**AN EXAMPLE OF SOLVING AN OPTIMIZATION PROBLEM USING GENETIC ALGORITHM**

A simple fisheries bioeconomic model is discussed below, where 3 classes of boat (*i*) catch 3 species of fish (*j*) in a given region. An optimal allocation of the resource is sought which maximises economic profit. Table 1 shows the data definitions. There are 5 variable classes (15 variables) in the nonlinear programme; i.e. boats (**xb**) and effort (**xe**) of boat class *i*, and catch (**xc**), landings (**xl**) and fishing mortality (**xf**) of species *j*. This problem is solved with the selected genetic algorithm solvers as well as a traditional nonlinear programming solver in order to compare the relative merits of the approaches. Both the continuous and integer forms of the model are considered, as strictly **xb** (number of boats) is an integer variable set. This problem also demonstrates the flexibility of GA and the type of models which can be considered.

**Table 1**: Model data.

|  |  |  |
| --- | --- | --- |
| Variable | Description | Values |
| Pj | Price of fish species *j* | {2.5, 2, 1.5} |
| Fi | Fixed Costs of boat class *i* | {1, 0.8, 0.8} |
| Vi | Variable Costs of boat class *i* | {0.01, 0.008, 0.008} |
| Qij | Catchability coefficient of fish species *j* by boat class *i* | |  |  |  | | --- | --- | --- | | {0.0002, | 0.0001, | 0.0 | | 0.0, | 0.0002, | 0.00005 | | 0.0, | 0.0, | 0.0002} | |
| Kj | Carrying capacity of fish species *j* | {2000, 2500, 4000} |
| Rj | Growth rate of fish species *j* | {0.1, 0.5, 0.3} |

The mathematical description of the nonlinear programme, with both a nonlinear objective function and nonlinear constraints, is given below;

https://www.economicsnetwork.ac.uk/cheer/ch13_1/images/mardl105.gif    (1)

subject to,

https://www.economicsnetwork.ac.uk/cheer/ch13_1/images/mardl106.gif    (2)

https://www.economicsnetwork.ac.uk/cheer/ch13_1/images/mardl107.gif    (3)

*xlj* £ *xcj* ( *j* =1,...,3)    (4)

**xb, xe, xf, xc, xl ³ 0**    (5)

where the objective function (1) denotes the economic profit for the fishery which is determined by revenue from landings minus the associated fixed and variable costs of the fishing. Equation (2) defines the fishing mortality, equation (3) then evaluates catch from this fishing mortality rate and equation (4) constrains landings to be no greater than catch. The (control) variables **xb**, **xe** and **xl** have upper bound values of {100, 100, 100}, {275, 160, 200} and {500, 500, 500} respectively.

By inspection, it is clear in the nonlinear programming problem above that **xf** are dependent variables, i.e. their value is dependent on the values of **xe** and **xb**. In turn **xc** is dependent on **xf**. It follows in the fitness function of the GA that these dependent variables can be simply calculated from the probabilistic (or control) variables **xe** and **xb**. A simple repair approach is implemented for equation (4) in the fitness function, where if the constraint is violated then xl\* is reset equal to the relevant xc\*. Therefore, when the constrained NLP model is built in the GA it can be interpreted as an unconstrained 9 variable model([4](https://www.economicsnetwork.ac.uk/cheer/ch13_1/ch13_1p16.htm" \l "note4)) where the returned fitness value is the NLP objective function value of economic profit (*r*).

**Table 2**: Continuous Solutions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Profit* | *Boats -***xb** | *Effort (h) -***xe** | *Time†* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *1* | *2* | *3* | *1* | *2* | *3* | *(secs)* |
| GAMS | 1176.458 | 0.90 | 6.96 | 2.32 | 275 | 160 | 200 | 0.3 |
| GENESYS | 1176.458 | 0.90 | 6.96 | 2.32 | 275 | 160 | 200 | 76 |
| GENOCOP3 | 1176.458 | 0.90 | 6.96 | 2.32 | 275 | 160 | 200 | 19 |
| FORTGA | 1176.444 | 0.90 | 6.95 | 2.34 | 275 | 160 | 199.22 | 268 |
| SGA | 1176.441 | 0.90 | 6.93 | 2.34 | 275 | 160 | 199.56 | 885 |

**Table 3**: Integer Solutions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Profit* | *Boats -***xb** | *Effort (h) -***xe** | *Time†* |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *1* | *2* | *3* | *1* | *2* | *3* | *(secs)* |
| GAMS | 1171.698 | 1 | 7 | 2 | 275 | 160 | 200 | \* |
| GENESYS | 1175.799 | 1 | 7 | 3 | 248.25 | 159.77 | 155.14 | 22 |
| GENOCOP3 | 1175.799 | 1 | 7 | 3 | 248.40 | 159.73 | 155.21 | 14 |
| FORTGA | 1175.799 | 1 | 7 | 3 | 248.14 | 159.77 | 155.14 | 67 |
| SGA | 1175.797 | 1 | 7 | 3 | 249.22 | 159.69 | 155.18 | 288 |

\* - This was an approximated solution from the continuous solution.

*†* - Time in CPU seconds until best solution.

In all the genetic algorithms implemented (tables 2 and 3), convergence to the optimal solution for the model is very close to full achievement. It is clear in these results that simply approximating an integer solution to the model from the optimal continuous solution does not give the optimal integer solution. As expected, it is also noticeable that the GA with integer restrictions achieves the optimal solution far quicker than the continuous cases. The differences in solution times between the GA solvers (for exactly the same model structure) is significant. This highlights the point that different GA implementations are more advanced and may be more suitable for some model types than others. Also, parameter options and values have a significant effect.

It should be noted that for the GAMS model, a starting point of the origin (i.e. all variables zero) is reported to be locally optimal thus stopping optimisation. The solver advises at this point to use a better starting solution. Any non-zero initial solution for **xb** and **xe** is satisfactory.

The typical convergence rates are shown between two selection strategies (i.e. tournament selection and roulette wheel selection) in FORTGA for the integer case. Three runs were performed with each selection method over 10,000 generations and the best run plotted in figure 1. In all runs the tournament selection performed best, achieving the best solution of 1176.444. Roulette wheel selection was not as successful, converging to a best value of 1170.760 in the 10,000 generations completed. For this model, it appears that the convergence is generally slower using the roulette wheel selection than tournament selection.

In the genetic algorithms, a maximum generation count was used as the stopping criteria, and determined by performing a number of optimisations until a satisfactory level was ascertained. Other parameters were also evaluated through a number of trial runs, although default parameter values were preferred where possible.