EVOLUTIONARY ALGORITHMS FOR FIRE AND RESCUE SERVICE DECISION MAKING

Dr. Alastair Clarke, Prof. John Miles and Prof. Yacine Rezgui
Cardiff School of Engineering, Cardiff, UK

ABSTRACT

Determining high performance strategies for the long term use and deployment of Fire Brigade resources in order to minimise the risk of loss of life and property is a highly complex problem. Software is available to allow fire Brigades to investigate the effectiveness of different strategies and deployments of resources (e.g. station and appliance location, staffing levels etc). This software (FSEC – Fire Service Emergency Cover toolkit) contains the information and procedures needed to define the risks, operation and geographical relationships for a given Fire Brigade, but only allows the evaluation of one solution at a time with no provision for searching for high performance solutions. To manually evaluate all potential solutions (totalling around 10^{55} for a typical brigade) is impossible in terms of time, even if it only took a few seconds to evaluate each solution.

This paper describes the development of a method to locate high performance solutions for the deployment of Fire Brigade resources, using Evolutionary Algorithms in conjunction with the FSEC Toolkit. Such algorithms allow the relatively rapid identification of areas of good potential solutions by sampling only a small percentage of the total search space. In order to achieve this, the FSEC software is being re-written in a more computationally efficient manner. This will then be coupled with an Evolutionary Algorithm in order to identify suitable solutions to the problem.

INTRODUCTION

In 1995, the Audit Commission report, “In the Line of Fire” [1] recommended that the Fire and Rescue Service should look at measures for preventing fires as well as response to incidents, with less emphasis on the value of property and more on the saving of lives. Following this it was suggested [2] that a new risk-based approach be investigated which additionally included the work of the fire service in dealing with incidents such as road traffic accidents (so called special services). The result of this was the development of a new risk-based assessment toolkit, which forms the basis of the FSEC software currently used by brigades to assess the effectiveness of their resource allocation and response strategies.

However before considering the development of optimisation software based on FSEC, which is the subject of the current research programme, it is first instructive to gauge the size of the problem by estimating the number of possible solutions, and the feasibility (or otherwise) of evaluating these potential solutions using the currently available tools. In this case, the number of potential solutions is given by the number of specific ways in which a Fire and Rescue Service can organise the resources available to them. For example, the South Wales Fire and Rescue Service has 50 fire stations comprising 19 wholetime stations, 5 day-crewed and 26 retained stations, employing approximately 1000 full-time firefighters and 600 retained firefighters.

In using the decision making tools being developed in this project, a brigade may wish to consider the potential to open new stations or move existing ones. Continuing the South Wales example, it is assumed that a further 20 potential sites for new or re-located fire stations have been identified, giving a total of 70 potential locations for active fire stations. Presuming that the brigade wishes to maintain the number of active stations at 50, then the number of ways in which 50 stations may be chosen from 70 sites must be calculated. The total number of combinations, N_s, assuming that order is unimportant (having stations A, B and C open is the same and having stations C, B and A open), is given by

$$N_s = {}_{70}C_{50} = \frac{70!}{50!(70-50)!} \approx 10^{17}$$

At a conservative estimate, each open station could have around 6 different vehicle or staffing combinations. For example, the station may be manned on a wholetime or retained basis, with a range of different vehicle types. Thus, for each selection of fifty station sites, the number of different configurations which may be achieved, N_c, is approximately 6^{50} ($\approx 10^{38}$). Thus the total potential configurations, N, for the South Wales brigade example is given by

$$N = N_s \times N_c = 10^{17} \times 10^{38} = 10^{55}$$

Thus it may be seen that the number of possible solutions for the configuration of a typical fire brigade area is truly massive.

Taking a simpler example, such as a medium-sized city like Cardiff, with 4 fire stations and a further 4 possible station sites, still results in a total number of solutions which is approximately 10^5. Of course, it is likely that there are a number of very similar solutions within these totals, with some solutions which are not practical to implement. Even allowing for these, and given that the FSEC model execution times are substantial (around 27 minutes for a typical brigade), the total number of potential solutions is still far above that which it is possible to evaluate by exhaustive search using the existing software. Therefore the only feasible way to locate good solutions (we prefer not to use the word optimal) is to use some form of search algorithm such as an evolutionary algorithm. These algorithms are very efficient at searching complex, highly constrained and multi-objective search spaces and can find areas of high performance by sampling a very small percentage of the search space.

EVOLUTIONARY ALGORITHMS

Optimization algorithms typically need a fully defined objective function complete with whatever constraints need to be applied. Also, many algorithms have problems with complex search spaces containing many local optima because they become “stuck” on one of many peaks within a complex search space. There is however a class of algorithms which can search with less well defined gradient information than that provided by an objective function by using instead a so called fitness function [3,4]. All that is required to form a fitness function is the ability to judge whether one solution is better than another. These algorithms also use a population of solutions to enable them to sample widely across the search space and hence they are much better at avoiding local optima. The algorithms are known collectively as Evolutionary Algorithms (EA) because they typically employ search techniques which are analogous to natural processes. For example genetic algorithms and genetic programming [3,5] mimic Darwinian evolution and Particle Swarm Analysis [6] mimics the behaviour of a flock of birds. EA have been applied to many complex decision making problems and found to perform well.

As an example of an evolutionary algorithm, the typical architecture of a genetic algorithm is shown in Figure 1. As shown, the process starts with the creation, usually at random, of an initial population, whose size represents only a small fraction of the total number of potential solutions. Each member of the population represents a potential solution to the problem being considered. Searching for a solution(s) using a population means that at each step, a genetic algorithm samples as many points within the search space as there are members of the population (assuming no two members are identical). This is one of the strengths of a genetic algorithm, enabling it to sample widely throughout the search space and identify areas of high performance (i.e. good solutions) on which the search can start to converge. The use of a population enables multiple high performance areas to be identified and helps the genetic algorithm to avoid convergence on local optima.

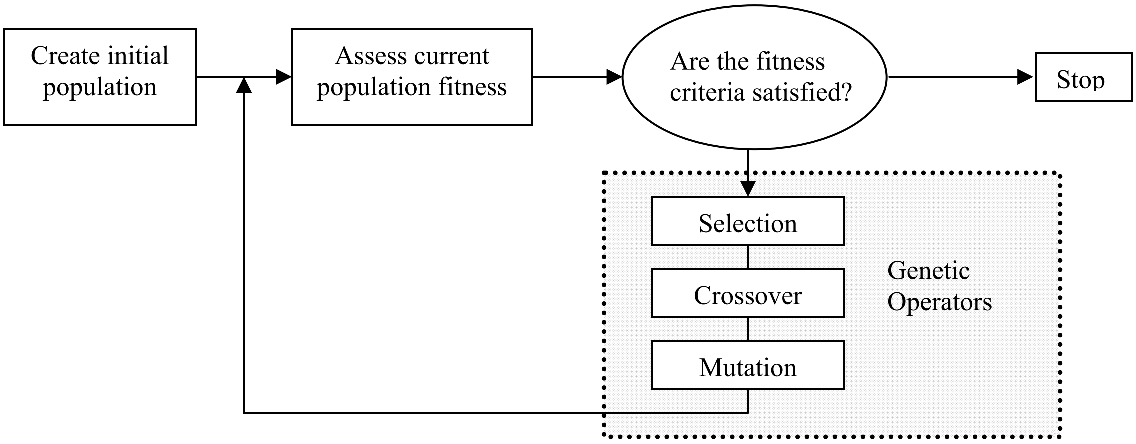

Figure 1: Genetic Algorithm architecture

Once the initial population is established, the basic process of the genetic algorithm is to adapt and modify the members of the population based on feedback relating to how good a solution each member of the population is, until one or more good solutions are found. The judgement of how well each member of the population performs is undertaken by the fitness function. The adaptation and modification is then undertaken by the other processes shown in Figure 1 and there is an iterative procedure, represented by the loop, which continues until some convergence criterion is satisfied. The process is analogous to Darwinian evolution in that there is a population of solutions. These solutions are subject to an environment (the fitness function) which tends to favour the reproduction of the solutions which are best suited to that environment. Hence solutions which suit the defined environment are evolved over a number of iterations (called generations).

Another feature of evolutionary algorithms is their ability to handle constraints and also to deal with conflicting constraints. The way this is typically achieved is to include the constraints in the fitness function and then to penalise those solutions which fail to meet one or more constraints. Also evolutionary algorithms cope well with multi-objective problems [7].

EA are very efficient at searching massive search spaces (e.g. 10^{84} feasible solutions [8]). To do so they only need to look at a very small fraction of the total number of solutions and although they cannot be guaranteed to find the absolute optimum, they are very good at finding areas of high performance within the search space. Each time that an EA examines a solution it runs its fitness function. In this work the fitness function is FSEC so for each EA run it will be necessary to run FSEC several hundred times for each run of the EA. As FSEC execution times are lengthy, this is a significant challenge, which is discussed later in this paper.

Examples of the domains to which EA have been applied include, Engineering Design [3, 9, 10], manufacturing scheduling [11], controlling steel rolling mills [12], the rehabilitation of water networks [8] and the analysis of ultra sound images [13]. All these problems are multi-objective and highly complex with many feasible options. Many contain both continuous and discrete variables, have non-linear search spaces with insufficient information to form an objective function and for all of them EA have been found to perform well.

Given their ability to cope with complex, multi-objective, highly constrained search spaces while avoiding becoming trapped on local optima, EA are potentially an excellent addition to the FSEC toolkit which currently can only be used to evaluate a single strategy in each run. They will allow the user to identify the complete range of high performance strategies and look at the trade offs between the various objectives.

A genetic algorithm has been developed for this project using Fortran 90. The algorithm conforms, in its basic structure, to that shown in Figure 1 and has been tested using a range of test functions where the solution is known. Figure 2 shows the evolution of the maximum population fitness with generation, as a typical optimisation run proceeds.

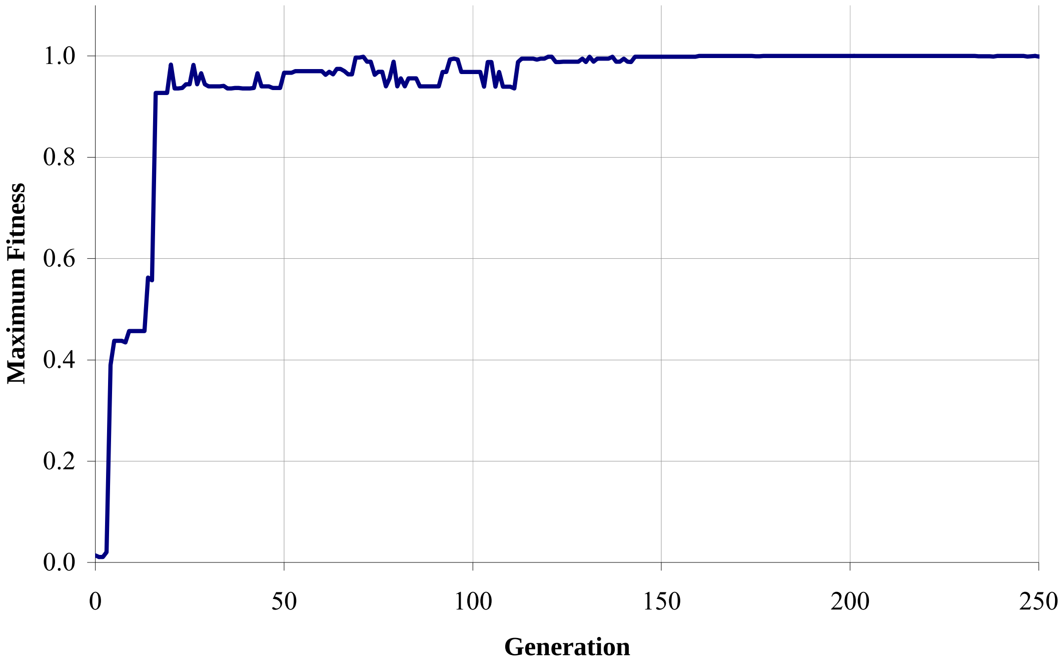

Figure 2: Example of change in maximum fitness with generation

In this case, the algorithm was being tested using sin(*x*) as a very simple fitness function. The problem was to optimize sin(x) where x lies within the range 0,π. The known solution is that sin(*x*) reaches a maximum value of 1 when $x = \pi/2$. Figure 2 shows that the maximum fitness rapidly reaches reasonably good solutions (i.e. where sin(*x*) > 0.9) within approximately 20 generations, and then takes approximately 100 further generations to find the absolute maximum solution to the problem. This is for a case where the answer is known, but for examples where the answer is not known, one would hope that the change in fitness will follow a similar path although to some extent this is a function of the problem space. This algorithm will be further developed for the fire resource optimisation problem, although work to date has concentrated on the development of FSEC for use as a fitness function.

DEVELOPMENT OF FSEC TOOLKIT FOR USE AS A FITNESS FUNCTION

As part of the optimisation algorithm, it in necessary to evaluate the effectiveness of potential solutions generated by the algorithm via the use of a “fitness function” – in this case this function must give a measure of how effective is the emergency response, given a particular configuration of fire service resources.

The software currently used by the Fire Service to investigate response strategies, the Fire Service Emergency Toolkit (FSEC) [14], was introduced to the UK’s Fire and Rescue Services in 2004 following an extensive development programme. FSEC is based on the Wings32 geographical information system, which allows FSEC to model the geographical relationships of a brigade's ground including travel times, census data, information relating to property types, the location of historical incidents and the location of fire stations and their current staffing and facilities. FSEC uses bespoke software to calculate the probable life losses and property damage based on a particular response strategy. The basis of these calculations is a derived relationship between response time and fatality rate for each of the major types of emergency faced by the Fire Service. There are four main types of incident considered by FSEC:

- Dwellings fires
- Special Services (including road traffic accidents)
- Other buildings incidents
- Major incidents (such as terrorist attacks, major chemical incidents etc)

Each of these incidents is considered in a separate software module within FSEC. The first module considers the risk to life caused by fires in the home. In this module, calculations are based on data available for actual dwelling fire incidents, the number of residents and a mathematic relationship between response times and fatality rates. Calculations within all modules of FSEC are based around small geographical areas known as Output Areas. In order to ensure a statistically robust estimate of casualty rates based on incident data, these small Output Areas are grouped into larger areas of similar risk based on socio-demographic data. In the event that it is not possible to form statistically robust areas, FSEC combines the incident data with elements of the Census information (which has been shown to correlate strongly with the rate of fire). Similar grouping procedures in order to achieve statistically robust data are used in each of the four FSEC modules. In addition to the model results, FSEC allows the user to observe the locations of fires from the incident data, in order to identify clusters of fires and appropriately target fire prevention measures.

The Special Services module considers nine types of incident, including road traffic accidents, lock in/out and ladder rescues. The casualty risk for these incidents is calculated based on the historical occurrence of such incidents per square kilometre. Again, hotspots of incidents can be identified within the software.

The third module, Other Buildings, calculates the risks to life and property in types of building other than dwellings. Here, the risk assessment is based on the numbers of each building type within the output area, their individual risk and occupancy type. Research based on national data has calculated the probability of a fire within each building type, and this is used together with data from the Fire Service's fire safety inspections and other sources as appropriate to calculate the risk to both life and property within the individual Output Area.

The Major Incidents module considers the risks from seven types of major incident, including aircraft crashes, major road/rail accidents, bombing and flooding incidents. The risk of each type of incident is calculated using national statistics.

The locations of each fire station and their allocated vehicle and firefighter resources are set up within FSEC, and the software calculates the time taken for each vehicle to arrive at each Output Area within the brigade. This is done using a mathematical model of the road network, and takes into account the turn-out time for a particular vehicle, travel time and the vehicles speed relative to the allowable road speed. From the calculated response times, FSEC uses mathematic relationships between vehicle response time and fatality rate to determine the number of lives lost in dwelling fires, special service incidents and other buildings fires, together with the property damage caused by other buildings fires.

Cost-benefit analyses are then performed which compare the results of the strategy being considered with the base-case results obtained using the current Fire Service configuration. However, FSEC is limited in that it is only possible to evaluate one potential solution at a time, and the evaluation of a range of alternative strategies is time consuming. In addition, it is graphics-based and as such has relatively long execution times. Thus the current FSEC software is not directly suitable for use as a fitness function, even taking into account the fact that an evolutionary algorithm can identify near-optimal solutions to a problem by only sampling a small percentage of the total number of solutions.

Therefore, the core of FSEC has been re-written for this work as a more computationally efficient Fortran-based software code, which is suitable for use with an evolutionary algorithm. The strategy employed is explained in Figure 3.

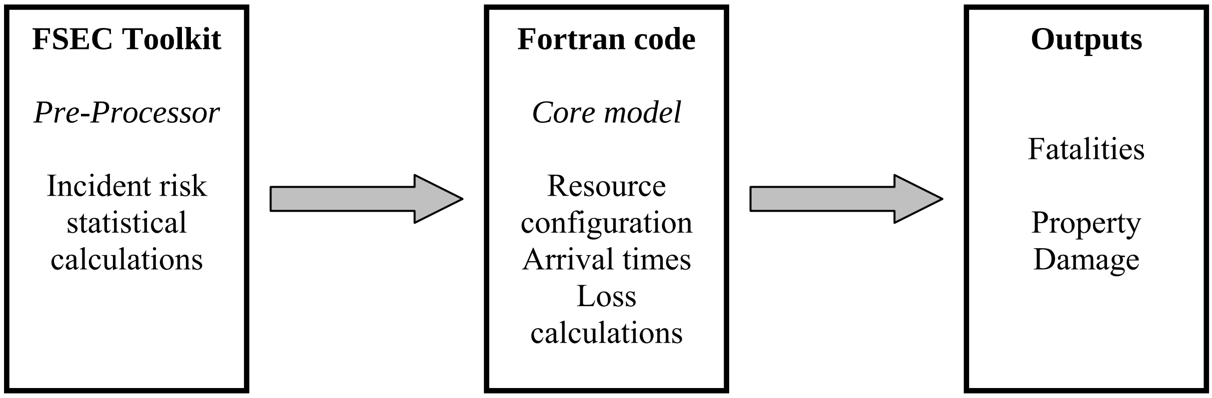

Figure 3: New model strategy

The original FSEC software is used as a "pre-processor" to perform the statistical calculations necessary in order to provide robust incident rates based on incident data. Since these calculations are not influenced by the configuration of fire service resources, it is only necessary to perform these pre-processing operations once for a particular run of the optimisation routine. Once the pre-processing is complete, the data is transferred to the core model, written specifically for this project in Fortran 90. The core model is based on the same methods and mathematic relationships as FSEC uses, but has been re-written in order to achieve sufficiently short execution times to allow its repeated use as a fitness function within an optimisation routine. The code examines the current station and vehicle configuration strategy, and calculates arrival times using basic road network information produced during the pre-processing phase together with individual vehicle details. From this, a mathematical relationship is used to calculate the likely fatalities and property loss levels based on the calculated vehicle arrival times. All calculations which depend on the particular resource / fire station configuration being examined are contained within the core model.

The execution times for the improved Fortran model are significantly reduced when compared to the original FSEC times. For example, a typical brigade FSEC model takes approximately 27.2 minutes to execute on a 2.8GHz Pentium 4 PC. The Fortran core model takes around 18 seconds to execute for the same data-set. Thus, the savings in computer time when evaluating a range of configurations within an optimisation routine can be seen. For example, to evaluate 500 different station or vehicle configurations within the original FSEC software would take some 226.7 hours of CPU time, plus the time taken to set-up each configuration within the software. In contrast, to evaluate the same 500 configurations using the new model would require one run of the original FSEC software (as a pre-processor) followed by 500 runs of the Fortran code, giving a total of 2.95 hours – representing a 98.7% time saving.

DISCUSSION AND CONCLUSIONS

Work is currently on-going to continue the development of the FSEC-based optimisation routine described in this paper. Following final validation of the Fortran "core" model, it will be coupled to the evolutionary algorithm, as described, and used to solve a range of Fire and Rescue Service optimisation problems in order to demonstrate the suitability of the technology to discover high

performance configurations for the Fire Service resources. Discussions are currently ongoing with a number of UK Fire and Rescue Services regarding the use of real FSEC brigade data to test the optimisation algorithm.
Conclusions which may be drawn at this stage of the work are:

- The problem of optimising the configuration of Fire Service resources is a highly complex problem
- There is a massive number of potential solutions
- It is not possible to evaluate all of these solutions in order to find high performance ones
- Evolutionary algorithms offer many advantages in dealing with huge, highly complex problems such as this
- A more computationally efficient version of the core features of FSEC has been developed in Fortran, and has suitable execution times to enable its use as the fitness function for an evolutionary algorithm
- Work is ongoing to couple the core FSEC model with a genetic algorithm

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of EPSRC for this work under grant EP/G026262/1, together with the support and assistance of the Fire and Resilience Directorate of the Department for Communities and Local Government, and the various Fire and Rescue Services who are co-operating with the research.

REFERENCES

1. Audit Commission, “In the Line of Fire”, The Stationary Office, London, 1995.
2. Fire and Emergency Planning Directorate, “Out of the Line of Fire”, Home Office, London, 1998.
3. Goldberg D.E., “Genetic algorithms in search, optimisation and machine learning”, Addison-Wesley, New York, 1989.
4. Parmee, I., “Evolutionary and adaptive computing in Engineering design”, Springer-Verlag, Berlin, 2001.
5. Koza J.R., “Genetic Programming: On the programming of computers by means of natural selection”, MIT Press, Cambridge MA, 1992.
6. Kennedy J & Eberhardt R, “Particle Swarm Optimization”, Proc of IEEE Int Conf on Neural Networks, Piscataway NJ, 1995, pp 1942-1948.
7. Coello Coello, C., Van Veldhuizen, D., & Lamont, D., “Evolutionary Algorithms for Solving Multi-objective Problems”, Kluwer Academic, New York, 2002.
8. Kapelan, Z, “Some recent developments in water distribution systems rehabilitation”, Parmee I (ed) ACDM2008, 2008.
9. Miles, J.C., Sisk, G., & Moore C.J., “The conceptual design of commercial buildings using a genetic algorithm”, Computers & Structures, 79, 2001, pp 1583-1592.
10. Shaw D., Miles J. & Gray A, “Determining the Structural Layout of Orthogonal Framed Buildings”, Computers & Structures, 86, 2008, pp 1856-1864.
11. Coello Coello C., Rivera D. & Cortes N., “Job shop selection using the clonal selection principle”, Parmee I (ed), ACDM VI, 2004, pp 113-124.
12. Stalker, I.D. & Bates, R.A., “Informing the design of special section roll sets, in Parmee”, I.(ed) ACDM VII, 2006, pp 81-90.
13. Bovenkamp, E., Dijkstra, J., Bosch, J & Reiber, J, “Multi-agent segmentation of IVUS images”, Pattern Recognition, 37(4), 2004, pp 647-663.
14. Fire Research Report 01/2008, “Fire Service Emergency Cover Toolkit: Executive Summary”, Department for Communities and Local Government, May 2008.