A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume

Lyndon While

The University of Western Australia, WA 6009, Australia
lyndon@csse.uwa.edu.au

Abstract. We present a new analysis of the LebMeasure algorithm for calculating hypervolume. We prove that although it is polynomial in the number of points, LebMeasure is exponential in the number of objectives in the worst case, not polynomial as has been claimed previously. This result has important implications for anyone planning to use hypervolume, either as a metric to compare optimisation algorithms, or as part of a diversity mechanism in an evolutionary algorithm.

Keywords: Multi-objective optimisation, evolutionary computation, performance metrics, hypervolume.

1 Introduction

Multi-objective optimisation problems abound, and many evolutionary algorithms have been proposed to derive good solutions for such problems, for example [4, 11, 1, 5]. However, the question of what metrics to use in comparing the performance of these algorithms remains difficult[7, 9, 4]. One metric that has been favoured by many people is *hypervolume*[4, 8], also known as the S-metric[10] or the Lebesgue measure[3]. The hypervolume of a set of solutions measures the size of the portion of objective space that is dominated by those solutions collectively. Generally, hypervolume is favoured because it captures in a single scalar both the closeness of the solutions to the optimal set and, to some extent, the spread of the solutions across objective space. It also has nicer mathematical properties than many other metrics[2, 12], although it can be sensitive to scaling and to the presence or absence of extremal values.

Unfortunately, hypervolume is very expensive to calculate for problems with more than 2–3 objectives. A recently published algorithm that seemed to solve this problem is the LebMeasure algorithm[2, 3]. The original analysis of LebMeasure claimed that this algorithm has complexity $O(m^3n^2)$ for m points in n objectives, thus allowing the efficient calculation of hypervolume for large sets of points in large numbers of objectives. However, we prove in this paper that although LebMeasure is polynomial in the number of points, it is exponential in the number of objectives in the worst-case, and thus its performance deterio-

C. A. Coello Coello et al. (Eds.): EMO 2005, LNCS 3410, pp. 326–340, 2005.

rates sharply as the number of objectives increases. We present three strands of evidence.

Empirical Data. We give some pathological patterns of sets of points that clearly exhibit exponential slowdown under LebMeasure as the number of objectives increases.

A Lower-Bound on the Complexity. For one pattern of sets of two points, we describe their slowdown with a recurrence relation in the number of objectives n, and we prove that this recurrence relation is equal to 2^{n-1}, thus proving that LebMeasure is exponential in the number of objectives in the worst case.

An Upper-Bound on the Complexity. It is hard to establish an exact complexity for LebMeasure in the general case, because it is difficult to be certain which sets of points will suffer the biggest slowdown. However, we can establish a likely upper-bound on the worst-case complexity by considering illegal sets of points that are likely to perform worse than any legal sets of points[1]. We do this for one pattern of sets of points, defining a recurrence relation for their slowdown in m and n, and proving that this recurrence relation is $O(m^n)$.

Taken together, this evidence demonstrates that in the worst case, LebMeasure is exponential in the number of objectives, and thus that it cannot provide a general solution to the performance problem (currently) inherent in calculating hypervolume.

The rest of this paper is structured as follows. Section 2 describes LebMeasure and its behaviour. Section 3 gives an empirical analysis of some patterns of sets of points which exhibit exponential growth (in terms of the number of objectives) under LebMeasure. Section 4 gives a recurrence relation for one pattern of sets of points, and proves that the time to process this pattern grows exponentially with the number of objectives. Section 5 discusses the possibility of placing an upper-bound on the complexity of LebMeasure, and Section 6 concludes the paper and outlines some future work.

2 The LebMeasure Algorithm

2.1 The Definition of Hypervolume

Given a set S containing m points in n objectives, the hypervolume of S is the size of the portion of objective space that is dominated by at least one point in S. We say that a point $\overline{x}$ dominates a point $\overline{y}$ iff $\overline{x}$ is at least as good as $\overline{y}$ in every objective *and* $\overline{x}$ is strictly better than $\overline{y}$ in at least one objective. The hypervolume of S is calculated relative to a reference point which is worse than (or equal to) every point in S in every objective. Given two sets of points S and S' containing solutions to a given problem, if S occupies a greater hypervolume than S', then S is taken to be a better solution to the problem.

[1] A set of points is legal only if its members are mutually non-dominating.