Exercise: Runtime Analyses of a Simple Evolutionary Multiobjective Optimization Algorithm

Advanced Control lecture
at Ecole Centrale Paris

Anne Auger and Dimo Brockhoff
firstname.lastname@inria.fr

March 14, 2013

Abstract

Using fitness-based partitions turned out to be a basic approach to prove upper bounds of the expected runtime of simple single-objective algorithms such as the (1+1)EA on easy pseudo-boolean functions. This approach can also be used to analyze simple evolutionary multiobjective optimizers. In this exercise, we consider the Simple Evolutionary Multiobjective Optimizer (SEMO) on the Leading Ones Trailing Zeros (LOTZ) problem.

1 Definitions

Definition 1 *The bi-objective maximization problem with objective function* $f(x) = (LO(x), TZ(x))$ *where*

$$LO(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j \qquad \text{and} \qquad TZ(x) = \sum_{i=1}^{n} \prod_{j=i}^{n} (1 - x_j)$$

for $x \in \{0,1\}^n$ *is termed the* Leading Ones Trailing Zeros (LOTZ) *problem.*

Algorithm 1 Simple Evolutionary Multiobjective Optimizer (SEMO)

init:
$t = 0$
choose $x^0 \in \{0,1\}^n$ uniformly at random
set population $P = \{x^0\}$
repeat
  choose a parent p uniformly at random from P
  mutate p to create child c by flipping one bit in p uniformly at random
  evaluate c on the objective functions
  if $\nexists q \in P : q \succ c$ **then**
    $\forall r \in P$ for which $c \succeq r : P = P \backslash \{r\}$ {delete dominated individuals}
    $P = P \cup \{c\}$ {add child to population}
  $t = t + 1$
until the end of time

In the following, we want to analyze the expected runtime of the Simple Evolutionary Multiobjective Optimizer (SEMO) the pseudo code of which can be found as Algorithm 1 below.

2 Questions

a) To understand better the LOTZ problem, draw a graph that illustrates the objective space for $n = 6$.

b) What are the individual optima of the two objective functions of LOTZ?

c) What is the Pareto set and what the Pareto front of LOTZ?

d) What is the expected time of SEMO to find the first Pareto-optimal point of LOTZ? To prove an upper bound on the expected time, use the following guiding questions:

- Where will, with high probability, the initial population of SEMO be in objective space? And why?
- Which objective vectors can be reached by SEMO's mutation from a given objective vector?
- What is special about SEMO's population on LOTZ until the first Pareto-optimal point is found?

- Use the fitness-based partitions idea to prove an upper bound on the number of function evaluations until the first Pareto-optimal point is found.

e) How long does it take to cover the entire Pareto front once one or more Pareto-optimal solutions are contained in SEMO's population?

   - Is it possible that SEMO loses a once found Pareto-optimal solution in general? Is it different for the considered LOTZ problem? Pay careful attention to the used dominance relations in the selection of Algorithm 1. Why, do you think, is the selection of SEMO designed the way it is?
   - How long (upper bound!) does it take for SEMO to come up with a new, never-visited Pareto-optimal solution?
   - What is a well-suited, simple fitness-based partition for analyzing SEMO solely on the Pareto front and how can you prove an upper bound on the above mentioned time to cover the entire Pareto front?