Appendix C

Population and evolutionary based methods

C.1 Introduction

In this appendix a brief overview of various adaptive stochastic approaches to global optimization are detailed. This presentation is by no means exhaustive and is meant only as an introduction to other related approaches to global optimization, some of which are, like particle swarms, inspired by phenomena found in nature.

The algorithms listed are all based on random sampling in the feasible region. Some of these approaches can be applied to both continuous and discrete global optimization problems.

C.1.1 Evolutionary computation

The evolutionary computation approach mimics the process of natural evolution by which superior individuals are selected from a population to generate offspring, which inherit disturbed parental genetic information. By this process of evolution simulation it is attempted to obtain individuals with above-average fitness. Reproduction of individuals is non-deterministic, allowing for the continual production of new genetic information (mutation).

Evolutionary computation can be subdivided into three strongly related but independently developed approaches [52], namely genetic algorithms, evolutionary programming and evolutionary strategies.

Genetic algorithms

Genetic algorithms (GA's) were first introduced by Holland [53, 54, 55] and subsequently studied by De Jong [56, 57, 58], Goldberg [59, 60, 61, 62], and others.

The genetic algorithm approach entails a 'population' of candidate solution points which is sequentially 'evolved' in a heuristic process which mimics biological evolution as found in nature. The adaptive search consists of a competitive selection process where the least fit candidates in the population have a low probability of survival. The remainder are then 'recombined' or 'paired' with other candidates by exchanging components or 'genes'. There is also a 'mutation' operator which may (randomly) adjust one or more set of genes in a candidate. Since this process of recombination and mutation occurs sequentially, each generation of candidate solution points will be biased toward regions in the problems space of increased fitness. Genetic 'drift', caused by mutation, prevents the search from stagnating and provides a means for refined 'local' search during the terminal phase of the search.

Evolutionary programming

Evolutionary programming (EP) was first introduced by Fogel [63, 64], and further studied by Atmar [65], Burgin [66, 67] and others.

Evolutionary programming was initially offered as an ambitious means of creating artificial intelligence. It involved the evolution of finite state machines to predict events on the basis of former observations. A finite state machine is an abstract machine which takes a sequence of symbols as an input and transforms them by means of a finite set of transition rules with finite states to a sequence of output symbols. The performance of this machine is usually gauged by its ability to predict events correctly.

Evolutionary strategies

Evolutionary strategies (ES) were first developed by Rechenberg [68, 69] and Schwefel [70, 71], and extended by Herdy [72], Kursawe [73] and others.

This approach was originally designed with the objective of solving difficult discrete and continuous parameter optimization problems. The main difference between ES and GA's lie in the calculation of fitness of a specific genotype, and the manner in which the operators (mutation, recombination and selection) manipulate this genotype. More specifically, while mutation is only used in GA's to avoid stagnation, this operator becomes the primary means of 'evolving' toward a solution in ES. A further difference is the manner in which selection is applied. Selection in the case of ES is absolutely deterministic, whereas this is not the case in the context of GA's. Therefore, arbitrary small differences in fitness can play a large role in deciding on the survival of a individual in ES.

C.1.2 Simulated annealing

This Monte Carlo based approach to global optimization is inspired by a physical analogy of the atomic structure in a crystalline material which strives to arrive at a stable configuration (minimum potential energy, globally or locally). This approach was first applied by Metropolis *et al.* [74] and subsequently developed further by van Laarhoven and Aarts [75] and Webb [76], amongst others.

In the analogy the current energy state of the thermodynamic system is equivalent to the current solution to the combinatorial problem; the energy equation for the thermodynamic system is analogous to the objective function, and the ground state is analogous to the global minimum. The major difficulty in implementation of the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrapment in local minima (quenching) is dependent on the "annealing schedule", i.e. the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds.

The simplified working of the algorithm is as follows: A random step in the problem space is taken and the energy state (objective function) evaluated at this position. Any step which yields a overall decrease in the overall energy state is accepted. The step size is reduced as the search progresses to facilitate a more refined search as the minima is approached.

C.1.3 Ant colony optimization

The ant colony optimization approach was first introduced by Dorigo *et al.* [77] and extended by Stutzle and Hoos [78, 79] among others.

The ant colony optimization (ACO) meta-heuristic takes it's inspiration from the foraging behavior of ants, in particular their ability to find the shortest routes between their nests and food sources. While traveling to and from these food sources ants deposit a pheromone trail which serves as markers, to aid themselves and other ants to return to the nest or to find the route to the food source. The ants are able to find the shortest route to their food supply when presented with a set of alternate routes by information inherent in the pheromone trail. Shorter routes get a higher density of pheromone deposit, because ants which choose this route, per chance, will more rapidly reconstitute the interrupted pheromone trail than those who choose the longer path. Because ants tend to choose, by probability, to follow the strongest pheromone trail, the pheromone will accumulate more rapidly on the shorter routes. This reinforces the use of shorter paths by process of positive feedback. This method of optimization has been successfully applied to the well known traveling salesman problem, among others.

C.1.4 Tabu Search

This method was first introduced by Glover [80] and studied further by Laguna [81]. The motivation behind this type of approach is to 'forbid' search moves to previously explored search points in the (usually discrete) solution space. With the tabu search philosophy it is sometimes allowable to temporarily accept new inferior solutions to avoid paths already investigated. This approach can lead to exploring new regions of D, with the goal of avoiding local minima and ultimately finding a solution by 'globalized' search.

Tabu search has traditionally been applied to combinatorial optimization (e.g., scheduling, routing, and traveling salesman) problems.The technique can be adapted to include continuous problems by discrete approximation (encoding) of the continuous problem (in a similar

fashion as done in genetic algorithms).

C.2 Clustering methods

The clustering method among others, was first presented by Becker and Lago [82] and remained unexploited until the work of Törn [83, 84]. It was further developed by a variety of authors (Dixon and Szegö [42]). Clustering global optimization methods can be viewed as a modified form of the standard multi-start approach, which performs a local search from several points distributed over the entire search domain. A drawback of multi-start is that when many starting points are used, the same local minimum may be identified several times, thereby leading to an inefficient global search. Clustering methods attempt to avoid this inefficiency by carefully selecting points at which the local search is initiated. The three main steps of clustering methods are:

(1) Sample points in the search domain D,

(2) Transform the sampled point to group them around the local minima, and

(3) Apply a clustering technique to identify groups that (hopefully) represent neighborhoods of local minima.

If this procedure successfully identifies groups that represent neighborhoods of local minima, then redundant local searches can be avoided by simply starting a local search at some point within each cluster.