

# **GA Optimization for Excel Version 1.2**

*Genetic Algorithm  
Optimization in Excel  
Spreadsheets*

**Quick Start Manual**

**Updated: October 4, 2006**

**By Alexander Schreyer**

## Introduction to Genetic Algorithms

Genetic algorithms (GAs) are based on biological principles of evolution and provide an interesting alternative to “classic” gradient-based optimization methods. They are particularly useful for highly nonlinear problems and models, whose computation time is not a primary concern. Similar to other search methods such as Simulated Annealing, they perform better than gradient-based methods in finding a global optimum if a problem is highly nonlinear and features multiple local minima. In general, GAs approach the entire design space randomly and then improve the found design points by applying genetics-based principles and probabilistic selection criteria. A thorough description of genetic algorithms can be found in [2].

Although a large number of modified algorithms are available, a GA typically proceeds in the following order:

1. Start with a finite population of randomly chosen chromosomes (“design points”) in the design space. This population constitutes the first generation (“iteration”),
2. Evaluate their fitness (“function value”),
3. Rank the chromosomes by their fitness,
4. Apply genetic operators (mating): reproduction (reproduce chromosomes with a high fitness), cross-over (swap parts of two chromosomes, chosen based on their fitness to create their offspring) and mutation (apply a random perturbation to parts of a chromosome). All of these operators are assigned a probability of occurrence,
5. Assemble the new generation from these chromosomes and evaluate their fitness,
6. Apply genetic mating as before and iterate until convergence is achieved or the process is stopped.

As can be seen above, the primary usefulness of the GA is that it starts by sampling the entire design space, possibly enabling it to pick points close to a global optimum. It then proceeds to apply changes to the ranked individual design points, which leads to an improvement of the population fitness from one generation to another. To ensure that it doesn’t converge on an inferior point, mutation is randomly applied, which perturbs design points and allows for the evaluation and incorporation of remote points.

The **main advantages** of GAs are:

- The nature of the optimization model does not need to be known. This makes GAs very interesting for complex problems or for users inexperienced in gradient-based optimization techniques.
- The optimization model and its constraints do not have to be continuous or even real values. No simplification of a problem is necessary to accommodate it to a particular algorithm (e.g. linearization).
- They are readily available and easily implemented.

The **main disadvantages** are:

- A large number of parameters need to be set. This is simplified by information from literature, but problem-specific adjustments might need to be made.

- Due to the comparatively very large number of function calls, GAs require significant computational resources. This makes them unattractive for optimization problems with computationally demanding analyses.

Developing a genetic algorithm routine for the Microsoft Excel spreadsheet program is a prudent option since:

- Excel is only shipped with a gradient-based optimization routine (SOLVER). Although it is very powerful, by the nature of the theoretical background it will not satisfactorily solve problems that are discontinuous (IF-THEN-ELSE or LOOKUP type). It may also converge at a local minimum/maximum in a highly nonlinear problem rather than the global optimum.
- Excel performs instantaneous cell-based calculations. This allows the rather large number of function calls that the GA requires to be run in a timely fashion.
- Many engineering, commerce and other problems have been implemented as Excel spreadsheets due to the pervasiveness of the Microsoft Office environment.

## Setup and Run Optimization Problem

Follow these steps to set up your model in an Excel spreadsheet and then run GA Optimization for Excel 1.2 to optimize it by varying design values and checking constraints.

1. **Create a calculation in an Excel spreadsheet.** You may use an already existing one or create a new one. If you already have a spreadsheet prepared, you may skip ahead to step 6).
2. As an example, we will be setting up a parametric formula, which features one global maximum and several local maxima. It was taken from literature [1] and is defined as:

$$F(X_1, X_2) = \cos^2(n \cdot \pi \cdot r) \cdot e^{(-r^2/\sigma^2)}$$

where:  $r^2 = (0.5 - X_1)^2 + (0.5 - X_2)^2$

$$n = 9$$

$$\sigma^2 = 0.15$$

This function is ideally suited to test a global optimization routine since it features several high and steep “ripples” in the vicinity of the optimal solution (which is at  $X_1 = 0.5$  and  $X_2 = 0.5$  with a function value of  $F = 1.0$ ). It can be expected that if a gradient-based optimization routine (e.g. the SOLVER in Excel) were started from an initial point not on the hill closest to the optimum, it would likely fail to reach the optimum since it could not escape from the surrounding ripples. Figure 1 illustrates this function in the  $X_1, X_2$ -space.

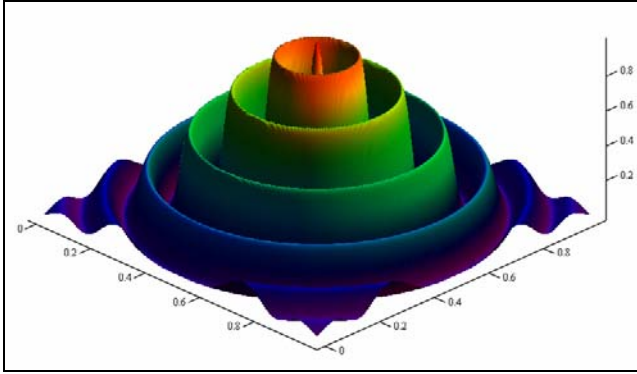


Figure 1 - Function plot

- In Excel, open a blank worksheet and designate cells for the design variables (B6 and B7 in this case). It is not necessary to enter values here at this point since starting values will be randomly generated later.

	A	B	C	D	E	F	G
4	<b>Design variables:</b>						
5				x10 (for integer calcs)			
6	X	0.50		5.00		(X=0.5 at optimum)	
7	Y	0.50		5.00		(Y=0.5 at optimum)	
8							
9	<b>Target function:</b>						
10	n	9	number of ripples				
11	s2	0.15					
12	r	0.00					
13	F	1.0000		(F=1.0 at optimum)			<b>NONLIN</b>

Then set up your target function with all of its parameters. In this case, cell B10 contains the n parameter, cell B11 contains the  $\sigma^2$  parameter and cell B12 contains the equation for r. Finally, cell B13 contains the function  $F(X_1, X_2)$ .

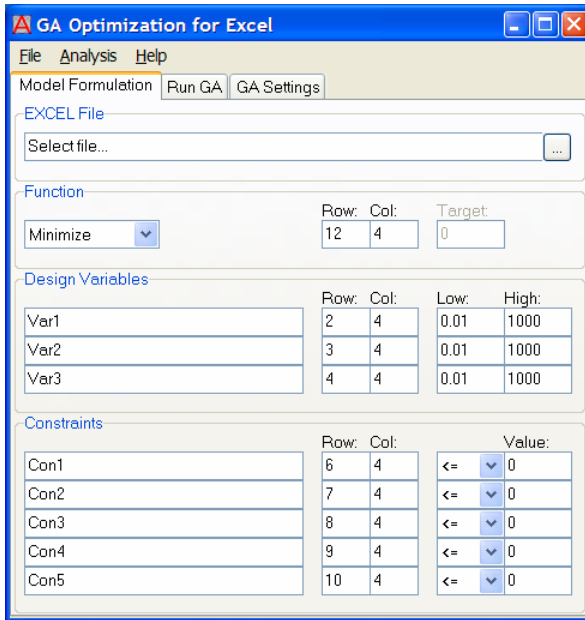
- It is always a good idea to test your Excel calculation with known values. Here, we know that the function value will be 1.0 if both  $X_1$  and  $X_2$  are 0.5. Don't forget to save the Excel sheet in a convenient location.
- We will implement this function into the maximization routine as:


$$\text{Maximize } F(X_1, X_2)$$

$$\text{such that } 0 \leq X_1, X_2 \leq 1 \quad (\text{side constraints})$$

Otherwise, this function has no constraints.

- Now **start GA Optimization for Excel 1.2**. It will open as shown below:

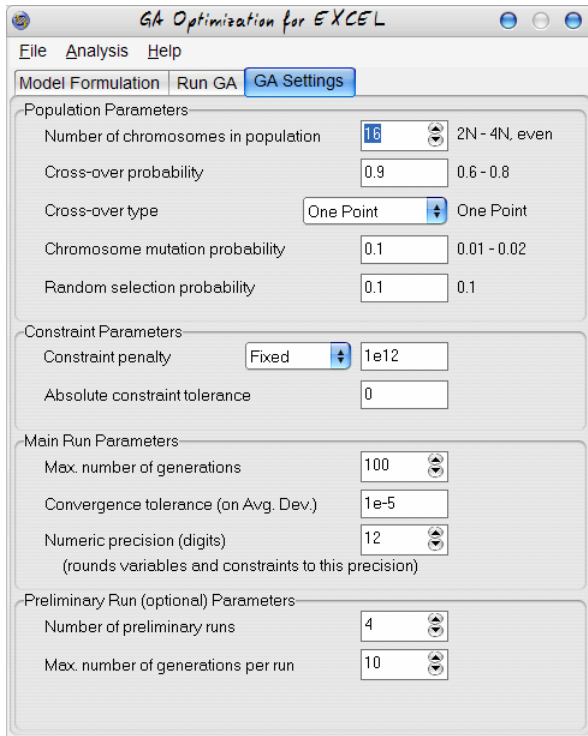


7. To link your Excel spreadsheet to this software, do the following:
  - a. Click on the file select button  and browse to the Excel file (.XLS) that you wish to use. If it isn't open already, Excel will open and show the file.
  - b. If you have calculations on several sheets in your Excel file, use the drop-down to browse to the one that contains the target function, design values and constraints. All of these cells must be on one sheet although calculations can be performed across sheets.
  - c. Select the optimization type from the drop-down (minimization, maximization, target value). If you selected "target value", then you must supply the target value to which you want to drive the target function. You may give the optimization a name as well, e.g. "Maximize profit". Enter the row and column numbers of the target function cell into the relevant boxes: For the above mentioned example, enter "13" into the "Row" box and "B" into the "Col" box.
  - d. Under "Design Variables", you may enter links to up to five cells in Excel with design variables. In our example above, these are cells B6 and B7. In the first two rows, give the variables names, e.g. "X1" and "X2". Then enter the cell references ("B", "6" and "B", "7", respectively). Finally, provide realistic bounds on these values. These are typically called side constraints and in the example above are 0 and 1, for the lower and upper bounds, respectively. If you have variables that may only consist of integer numbers, you may check the "Int" box on a per-variable basis. (This does not make sense in our example above.)
  - e. **Please note:** If you have less than five design variables, then you must specify cell references to unused cells in your spreadsheet. GA Optimization for Excel 1.2 will assign values to all five variables during each iteration and these must be put somewhere even though they are not used. This will be dealt with programmatically in future versions, but for now you have to do it this way.

- f. If you have constraints anywhere in your calculation, you must set up the constraints in the respective section. First name the constraints, e.g. “Check max design stress”. Then assign cell references. Finally, you must specify the type of constraint (“less or equal  $\leq$ ”, “greater or equal  $\geq$ ” or “exactly equal  $=$ ”) and the constraint value.  
Since it is hard to attain an exact match for an equality constraint due to numeric precision issues, it is prudent to specify an “Absolute constraint tolerance” on the “GA Settings” tab.
  - g. **Please note:** As with the design values, if you have less than five constraints, you must define “dummy constraints”. To do this, enter the value 1 into as many cells in Excel as you have unused constraints. Then define constraints for these cells of the type “Constraint in Row X / Col Y must be  $\leq 1$ ”. This way, the constraint will be ineffective during the optimization. Again, this will be dealt with programmatically in future versions, but for now you have to do it this way.
8. Now you are ready to run the optimization. Select the “Run GA” tab and click on “Run” or select “Run” from the “Analysis” menu or simply hit the F2 key.
  9. The tabs will switch to the “Run GA” tab and show you intermediate output and the final result. You can also watch the Excel spreadsheet and see how the values change. Voila! You should have an optimized result in a few seconds.
  10. If you run into trouble, then a popup dialog will usually point you to the source of the problem. In general, check the Excel file location, the settings on the “GA Settings” tab or the “Model Formulation”.
  11. Finally, it is a good idea to save the model (e.g. the GA Optimization for Excel 1.2 settings to a file. To do this, select File > Save Model As... from the menu. You can resume the model later easily by loading its definition from file.

## Settings

You may influence the Genetic Algorithm (GA) behaviour directly by modifying the settings on the “GA Settings” tab. For a more detailed description of the various options, consult a textbook on GAs, e.g. [2].



**Number of chromosomes in population** – This must be an even number. Two to four times the number of design variables is suggested, but the chance for an initial close value to the optimum is increased with the number of chromosomes. A large number will, however, increase the number of computations.

**Cross-over probability** – Determines probability that a chromosome is randomly picked for application of a cross-over operation.

**Cross-over type: One point** – cuts chromosome at one point for cross-over, **Two point** – cuts at two points, **Uniform** – applies cross-over with uniform probability, **Random** – applies randomly picked cross-over method (from the three mentioned above).

**Chromosome mutation probability** – Determines probability that a chromosome is randomly picked and mutated. This enhances the “gene pool” and may help free a calculation that got “stuck” in a particular region of the design space.

**Random selection probability** – Determines probability that a chromosome is randomly picked and carried over to the next generation without any modification.

**Constraint penalty** – This is the value that is set as the function value if a constraint is encountered. Since this is a high number, the chromosome will be assigned a low fitness, effectively removing it. You may choose the “Fixed” option if the full penalty is to be applied regardless of how close to the feasible space the design point is. For a more gradual penalty, choose the “Linear” or “Exponential” options.

**Absolute constraint tolerance** – This should only be used if an equality constraint is used.

**Max. number of generations** – Stop calculations after this number of generations.

**Convergence tolerance** – After genetic mating, an average deviation of the individual chromosomes from the population average is calculated. This is then compared to a predefined convergence tolerance value and further computations are stopped if the criterion is fulfilled as follows:

$$\frac{\sum |F_{avg} - F_i|}{n_{chromosomes}} \leq \epsilon_{tol}$$

**Numeric precision** – Determines the numeric precision for design values and constraints.

**Number of preliminary runs** – Using preliminary runs improves the initial starting generation significantly. It will add to the total number of function calls. Using even a few will improve the result greatly.

**Max. number of generations per preliminary run** – Don't set this number too high.

## References

- [1] Charbonneau, P., Knapp, B. (1995). "Users guide to PIKAIA 1.0". NCAR Technical note TN-418-IA
- [2] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley