Intelligent Systems Engineering

EC3 Evolutionary Optimization

Mathematical Considerations

Petr Musilek

University of Alberta

Outline

1. Convergence
    - Premature Convergence
2. Representation
    - Binary Representation
    - Other Representations
3. Selection
    - Plus/Comma Selection
    - Proportional Selection
    - Tournament Selection
    - Linear Ranking Selection
    - Extensions

These notes are based on [Keller et al. 2017] chapter 11.

Some Mathematical Considerations

Analysis of optimization approaches can be performed

- analytically (i.e., by mathematical theorem and proof)
- by empirical observation (made effective by increase speed of computers)

Empirical observation means that it is often possible to arrive at a good understanding of the mathematical properties of an approach through statistical estimation based on repeated sampling of a given procedure on a particular problem of interest (this is limited to that particular set of observations and it cannot be generalized those results to other problems).

Early in the development of EC techniques, there were broad beliefs that certain design choices would offer superior performance generally:

- binary representation would offer an intrinsic advantage over other representations regardless of the problem
- proportional selection would provide an optimal way to create offspring in the search for improved optimization performance

However, mathematical analysis has shown that there truly is no single best approach to computational problem solving generally, a.k.a. no free lunch principle.

There is always a challenge of generalizing the results of an evolutionary algorithm on a particular problem to other problems, even though they may appear to be closely related.

Convergence

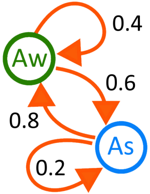
Convergence is the tendency for populations in evolutionary computing to stabilize over time

- it causes evolution to halt because all individuals in the population are identical
- full convergence might be seen in algorithms using only crossover (cf. PMX in slide set 15)
- premature convergence - population is 'stuck'; it has converged to a single solution, that is not highly valued (e.g. close to global optima)
- can be avoided with a variety of diversity-generating techniques
- not necessarily a negative thing - populations often stabilise after a time, in the sense that the best individuals all have a common ancestor and their characteristics are very similar/identical (to each other and from generation to generation)

Convergence with Probability 1

To show that EAs can converge to an optimum solution with probability 1, they can be constructed in a Markovian framework. The essence of this approach is to consider different configurations of an evolving population to be states in a Markov chain.

A Markov chain is a stochastic process (a time-indexed sequence of random variables). It is defined over a set of states (e.g., "awake" and "asleep"). The probability of transitioning from one state to another is time invariant and depends only upon the current state, e.g.



$$\begin{array}{ccc} & \text{Awake}[t+1] & \text{Asleep}[t+1] \\ \text{Awake}[t] & \left[\begin{matrix} 0.4 \\ 0.8 \end{matrix}\right. & \left.\begin{matrix} 0.6 \\ 0.2 \end{matrix}\right] \\ \text{Asleep}[t] & & \end{array}$$

Since the transition probabilities depend on the states in the system and do not depend on any particular time, the probability of being in a state of two time steps in the future can be found by multiplying the probability matrix

$$\boldsymbol{P} = \begin{bmatrix} 0.4 & 0.6 \\ 0.8 & 0.2 \end{bmatrix}$$

by itself (squaring the matrix). Iterating from one point in time to another, say, k steps ahead, in a Markov chain is accomplished by raising P to the k^{th} power. E.g. 2-step transition matrix is

$$\boldsymbol{P}^2 = \begin{bmatrix} 0.64 & 0.36 \\ 0.48 & 0.52 \end{bmatrix}$$

So, if you are in the state Awake at a time t, then there is a 0.64 probability that you will be in the state Awake at time $t + 2$.

It is natural to think about what state you would be in if time went to infinity. In our case,

$$\boldsymbol{P}^{\infty} = \begin{bmatrix} 0.5714 & 0.4286 \\ 0.5714 & 0.4286 \end{bmatrix}$$

That is, no matter which state you are in at time t, there is (approximately)

- a 0.5714 probability that you will be in the state Awake as time goes to infinity, and
- a 0.4286 probability that you will be in the state Asleep as time goes to infinity.

The same principle can be used to study the long-term behavior of some evolutionary algorithms in which the time history of how the population arrived at its present state is not pertinent to the population's future trajectory, and also the probabilities for transitions for one configuration to another are fixed (stationary).

If an evolutionary algorithm is constructed

- with a form of selection called elitist selection (the absolute best solution/s in the population is always retained into the next generation), and
- such that a mutation operation can reach any state with nonzero probability (e.g., applying a Gaussian random mutation on all parameters),

then the transition matrix for the Markov chain can be written as $\boldsymbol{P} = \begin{bmatrix} 1 & \mathbf{0} \\ \boldsymbol{R} & \boldsymbol{Q} \end{bmatrix}$

where 1 is a 1×1 identity matrix (the absorbing state that has the global best solution), $\boldsymbol{R}$ is a strictly positive (all entries > 0) $t \times 1$ submatrix, $\boldsymbol{Q}$ is a $t \times t$ transition submatrix, $\mathbf{0}$ is a $1 \times t$ matrix of zeros, and t is a positive integer based on the size of the state space.

Essentially, if the population already contains the best possible solution, then it is in the state "1" and will stay there forever. If the population doesn't contain the best solution, then the submatrix $\boldsymbol{R}$ describes the probabilities of transitioning to "1" in the next step, and submatrix $\boldsymbol{Q}$ describes the probabilities of transitioning elsewhere.

This is a special case of a more general transition matrix

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{I}_a & \mathbf{0} \\ \boldsymbol{R} & \boldsymbol{Q} \end{bmatrix}$$

where $\boldsymbol{I}_a$ is an $a \times a$ identity matrix. In the limit, as k tends to infinity,

$$\lim_{k \to \infty} \boldsymbol{P}^k = \begin{bmatrix} \boldsymbol{I}_a & \mathbf{0} \\ (\boldsymbol{I}_t - \boldsymbol{Q})^{-1}\boldsymbol{R} & \boldsymbol{Q} \end{bmatrix}$$

The components "0" indicate that given infinite time, there is zero probability that the chain will be in a state that is not an absorbing state, and the absorbing state(s) was defined as a state(s) that contains a global optimum due to elitist selection. So, this means that there is convergence with probability 1 to a global optimum. Special properties of the matrix entries $\boldsymbol{R}$ and $\boldsymbol{Q}$ provide for this result, implying a stronger form called complete convergence.

This result is of limited utility (infinite time to discover a globally optimal solution); in addition, without elitist selection and without a mutation operation that can reach all possible states, this proof of convergence with probability 1 does not hold.

In particular, if crossover is substituted for mutation, then the result does not guarantee convergence to a global optimum, but rather only to a homogenous state in which all solutions are identical and therefore no new solutions are possible.

If this homogeneous solution is not the global optimum (or is less than desirable), so called premature convergence occurs. The term is often used incorrectly to describe the effect of converging to a local optimum, but the origin of the term applies directly to the case in which no further progress is likely because the population lacks diversity, which effectively stalls an evolutionary algorithm that is heavily reliant on crossover or recombination (cf. the PMX example with TSP). It also appears that there is more likelihood of stalling at a point if that point is of higher quality/fitness.

Premature Convergence

The most common methods for overcoming premature convergence include

- restarting from a new randomized population,
- using heuristics to move the population to a new collection of points (e.g., by hill climbing), or
- redesigning the approach.

Early EA literature in evolutionary often recommended very high rates of crossover and very low rates of mutation, which made premature convergence more likely.

Observing repeated premature convergence in an EA suggests, at least,

- reconsidering the variation operators that are being used and
- giving consideration to modifying the probabilities of applying those operators, or
- creating new variation operators that are better tailored to the problem.

Representation

Designing an evolutionary optimization algorithm requires choices of

representation, **selection**, and **variation**

operators.

In terms of representation, there used to be a general belief that it would be beneficial to represent solutions using binary strings, e.g.

For optimization problem in $\mathbb{R}^n$, i.e. nstead of treating solutions directly as a vector

$$\boldsymbol{x} = [x_1, \ldots, x_n] \in \mathbb{R}^n$$

the solution would be transformed into a series of bits $[x_1, \ldots, x_k]$, i.e. a bit string of length k. The greater the degree of desired precision, the larger the value of k. The belief was that longer strings generated more opportunities for an evolutionary algorithm to explore the subspace of possible solutions via substrings, and that this would provide a greater “information flow”.

However, many problems do not lend themselves easily to a description in binary strings, e.g. representing a TSP as a series of 1's and 0's is anything but straightforward.

Suppose there are 5 cities. An intuitive representation would be an ordered list of cities, such as

$$[1\ 2\ 3\ 4\ 5]$$

Encoding these in binary could be done as

$$[001\,010\,011\,100\,101]$$

with each 3-bit segment corresponding to the number of a city. But this representation is not easily varied by mutation or recombination; e.g., mutating the fifth bit from 1 to 0 yields

$$[001\,000\,011\,100\,101]$$

and then the second city to visit is city "zero," which does not exist. Similarly, crossing two such bit strings would likely generate offspring that would not correspond to legal tours.

Fortunately, the notion that binary representations are always better than other representations is false

- in fact, there is no “best” representation across all problems, and
- under some conditions there is a provable mathematical equivalence of representations of different cardinality

Thus, the choice of a representation for a particular problem is often a matter of which provides the greatest intuition to the practitioner as the problem solver: with experience, they can gain better intuition about the effects of different representations, and how they are coupled with variation operators in order to search a solution space (landscape) for successively better answers to a problem of interest.

binary – real – integer – permutation-based – messy encodings – tree structures

Some important aspects to consider when selecting a representation include:

- It should optimally provide immediate information about the solution itself. For example, in the TSP, the list of cities is suggestive of the solution.
- It should be amenable to variation operators whose mathematical properties are well understood and that can exhibit a gradation of change (i.e. they should be available to make both small changes and big changes to any given parent(s), and that the likelihood of these different sized changes can be controlled).

  E.g. a Gaussian variation operator allows generating offspring that are close to or far from a parent, and this can be controlled by changing the standard deviation in each dimension.
- Unless the objective is to explore the utility of a novel representation, utilizing representations that have been studied and for which results have been published may allow more systematic and meaningful comparisons.

Selection

Selection describes

- either the process of eliminating solutions from an existing population
- or making proportionally more offspring from certain parents.

Some common forms of selection include

- plus/comma,
- proportional,
- tournament, and
- linear ranking.

Each has different effects on the likelihood of particular individuals to survive as parents or create offspring, and thus each has conditions that favor or disfavor its utility.

Plus/Comma Selection

The notation $(\mu+\lambda)$ and (μ,λ) are now commonplace in EAs; they refer to the cases where

$(\mu+\lambda)$ μ parents create λ offspring and the best μ individuals are selected from among the $\mu+\lambda$ to be parents of the next generation, or

(μ,λ) μ parents create λ offspring and the best μ individuals are selected only from among the λ offspring to be parents for the next generation.

Thus,

- in $+$ selection, parents and offspring compete to be parents for the next generation, while
- in , selection, the parents die each generation and a surplus of offspring must be created.

Some variations of these approaches include

1. the case of $\mu+1$, which is sometimes referred to as "steadystate" or "continuous" selection, and
2. allowing parents to survive for some maximum number of generations g before being removed in the comma selection process.

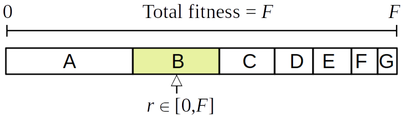
Proportional Selection

Proportional selection picks parents for reproduction in proportion to their relative fitness; thus

- the procedure is constrained to maximization problems on strictly positive fitness scores
- if the goal was to find the minimum off a function $f(\cdot)$ while using using proportional selection, the problem would need to be turned first into a maximization problem, such as find the maximum of $1/f(\cdot)$

The probability of selecting an individual in the population for reproduction is determined as

$$P_i = f(i)/\sum_{j=1}^{\mu} f(j)$$

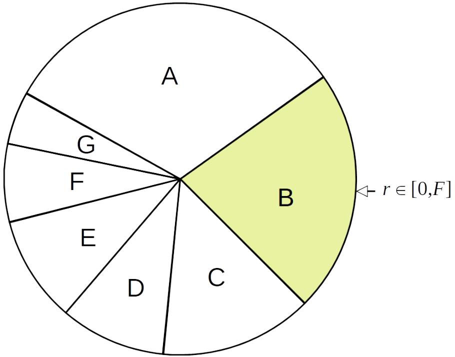


where p_i is the probability of selecting the ith individual,
there are μ existing individuals, and $f(i)$ is the fitness of the i^{th} individual.

This is sometimes also called roulette wheel selection (see the illustration below).

Proportional selection is applied to individuals generally by selecting one individual for mutation, or two (or more) individuals for recombination until the population size for the next generation has been filled.

Rather than working directly on the fitness values, proportional selection can also work on the relative ranking of solutions (thus making it applicable to minimization problems).



What would happen if there was a "super-individual" s with $f(s) \gg f(i); i \neq s$?

Tournament Selection

The more common of many different forms of tournament selection

- selects a subset of size q (often $q = 2$) from the existing population
- and selects the best of those q individuals to be a member of the next generation.

The process is repeated until the population is filled.

The process can be conducted with or without replacement, that is, individuals that are selected out of a q-tournament can be given an opportunity (or not) to be selected again in another q-tournament.

As with proportional selection, tournament selection allows the possibility that solutions that are less than best can propagate into a future generation.

Linear Ranking Selection

Linear ranking selection maps individuals to selection probabilities according to a prescribed formula based on the rank of the solution.

An early approach (among many variations) assigned a probability to the i^{th} ranked individual as

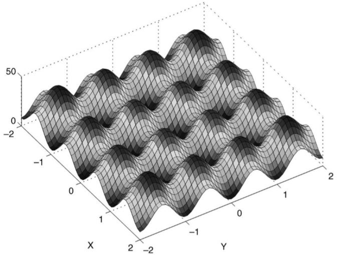
$$p_i = (\eta^+ - (\eta^+ - \eta^-)[i-1]/[\mu-1])/\mu$$

where p_i is the probability of selecting the i^{th} individual, μ is the number of individuals in the population, η^+ and η^- are user-controlled constants constrained by $1 \leq \eta^+ \leq 2$; $\eta^- = 2 - \eta^+$. For example, if $\mu = 100$ and $\eta^+ = 1.5$, the probability of selection of the i^{th}-ranked solution is

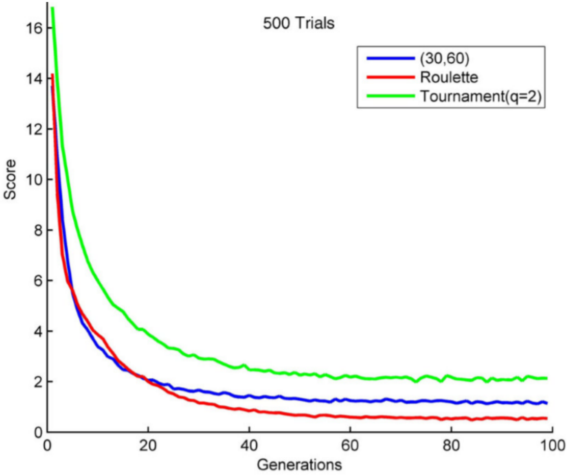
$$\begin{aligned} P_1 &= (1.5 - (1)(0/99))/100 = 0.0015 \\ P_2 &= (1.5 - (1)(1/99))/100 = 0.00148989\ \ldots \\ P_{100} &= (1.5 - (1)(100/99))/100 = 0.0048989\ \ldots \end{aligned}$$

Thus, better solutions are favoured over lesser solutions.

Effects of different selection operators: consider a simple EA to search for the minimum of Rastrigin's function $(f(x,y) = 20 + x^2 - 10\cos(2\pi) + y^2 - 10\cos(2\pi y))$ in 2D (dimensions).



- initialization: 30 ind. $U[10,10]$ in each D
- variation: $N(0,0.25)$ applied to each D



Extensions to Basic Forms of Selection

- Elitist selection: automatically ensures that the best solution in a population is retained in the next generation.
- Nonlinear ranking: permits higher selective pressures

In generally, there is a continuum of selection methods from “soft” to “hard”

- the harder the selection, the faster the better solutions can overtake the population
- this can be described in terms of **takeover time**: the expected number of generations required to fill a population with copies of what is initially a single best individual when applying only selection (no variation)
- the order of selection strength from weakest to strongest:

proportional – linear ranking – tournament selection – plus/comma