Project 1 — Finding "Good" Distance Preserving Mappings

Last Updated Fr., Sept. 3, 1999, 2p.

Due Date: Fr., Oct. 22, 1999 11p (electronic submission); if you take the late option you due date is We., Oct. 29, 1999, 11p.

Recommended Schedule for the Project1:
Sept. 10-15: Try to understand the problem you have to solve; think about possible subproblems and possible approaches, you might explore.
Sept. 15-Oct.17: Perform your chosen project
Sept 23-24: design and submit benchmark
Oct. 17-22: Write your project report.

Project Description (Initial Specification)

The objective of this project is to find good mappings

$$\psi : [0,1]^n \rightarrow [0,1]^2$$

from n-dimensional space to 2-dimensional space so that 2D representation is a good approximation for the distances of points in n-dimensional space. For example, if p1 and p2 are very close to each other in n-dimensional space, $\psi(p1)$ and $\psi(p2)$ should be very close in 2-dimensional space, whereas if p1 and p2 are very far from each other $\psi(p1)$ and $\psi(p2)$ should also be very far away from each other in 2-dimensional space. We call mappings that satisfy this property *distance-preserving*. In general, finding good distance preserving mappings is a complex optimization problem; it has several important practival applications in data visualization.

In more detail, the problem is to find for a given set of points P in n-dimensional space a mapping ψ that returns the minimum (or a low value) for the error-function e given below:

Let
d_n be a n-dimensional distance functions
d_2 be a 2-dimensional distance function
$P \subset [0,1]^n$ be the set of points to be mapped
be a functions that computes the cardinality of a set

$$e(P,\psi) = \frac{\sum_{p,p' \in P (p \neq p')} |d_n(p,p') - d_2(\psi(p),\psi(p'))|}{(\#(P))^2 - \#(P)}$$

Frequently, an additional constraint (C) is used to restrict the maximum error per point:

$$(C)\ \forall p1 \in P \forall p2 \in P (|d_n(p1,p2) - d_2(\psi(p1),\psi(p2))| \ \leq \ tol)$$

with $tol \in (0,1)$.

Only one distance functions will be considered in the project which is called normalized absolute distance (NABS), whose definition is given below:

Let $p = (x_1, ..., x_n)$ and $p' = (x'_1, ..., x'_n)$

$$(NABS)\ d_n(p, p') = \frac{\sum_{i=1}^{n} |x_i - x'_i|}{n}$$

In the project an evolutionary computing approach has to be employed that uses the n-dimensional distance matrix for P as an input, and tries to find an optimal placement of m points in 2D, minimizing the errorfunction e with respect to the normalizeed absolute distance function. Use roulette wheel as your selection method, when using this approach.

For this particular project, you can assume assume that n is either 3 or 9 (we will have benchmarks that involve 3-dimensional, and 9-dimensional approximations), and the cardinality m of P is 20 or 100. Moreover, a benchmark of example problems will be generated (all students in the class will be participating in producing this benchmark) during the course of the project that has to be run to test the performance of your approach. Each student has to provide a single benchmark near the end of September; students belonging to the Sun-Group provide a 3-dimensional benchmark, whereas each student of the Moon-Group provides a single 9-dimensional benchmark.

It is important that you run your system for the benchmark that should be available approx. October 5, 1999. Report the average fitness of the final generation, best solutions found, including their fitness encountered for each run belonging to the benchmark with respect to the (NABS) measure — even if you do not use (NABS) as your fitness function during the evolution process.

Other topics that possibly can be investigated in the project include:

- Does your method scale up, if m is increased? Compare the error of making 9 approximation with 20 points with the error obtained for making one approximation for 100 points!
- Conduct knowledge discovery experiments in which you give a particular structure in high-dimensional space, and see if it can be visualized in 2-D-space in a way that the n-dimensional structure becomes visible.
- Explore to which extent the distance preserving mapping can be used for data compression tasks?
- Consider f with the constraint (C)
- Explore the use of data visualization techniques to use distance preserving mappings to visualize convergence of an evolutionary computing systems.
- Use visualization techniques for the better interpretation of your project results.
- Employ another search technique (e.g. a simple hillclimbing strategy), and compare its found results with those obtained with your GA/GP approach.
- Explore to which extent distance preserving mappings can be used to detect clusters in a data collection.
- If you have any other idea, just talk to me, so that I can let you know if this idea is suitable for the project.

You can use any programming language to perform the project. Moreover, you can analyze other interesting aspects of the problem. The only constraint is that you have to use evolutionary computing to find a good approximation ψ, and you have to run your approach for a benchmark (but you can also conduct experiments of your own). However, up to 50% of the project time can be (but not necessarily has to be) allocated to other interesting aspects of the problem.

The evaluation of the project will center on your potential to conduct a scientific project. Optimality is only one of several evaluation parameters; furthermore, your results should be clearly presented, and be conclusive. Limit yourself to one or two aspects of the problem, and explore this (these) aspect(s) systematically — do not try to do too many different things, because time is limited. Moreover, analyze and explain the results you obtained as good as you can (it is okay if you present speculations). Leave yourself sufficient time at the end to be able to write a clear and understandable report.