Hybridizing an Evolutionary Algorithm with Mathematical Programming Techniques for Multi-Objective Optimization

Saúl Zapotecas-Martínez
CINVESTAV-IPN
Depto. de Computación
México, DF 07360, MEXICO
saul.zapotecas@gmail.com

Carlos A. Coello Coello*
CINVESTAV-IPN
Depto. de Computación
México, DF 07360, MEXICO
ccoello@cs.cinvestav.mx

ABSTRACT

In recent years, the development of multi-objective evolutionary algorithms (MOEAs) hybridized with mathematical programming techniques has significantly increased. However, most of these hybrid approaches are gradient-based, and tend to require a high number of extra objective function evaluations to estimate the gradient information required. The use of nonlinear optimization approaches taken from the mathematical programming literature has been, however, less popular (although such approaches have been used with single-objective evolutionary algorithms). This paper precisely focuses on the design of a hybrid between a well-known MOEA (the NSGA-II) and two direct search methods taken from the mathematical programming literature (Nelder and Mead's method and the golden section algorithm). The idea is to combine the explorative power of the evolutionary algorithm with the exploitative power of the direct search methods previously indicated (one is used for unidimensional functions and the other for multidimensional functions). Clearly, these mathematical programming techniques act as local search engines, whose goal is to refine the search performed by the MOEA. hybridization is quite promising.

General Terms

Hybrids, Algorithms.

Keywords

Multi-objective optimization, NSGA-II, hybrid algorithms, Nelder-Mead method.

1. INTRODUCTION

Despite the current widespread use of evolutionary algorithms for solving multi-objective optimization problems, their computational cost (measured in terms of the objective function evaluations that they require) remains as one of their main limitations when applied to real-world applications.

In order to address this issue, a variety of hybrid approaches have been proposed in the last few years, mainly focusing on the use of local search engines coupled to different types of MOEAs. Most of these hybrid approaches rely on local search engines based on gradient information. Few of these approaches, however, attempt to hybridize direct search methods (which do not require gradient information) with a MOEA.

In this paper, we present a new multi-objective hybrid algorithm based on the NSGA-II [1], coupled with two mathematical programming methods: Nelder and Mead's method [2] (which is used for multidimensional optimization) and the golden section (which is used for unidimensional optimization). Both approaches act as local search engines, whereas the NSGA-II evidently acts as the global search engine.

2. OUR PROPOSED APPROACH

Our proposed approach is called *Nonlinear Simplex Search Genetic Algorithm* (NSS-GA), and combines the explorative power of a MOEA with the exploitative power of the Nelder and Mead method (*nonlinear simplex search* (NSS)), which acts as a local search engine. Below, the general approach of the NSS-GA is described.

2.1 Local search

The general idea of this phase is to intensify the search towards better solutions for each objective function, based on an individual of the population.

The main goal of this phase is to obtain the λ set using classical optimization methods. Because the Nelder and Mead algorithm was designed to optimize multidimensional functions, when dealing with unidimensional optimizations, the golden section method is used instead. Thus, the λ set is defined as:

$$\lambda = \lambda_1 \cup \lambda_2 \cup \cdots \lambda_k \cup \Upsilon$$

where λ_i is a set of optimal solutions for the i-th objective function of the MOP and Υ is a set of optimal solutions for the aggregating function described in Section 2.1.2.

2.1.1 Selection Mechanism

In the population P, we choose the individual $x_\Delta \in P$ to optimize its i-th objective:

$$x_\Delta = x_l | x_l = \min_{\forall x_l \in \mathcal{P}^*} \{f_i(x_l)\} \quad (1)$$

*The second author acknowledges support from CONACyT project no. 45683-Y.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-131-6/08/07 ...$5.00.

where $\mathcal{P}^*$ is a set of nondominated solutions within the population P. In other words, the selected individual is the best nondominated solution with respect to the i-th objective function.

2.1.2 Aggregating Function

The vector $\vec{H} = [f_1^*, f_2^*, \ldots, f_k^*]$, consists of the minimum values f_i^* of the k objective functions in the current generation. We select individual x_a from the population P, such that we minimize:

$$G(x_a) = \sum_{i=1}^{k} \frac{|\vec{H}[i] - f_i(x_a)|}{|\vec{H}[i]|} \quad (2)$$

In this way, the local search minimizes the aggregating function defined by:

$$\psi(x) = ED(\vec{H}, \vec{F}(x)) \quad (3)$$

where $\vec{F}$ is vector of objective functions values of each individual and ED is the Euclidean distance between the $\vec{F}$ and $\vec{H}$ vectors

2.2 Considerations for the NSS Algorithm

There are functions for which the NSS algorithm becomes inoperable. In order to deal with these and other more complex functions, the NSS method has undergone some modifications. We propose here a new strategy to guide the improvement process towards promising areas during each generation of the hybrid algorithm. Such strategy is described next.

2.2.1 Building the Simplex

The selected solution x_Δ (x_a for the case of the aggregating function) is called "*simplex-head*", which is the first vertex of the n-simplex. The remaining n vertices are created in two phases:

1. *Reducing the Search Domain*: We use a strategy based on genetic analysis of a sample taken from the population. From this sample, we identify the average and standard deviation of the genes (decision variables) in each individual. Based on that information, we define the new search space as:

$$\begin{aligned} low_bound_{new}^j &= m(P_m(j)) - \sigma(P_m(j)) \\ up_bound_{new}^j &= m(P_m(j)) + \sigma(P_m(j)) \end{aligned} \quad (4)$$

   where P_m represents the individuals in the sample taken from the population (such individuals are those with the best fitness with respect to the objective function to optimize), $m(P_m(j))$ is the average and $\sigma(P_m(j))$ is the standard deviation in the j-th parameter of the sample P_m.

2. *Build Simplex Vertices*: The remaining vertices are determined using either the Halton or the Hammersley sequence (each has a 50% probability of being selected). For both sequences, the components are in $[0, 1]$ and are mapped to the new interval acording to:

$$c' = low_bound_{new} + c \cdot (up_bound_{new} - low_bound_{new})$$

   where c is the component to be mapped to the interval $[0, 1]$ and c' is the component already mapped to the desired interval.

2.2.2 Bounded Variables for the NSS Algorithm

The NSS method was conceived for unbounded domain problems. When dealing with bounded variables, the created vertices can be located outside the allowable bounds after some movements of the NSS method. We adopted a simple strategy to deal with bounded variables:

Let Δ_{new} be the new vertex created by some NSS movement. The j-th component of the vertex is established as:

$$\Delta_i^{(j)} = \begin{cases} low_bound_j & \text{, if } \Delta_i^{(j)} < low_bound_j \\ up_bound_j & \text{, if } \Delta_i^{(j)} > up_bound_j \\ \Delta_i^{(j)} & \text{, otherwise.} \end{cases} \quad (5)$$

where low_bound_j and up_bound_j are the lower and upper bounds in the j-th parameter, respectively.

2.2.3 Stopping Criteria

Two stopping criteria are adopted in this work. The first criterion imposes convergence towards a vertex better than the worst vertex within the simplex (x_w).

However, adopting this stopping criterion does not guarantee that the NSS method has an efficient performance. Convergence can be slow and may require a large number of evaluations of the objective function. For this reason, we use a second stopping criterion which consists of defining a convergence threshold ϵ. Thus, the local search is stopped if:

1. It does not generate a vertex better than x_w after performing $(n + 1)$ iterations, or

2. if after performing $2(n+1)$ iterations, the convergence is $\leq \epsilon$.

where n is the number of decision variables of the function to be optimized.

3. CONCLUSIONS

We have proposed a hybridization of the NSGA-II with the Nelder and Mead method, in which the former acts as a general search engine, and the latter works as a local search engine. For the local search, we use a nonlinear aggregating function. The proposed approach, called NSS-GA, was found to be competitive with respect to the original NSGA-II over a set of test functions taken from the specialized literature and results were assessed using the Inverted Generational Distance, Spacing and Coverage Indicator metrics, when performing only 4,000 fitness function evaluations.

We consider that the strategies adopted to reduce the search domain and to construct the initial simplex, significantly enhance the performance of the Nelder and Mead method, with respect to its traditional implementation.

4. REFERENCES

[1] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions Evolutionary Computation*, 6(2):182–197, 2002.

[2] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7:308–313, 1965.