4M17 COURSEWORK FINAL ASSIGNMENT

Introduction

In this second 4M17 coursework exercise you will investigate the performance of two of the methods introduced in lectures on a problem that is easy to describe but harder to solve.

This assignment counts for 50% of your final grade for 4M17.

The Problem

The problem you will be attempting to solve is *Keane's Bump Function* (KBF). This is an n-dimensional constrained optimization problem defined as follows:

Maximize
$$f(\mathbf{x}) = \left| \frac{\sum_{i=1}^{n} (\cos x_i)^4 - 2\prod_{i=1}^{n} (\cos x_i)^2}{\sqrt{\sum_{i=1}^{n} i(x_i)^2}} \right|$$

subject to
$$0 \le x_i \le 10$$
$$\prod_{i=1}^{n} x_i > 0.75$$
$$\sum_{i=1}^{n} x_i < \frac{15n}{2}$$

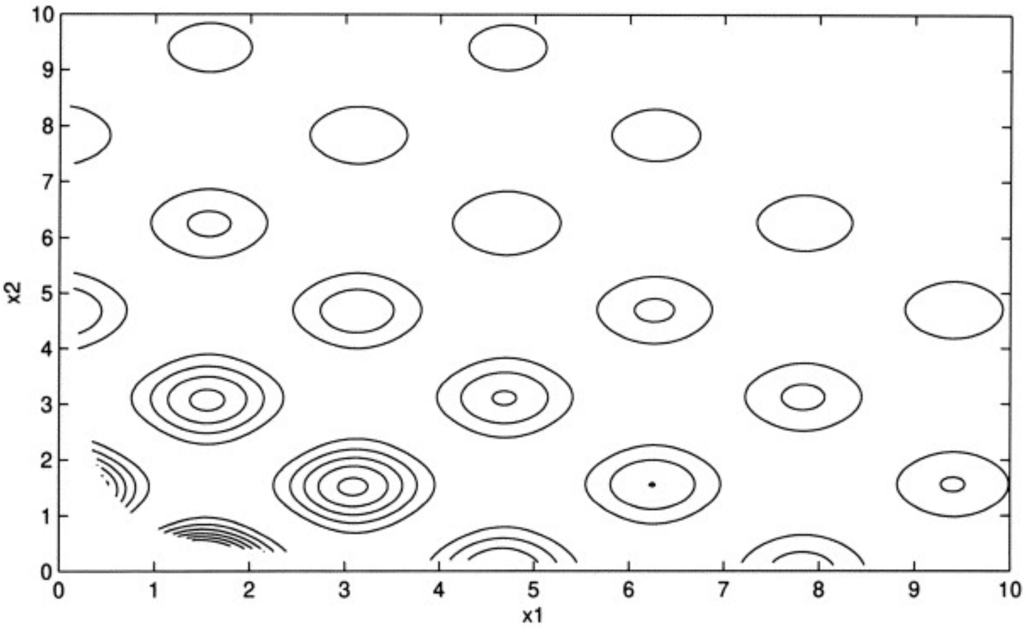
The 2-dimensional form of this problem, which is the only one easy to visualise, is therefore:

Maximize
$$f(x_1, x_2) = \left| \frac{(\cos x_1)^4 + (\cos x_2)^4 - 2(\cos x_1)^2 (\cos x_2)^2}{\sqrt{x_1^{\,2} + 2x_2^{\,2}}} \right|$$

subject to
$$0 \le x_1 \le 10$$
$$0 \le x_2 \le 10$$
$$x_1 x_2 > 0.75$$
$$x_1 + x_2 < 15$$

As shown overleaf in Figure 1, this is a problem with multiple local maxima (not quite symmetrical in $x_1 = x_2$) and a global maximum at a constraint boundary. It is therefore a hard optimization problem to solve.

On the plus side, however, all the control variables are of the same type (continuous variables) and have similar (in fact, identical) scales, all the constraints are inequality ones, and the feasible space is not disjoint.



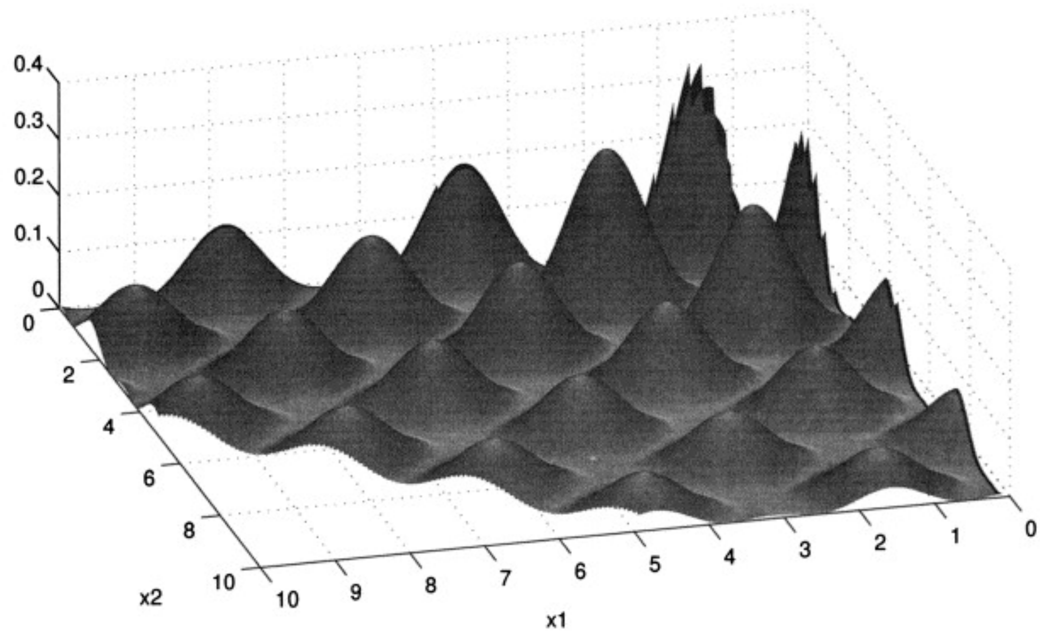


Figure 1: A contour map and a 3D plot of Keane's Bump Function for $n = 2$.[1]

Coursework Tasks

The aim of this coursework exercise is to compare the performance of two of the optimization algorithms you have learnt about in 4M17 on the **8-dimensional** version of Keane's Bump Function (8D-KBF).

[1] Image taken from: M.A. El-Beltagy and A.J. Keane "A comparison of various optimization algorithms on a multilevel problem", *Engineering Applications of Artificial Intelligence* **12**, 639–654 (1999).

One of the algorithms **must** be one of the methods covered in Professor Parks' lectures (i.e. Simulated Annealing, a Genetic Algorithm, an Evolution Strategy or Tabu Search); the other algorithm can be another of these methods or another stochastic global optimization method (e.g. Particle Swarm Optimization). The option allowing your own choice of the second optimizer is particularly geared towards Part IIB students who might be using such a method in their MEng project and postgraduate students who might be using such a method in their research.

You may **either** write your own optimization codes **or** find and use suitable (free) software on the internet, or indeed write one code and find the other. Links are provided to some resources in the *Final Assignment Resources* section of the 4M17 Moodle site. However, you are not restricted to using these.

Any popular programming language (MatLab, C++, Python, Fortran etc.) can be used.

You should only use software sourced from the web for which you can obtain the source code, e.g. the uncompiled C++, to enable you to debug it, if necessary, and perhaps to modify it, if it does not do exactly what is required.

Investigate the performance of the algorithms you have sourced or implemented on the 8D-KBF. Allow up to 10,000 objective function evaluations in each algorithm run you execute. In analysing the performance of the algorithms, bear in mind the comments made on performance measures in the *Common Issues* section of the lecture notes.

Investigate the effects on the performance of your algorithms of varying some of the parameters or implementation options that control them; for instance, for a Genetic Algorithm investigate the effect of different population sizes/numbers of generations, different crossover and mutation probabilities, different selection schemes etc. An exhaustive exploration of all possibilities is not required, just a systematic examination of the way in which the control parameters/implementation options chosen affect algorithm performance.

If you are using a code you have found, as opposed to one you have written, rather more investigations will be expected.

In these studies the same set of random number generator seeds should be used in each set of runs.

For population-based search methods (e.g. Genetic Algorithms and Evolution Strategies), the initial populations should be generated by random sampling; initial populations should not be 'seeded' with solutions close to the global optimum.

For search methods that start from a single solution (e.g. Simulated Annealing and Tabu Search), the initial solution should be randomly generated and varied from run to run.

Final Report: Submission

Write a report detailing and analyzing the results of your investigations. This should be uploaded as a pdf document via Moodle by 4pm on the first day of Lent Full Term, **Tuesday 16th January 2024**. Please make sure you include a completed copy of the coversheet at the front of the pdf.

Part IIB students should **only** include their Coursework Candidate Numbers (CCN), and must ensure that their names do **not** appear in the report or in any of the file names or code listings.

MPhil students who have been given an anonymous identifier similar to a CCN should use that, and must ensure that their names do **not** appear in the report or in any of the file names or code listings.

Other postgraduate students (e.g. first year PhD students) should ensure that their names and CRSids are included on their reports.

A specially prepared coursework coversheet for this exercise is available from the *Final Assignment* section of the 4M17 Moodle site.

Final Report: Content

- ☐ The report **should** include (in Appendices) listings of the source code of the algorithms you have implemented or found. The listings of codes you have written **should** be reasonably well commented. The source (e.g. the url) of any found codes **should** be clearly identified.
- ☐ Detailed descriptions of the ways in which the algorithms work are **not** required. However, if you have implemented an idea of your own (**which is by no means discouraged**) or used a feature not described in the 4M17 lecture notes – perhaps one included in a code you found – the basic principles of this **should** be explained in your report.
- ☐ Your report **should** discuss problem-specific implementation details, such as how you have chosen to handle any constraints.
- ☐ You **should** include in your report figures showing representative examples of the search patterns followed in (x_1, x_2) space by the two optimization methods you have tested when tackling the **two-dimensional** version of Keane's Bump Function (2D-KBF). Such figures are useful for confirming that the algorithms are performing as expected.
- ☐ The **main focus** of your report should be an evaluation of the performance of the two methods you have applied to the 8D-KBF, and a discussion of the effects on this performance of changing the algorithms' control parameters/implementation details.

Final Report: Marking

The mark awarded will depend on:

- how well written any code you have written is (lack of bugs, well commented),
- the quality of the computational experiments and investigations undertaken,
- the quality of presentation with a particular focus on the presentation of data/results,
- and the quality of performance analysis/comparison and discussion.

Practical Issues

Demonstrator Help

Apart from the scheduled coursework session you will be expected to work independently.

Questions related to the methods being investigated should be directed in the first instance to Prof. Geoff Parks (email: gtp10@cam.ac.uk).

Questions about the final report or any other logistical matters should be directed to Prof. Geoff Parks (email: gtp10@cam.ac.uk).

Computer Usage

As the problem functions (objective and constraints) for the problem to be solved are simple mathematical functions and the number of iterations allowed in each run is quite small, the running of the algorithms you write or find should not be very expensive computationally, and you should be able to do so on-line on the CUED Teaching System without offending anyone. Obviously, if you have access to good computing facilities of your own or through your College or research group (in the case of postgraduate students), it may be more convenient to use these. If your laptop is up to the task, feel free to use that.

When you come to investigate the effects of changing the control parameters/implementation details of your algorithms and need to do series of runs where only the random number seeds are being changed, you may find it more convenient to run these using a script. For specific advice/ instructions you are referred to the relevant page on the CUED help system:

https://help.eng.cam.ac.uk/central-computing-system/running-long-programs/

Summary

Key Dates

Friday 17 November 2023	9am	Coursework Introduction	CUED LR3
Friday 24 November 2023	9am	Coursework Session	CUED LR3
Tuesday 16 January 2024	4pm	Report submission deadline	via Moodle