d) no preference method. The approach of scalarization using a goal vector implemented in this paper is an apriori preference method.

The method of scalarization used is similar to Mahendra et al. [14] simplifies the multiple objective function space to a single value. Scalarizing function $s : Q \times F \to \mathbb{R}^1$, where goal vector or reference point $\vec{q}(\vec{x}) \in Q \subset \mathbb{R}^m$. The MOO problem is then replaced with the following single objective optimization problem called the weighted modified L_p problem.

$$\underset{\vec{x}\in\mathbb{R}^n}{Minimize}\, L_p(\vec{f}, \vec{w}, \vec{q}^{\,\star})_{\vec{x}} = \left(\sum_{i=1}^{m} w_i \left(\frac{|f_i(\vec{x}) - q_i^{\star}|}{Max_i |f_i(\vec{x}) - q_i^{\star}|} \right)^p \right)^{1/p} \tag{1}$$

where $1 \leqslant p \leqslant \infty$, $w_i \geqslant 0\ \forall i \in 1, 2, \ldots, m$.

This paper uses Eq. (1) with $p = 2$ and $w_i = 1/m$, i.e. modified L_2 metric for scalarization. Represented as $\|.\|_2$ from here onwards in this paper. Such scalarization clusters values near to 1. The floating point representation by digital computers/compilers is best represented near to value 1 or 0. We have minimal loss of significant digits thus better accuracy is ensured. Eq. (1) is bounded within [0,1]. On the contrary the weighted Euclidian distance measure used by Deb et al. [7] does not have a fixed upper bound. It was observed for MACO the ranking based on fixed bounds was more efficient in comparison with the weighted Euclidian distance measure. As the optimization proceeds the solution approaches towards the goal vector, but Eq. (1) fails to track the progress in convergence. In order to evaluate the performance and progress in optimization, a new measure 'Minimum Normalized Distance' (MND) from the goal vector is defined. *MND* has no role to play in the optimization algorithm. It is just a measure for better post-processing. This is similar to the minimization problem defined by Osyczka [18]

$$MND = Min\left(\sum_{i=1}^{m} \left(\frac{f_i(\vec{x}) - q_i^{\star}}{q_i^{\star}} \right)^2 \right)^{\frac{1}{2}} \tag{2}$$

The method of determining ranking is based on either (i) non-dominated sorting (Pareto based) described by Eq. (3). Fig. 1 illustrates the ranking methodology.

$$\vec{f}^{\star} \prec \vec{f} : \iff \underset{i\in 1,2,\ldots m}{\forall} (f_i(\vec{x}^{\star}) \leqslant f_i(\vec{x})) \wedge \underset{j\in 1,2,\cdots m}{\exists} (f_j(\vec{x}^{\star}) < f_j(\vec{x})) \tag{3}$$

The candidate solutions on the curve 1 belong to the Pareto optimal (non-dominated) solutions. Similarly curves 2 and 3 represent less non-dominated solutions. For the purpose of ranking, the non-dominated sorting curves 1, 2 and 3 are constructed from the given data. Solutions on curve 1 are given highest priority; next preference is given to curve 2 then to curve 3. Scalarization step is performed to select best candidate design from series of promising designs lying on the Pareto front.

Or (ii) sorting based on ϵ-dominance [14,13] described by Eq. (4).

$$\begin{aligned} \vec{f}^{\star} \prec \vec{f} : \iff & \underset{i\in 1,2,\ldots m}{\forall} (f_i(\vec{x}^{\star}) \leqslant f_i(\vec{x}) - \epsilon_i) \wedge \underset{j\in 1,2,\cdots m}{\exists} (f_j(\vec{x}^{\star}) \\ & < f_j(\vec{x}) - \epsilon_j) \exists L_p(\vec{f}, \vec{w}, \vec{q})_{\vec{x}^{\star}} < L_p(\vec{f} - \vec{\epsilon}, \vec{w}, \vec{q})_{\vec{x}}; \\ & \epsilon_i = \zeta(f_i^{max} - f_i^{min}); \quad \zeta \in (0, 1) \end{aligned} \tag{4}$$

$\zeta$is a user-defined parameter which determines the ϵ-dominance zone. The bounds for ζ is specified in later in this paper. In Fig. 1 there are a few candidates in the less non-dominated Pareto fronts (marked as 2 and 3) which are closer to the goal vector than many candidate solutions in the non-dominated Pareto front (numbered as (1)). The selection of all such closest solutions belonging to various less/non-dominated Pareto fronts refers to ranking based on ϵ-dominance. The sorting using ϵ-dominance is shown in form of loops. Yet our interest is in finding the closest solution to the GOAL vector on the non-dominated Pareto front. The distance of a solution

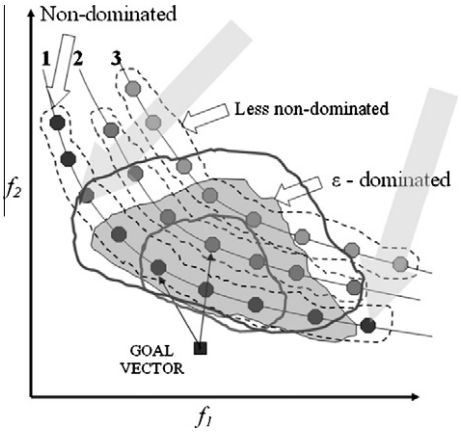

Fig. 1. Reference point based MOO.