Theoretical Computer Science 561 (2015) 24–36

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Population size matters: Rigorous runtime results for maximizing the hypervolume indicator

Anh Quang Nguyen [a,*], Andrew M. Sutton [b], Frank Neumann [a]

[a] Optimisation and Logistics, School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia
[b] Institut für Informatik, Friedrich-Schiller-Universität Jena, 07743 Jena, Germany

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form 26 May 2014
Accepted 15 June 2014
Available online 19 June 2014

Keywords:
Evolutionary multi-objective optimization
Hypervolume indicator
Genetic programming
Theory
Runtime time analysis

ABSTRACT

Evolutionary multi-objective optimization is one of the most successful areas in the field of evolutionary computation. Using the hypervolume indicator to guide the search of evolutionary multi-objective algorithms has become very popular in recent years. We contribute to the theoretical understanding of these algorithms by carrying out rigorous runtime analyses. We consider multi-objective variants of the problems OneMax and LeadingOnes called OneMinMax and LOTZ, respectively, and investigate hypervolume-based algorithms with population sizes that do not allow coverage of the entire Pareto front. Our results show that LOTZ is easier to optimize than OneMinMax for hypervolume-based evolutionary multi-objective algorithms, which is contrary to the results on their single-objective variants and the well-studied $(1+1)$ EA. Furthermore, we study multi-objective genetic programming using the hypervolume indicator. We show that the classical ORDER problem is easy to optimize if the population size is large enough to cover the whole Pareto front and point out situations where a small population size leads to an exponential optimization time.

1. Introduction

Many real-world applications involve optimization problems with multiple objectives and evolutionary algorithms have become very popular for handling problems in the area of multi-objective optimization [9,10]. Evolutionary algorithms seem to be well-suited for these types of problems since, in multi-objective optimization, the goal is to compute a set of solutions that represent the trade-offs with respect to the given objective functions. Evolutionary multi-objective algorithms aim to compute a set of such trade-offs by iteratively evolving a population of solutions into a reasonable collection that represents a good set of trade-offs with respect to the given objective functions.

Hypervolume-based evolutionary algorithms have become very popular in recent years for multi-objective optimization [1,4,12,22]. These algorithms work with a fixed population size of μ and try to maximize the volume of space that is covered by the μ objective vectors corresponding to each individual of the population.

Despite their popularity, it is difficult to understand from a theoretical point of view how hypervolume-based evolutionary algorithms work. Many studies in recent years have concentrated on the optimal hypervolume distribution for a wide range of problems, or the cost of computing or approximating the value of the hypervolume indicator for a given set of μ points [5,6]. These studies tackled the difficult task of determining the configuration of the optimal distribution of the μ

* Corresponding author. Tel.: +61 425 631 956.
E-mail address: a.nguyen@adelaide.edu.au (A.Q. Nguyen).

http://dx.doi.org/10.1016/j.tcs.2014.06.023

individuals without considering the computational costs to reach this goal. Other studies relate the optimal hypervolume distribution to the best achievable approximation ratio when using μ solutions to cover the Pareto front [3,7].

In this paper, we want to put forward the theoretical understanding of the optimization process of hypervolume-based evolutionary algorithms. We do this by carrying out rigorous runtime analyses of hypervolume-based evolutionary algorithms and point out, using popular example functions, when and why they are able to achieve an optimal hypervolume distribution in expected polynomial time. On the other hand, we show that even on very simple problems, hypervolume-based algorithms can have significant difficulties achieving the optimal configuration if the population size is not set correctly.

To our knowledge, the only article on the runtime analysis of hypervolume-based algorithms is the article by Brockhoff et al. [8]. We push forward the runtime analysis of hypervolume-based evolutionary algorithms by studying the algorithm called $(\mu + 1)$ SIBEA introduced in [8] in order to extend these initial investigations.

We start by analyzing the OneMinMax problem introduced in [14], which is the multi-objective version of the famous OneMax problem. We show that as long as μ is large enough to cover the entire Pareto front, $(\mu + 1)$ SIBEA computes the whole Pareto front in expected polynomial time. Furthermore, we point out that a smaller population size might lead to plateaus that are hard to leave in polynomial time. In particular, we show that there exists an initial population of size $\mu = O(\sqrt{n})$ such that the optimization time of $(\mu + 1)$ SIBEA is exponential with probability very close to 1.

After having investigated OneMinMax, we turn our attention to the multi-objective problem LOTZ introduced in [18]. Extending the investigations of [8], we show that $(\mu + 1)$ SIBEA optimizes LOTZ in expected polynomial time if $1 < \mu < n/3$. This shows that LOTZ is easier to optimize than OneMinMax by $(\mu + 1)$ SIBEA for small μ.

This paper extends a preliminary conference version [20] by additional investigations into the field of tree-based genetic programming. Genetic programming has been shown to be a successful approach to evolve syntax trees that solve a given task [17]. It is frequently used in the area of symbolic regression [13]. We analyze multi-objective genetic programming algorithms using the hypervolume indicator and consider the multi-objective formulation of the ORDER problem [11,15]. Multi-objective genetic programming algorithms for this problem have already by analyzed with respect to their expected optimization time in [19]. We show that hypervolume-based multi-objective genetic programming algorithms solve the ORDER problem efficiently if the population size is large enough. For smaller population size we point out situations leading to an exponential optimization time.

The outline of the paper is as follows. In Section 2, we introduce the framework that is subject to our investigations. Section 3, studies the behavior of $(\mu + 1)$ SIBEA on OneMinMax and these studies are extended to LOTZ in Section 4. Our results for multi-objective genetic programming using the hypervolume indicator are presented in Section 5. We finish with some concluding remarks.

2. Preliminaries

Let $\mathcal{X}$ be a finite domain. In multi-objective optimization, we work with a vector-valued fitness function $f : \mathcal{X} \to \mathbb{R}^m$ where $m \in \mathbb{N}$ and the fitness of an element $x \in \mathcal{X}$ is given by the vector $f(x) = (f_1(x), \dots, f_m(x))$. Assuming we want to maximize each function f_i, we say $f(x) \geq f(x')$ if and only if $f_i(x) \geq f_i(x')$ for $x, x' \in \mathcal{X}$ and all $i \in \{1, \dots, m\}$. We say a solution x dominates a solution x' if $f(x) \geq f(x')$ and $f(x) \neq f(x')$.

For a fitness function $f : \mathcal{X} \to \mathbb{R}^m$, we define

$$\mathscr{P} = \{x \in \mathcal{X} : f(x) \leq f(x') \Longrightarrow f(x) = f(x'),\ \forall x' \in \mathcal{X}\}$$

as the Pareto set of f. We sometimes call a point $x \in \mathscr{P}$ *Pareto optimal*.

The *hypervolume indicator* is a set measure that identifies a set of elements in $\mathbb{R}^m$ (corresponding to images of elements in $\mathcal{X}$) with the volume of the dominated portion of the objective space. In particular, given a reference point $r \in \mathbb{R}^m$, the hypervolume indicator is defined on a set $A \subset \mathcal{X}$ as

$$I_H(A) = \lambda\left(\bigcup_{a \in A} [r_1, f_1(a)] \times [r_2, f_2(a)] \times \cdots \times [r_m, f_m(a)]\right)$$

where $\lambda(S)$ denotes the Lebesgue measure of a set S and $[r_1, f_1(a)] \times [r_2, f_2(a)] \times \cdots \times [r_m, f_m(a)]$ is the orthotope with r and $f(a)$ in opposite corners.

We define the *extreme points* of a problem as

$$\{x \in \mathscr{P},\ \exists i \in \{1, \dots, m\} : f_i(x) \geq f_i(x'),\ \forall x' \in \mathcal{X}\}.$$

The extreme points are the points in $\mathscr{P}$ that maximize at least one objective function.

We study variants of $(\mu + 1)$ SIBEA algorithm introduced by Brockhoff et al. [8] outlined in Algorithm 1. The algorithm uses a population of size μ and produces in each generation an offspring by mutation to create an intermediate population of $\mu + 1$ individuals. The new parent population is then obtained in a steady-state fashion by deleting the individual that results in the smallest loss to the hypervolume indicator. This potential loss (called the contribution of an individual) is calculated as the difference between the indicator value of the population containing the individual, and the indicator value of a hypothetical population in which that individual is chosen for deletion.