Minimize:

\[ F(x) = \left( \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2 \prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} x_i^2}} \right) \quad (1) \]

Subject to:

\[ g_1(x) = 0.75 - \prod_{i=1}^{n} x_i \leq 0 \quad (2) \]

\[ g_2(x) = \frac{\sum_{i=1}^{n} x_i - \frac{15n}{2}}{2} \leq 0 \quad (3) \]

\[ 0 < x_i < 10 \quad i = 1, n \quad (4) \]

You are to use a Genetic Algorithm to solve the above optimization problem by writing your own GA code in your choice of computer language (Fortran, C/C++, Matlab, etc.). You are to use a value of 10 for \( n \). An example of computer code used to evaluate the objective function is provided at the end of this assignment statement. The constraints should not be difficult to code. P.S. notice the strict inequality on the side constraints.

**To Do:**

In order to determine how the various methods and parameters affect the performance of the GA, please perform the following tasks:

1. Solve the problem using 2 different crossover schemes discussed in class. Determine the relative performance of them and provide an explanation of why you think your results are as they are.

2. Solve the problem using at least 3 different values of the crossover rate. Discuss your findings regarding how the crossover rate affects the efficiency of the algorithm. Do you think this is in any way problem dependent?

3. Solve the problem using at least 3 different values of the mutation rate. Discuss your findings regarding how the mutation rate affects the efficiency of the algorithm. Do you think this is in any way problem dependent?
4. Solve the problem using at least 3 different values of the initial population size. Discuss your findings regarding how the mutation rate affects the efficiency of the algorithm. Do you think this is in any way problem dependent?

Efficiency of the algorithm is to be expressed in terms of the quality of the solution found relative to the number of objective function evaluations. Your efficiency values will be relative, not absolute, because you do not have a priori knowledge of the true global optimum for this problem.

To Hand In:
Your write-up should include the following:

- A 1-2 paragraph description of the problem you are solving.
- A discussion of the means by which you will solve it. This will include your choice of encoding, operators, convergence criteria, etc., and the reasons for your choices. It will also include a description of the trials you will run and the reasons for your trial selections.
- A discussion of not more than 4 pages describing your results. This section should contain any charts, graphs, and tables needed to thoroughly demonstrate your findings.
- A printout of your (very well commented) computer code (single spaced, small font, duplex) for my inspection.

Reminder:
This is to be an individual assignment and each student must create his/her own computer code. There is to be no sharing of code. Philosophical discussions between students are fine and even encouraged, but implementation questions should be brought to me. Also remember that quality is more important than quantity.

C Code Snip-it:
double optfun (double x[])
{
    double sum1 = 0.0, prod = 1.0, sum2 = 0.0, obj;  // Initialize storage variables.
    for (int i = 0; i<N; ++i) {
        sum1 += pow(cos(x[i]), 4);  // Numerator 1st term.
        prod *= pow(cos(x[i]), 2);  // Numerator 2nd term.
        sum2 += (double)(i+1) * pow(cos(x[i]), 2);  // Denominator
    }
    sum2 = sqrt(sum2);  // Square root in denominator
    obj = (sum1 - 2.0 * prod1)/sum2;  // Numerator / Denominator
    if(obj > 0.0) obj = - obj;  // Negative of the Absolute Value
    return obj;  // Return final obj. fctn. value.
}