

Strategic Fire and Rescue Service decision making using evolutionary algorithms

A. Clarke*, J.C. Miles

School of Engineering, Cardiff University, UK

ARTICLE INFO

Article history:

Available online 28 April 2012

Keywords:

Evolutionary algorithms
Optimisation
Decision making
Fire
Emergency services
Genetic algorithms

ABSTRACT

This paper describes the development of a novel, risk based method to locate high performance solutions for the deployment of Fire and Rescue Service (FRS) resources, such as fire stations and appliances, using evolutionary algorithms in conjunction with Fire Service Cover Models. Such algorithms allow the relatively rapid identification of areas of good potential solutions by sampling only a small percentage of the total search space. A real example of the use of the software to optimise vehicle locations is presented which identifies significant potential increase in efficiency and effectiveness over the existing vehicle locations.

© 2012 Civil-Comp Ltd and Elsevier Ltd. All rights reserved.

1. Introduction

Fire and Rescue Services within the UK have been extensively modernised since the 1995 Audit Commission report, “In the Line of Fire” [1] which recommended that the Fire and Rescue Service should place priority on preventing fires as well as response to incidents, with less emphasis on the value of property and more on the saving of lives. A new risk-based approach was instigated [2] which additionally included the work of the Fire Service in dealing with incidents such as road traffic accidents, flooding and other emergency incidents (so called special services). The result of this was the development of a new risk-based assessment methodology, which forms the basis of the Fire Service Emergency Cover (FSEC) software currently used by FRS’s to assess the effectiveness of their resource allocation and response strategies.

Determining high performance strategies for the long term use and deployment of FRS resources in order to minimise the risk of loss of life and property is a highly complex problem. Resource deployment strategies are currently determined using a combination of professional experience, FRS practice and modelling. The FSEC model allows FRS’s to investigate the effectiveness of different strategies and deployments of resources (e.g. station and appliance location, staffing levels, etc.). This software contains the information and procedures needed to define the risks, operation and geographical relationships for a given FRS, but only allows the evaluation of one solution at a time with no provision for searching for high performance solutions.

2. Problem scale

In order to consider the necessity of using optimisation methods to solve the problem, it is 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 (UK) Fire and Rescue Service has 50 fire stations comprising 20 wholetime crewed stations, 2 day-crewed (continuously staffed during daytime, and staffed by fire-fighters who reside near the station and who are summoned in an emergency via pagers) and 28 retained stations (staffed by fire-fighters who live or work nearby and who are alerted via pagers), employing approximately 1000 full-time fire-fighters and 600 retained firefighters.

In using the decision making tools being developed in this project, a FRS may wish to consider the potential to open new stations or move existing ones. Continuing the hypothetical 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. Keeping the number of active stations at 50, then the number of ways in which 50 stations may be chosen from 70 possible 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.

$$N_s = {}_{70}C_{50} = \frac{70!}{50! \times 20!} \approx 10^{17} \quad (1)$$

At a very conservative estimate, each open station could have around six different vehicle or staffing combinations. For example,

* Corresponding author.

E-mail address: clarkea7@cf.ac.uk (A. Clarke).

the station may be manned on a wholetime or retained basis, with a range of different vehicle types. Thus, for each selection of 50 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 FRS example is given by.

$$N = N_s \times N_c = 10^{17} \times 10^{38} = 10^{55} \quad (2)$$

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

Taking a simpler example, such as a medium-sized city like Cardiff, with four fire stations and a further four 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 min for a typical FRS), the total number of potential solutions is still far above that which it is possible to evaluate by exhaustive search using the existing software. Even if (as in the case study presented here) Fire Services only wished to optimise vehicle locations within a fixed station network, the problem is still prohibitively large. Therefore the only feasible way to locate good solutions 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.

3. Evolutionary algorithms

Optimisation 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 (EAs) 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 Fig. 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.

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 Fig. 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 based on 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],

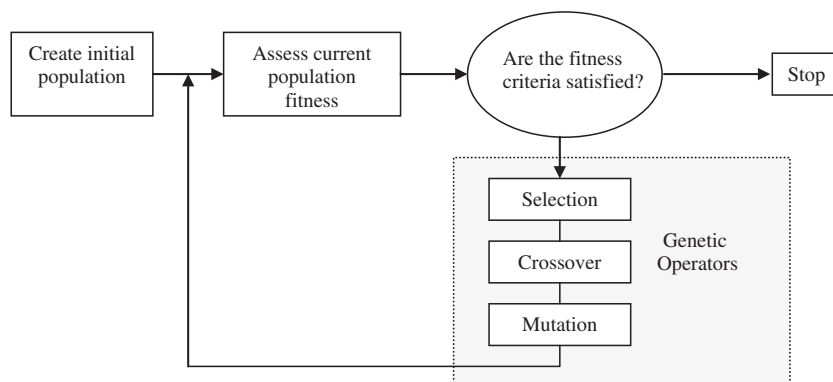


Fig. 1. Genetic algorithm architecture.

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.

4. A fitness function based on the FSEC methodology

As part of the optimisation algorithm, it is 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. This is based on the methodology used within the FSEC toolkit to calculate life and property losses based on Fire Service response, whilst maintaining appropriate levels of firefighter safety.

4.1. FSEC methodology

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 Fire Service area 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. The FSEC model is shown schematically in Fig. 2.

Calculations within FSEC are based around small geographical regions (of a few hundred homes for example) called Output Areas. Within each output areas, four main types of incident are considered:

- Dwellings fires (in the home).
- Special Services (including road traffic accidents).
- Other buildings incidents (i.e. in non-domestic buildings).
- Major incidents (such as terrorist attacks and major chemical incidents).

Each of these incidents is considered in a separate software module within FSEC. Full details are given by Wright et al. [15] but the calculation procedure for the Dwellings module is outlined here for completeness. The other modules adopt similar calculation procedures.

The FSEC Dwellings module predicts fatalities caused by fires in the home, and is based on statistical incident data from the Fire and Rescue Service. The calculations are performed on small geographical areas known as “Output Areas”. These will typically, for residential areas, have around 200 persons living in them. Based on four risk factors derived from census data (the percentage of lone pensioners, percentage of rented accommodation, the number of single parent families, and the percentage of the population suffering from a long terms illness), these are grouped into larger “Risk Areas” so that the data used for predicting Dwelling fire casualty rates is statistically significant. These are adjacent output areas are deemed to have similar levels of fire risk. Once the risk areas have been formed, they are further merged into larger Risk Groups. These are not necessarily geographically adjacent, but ensure that the dataset used to calculate the dwelling fire frequencies is large enough to be statistically robust and allow real fire statistics to be used to predict the fire frequency. This is measured using a robustness calculation – in the event that robust risk groups cannot be formed, the software uses a regression formula based on census data to predict the fire frequency. The rate of dwelling casualty per year, predicted by this process, is then divided between the output areas based on output area population, to give a predicted number of dwelling casualties per annum within each output area.

Each risk area (and, by extension, output area) will have been assigned a Planning Scenario by the Fire Service which specifies the vehicles required to attend an incident, e.g. two pumps. From this planning scenario, the arrival times of vehicles which reach the output area and meet the planning scenario can be determined. Calculations are performed within six time periods during the day, since not all vehicles are available or are manned in the same way throughout a particular 24 h period.

The number of casualties which become fatalities is calculated using the arrival-time based fatality rates shown in Table 1.

For example, if a vehicle arrives at an incident in a time between 5 and 9.99 min, that vehicle has an associated fatality rate, F_i , of 0.02596 fatalities per dwelling fire casualty. However, in order to simulate the advantageous effects of multiple vehicles arriving at the scene according to the planning scenario, a partial benefit factor is additionally applied to the arriving vehicle. These partial benefit factors are shown in Table 2 for up to four vehicles (depending on the planning scenario).

Thus for a given number of dwelling casualties (P_{dwel}) calculated from historical data as previously described, in a given year the predicted number of fatalities is given by.

$$Fatalities = P_{dwel}[(PB_1 \times F_1) + (PB_2 \times F_2) + (PB_3 \times F_3) + (PB_4 \times F_4)] \quad (3)$$

Depending on the number of vehicles in the planning scenario, and their arrival times, appropriate Fatality Rates and Partial Benefit factors are selected from Tables 1 and 2 and used in Eq. (3) to predict the number of fatalities per year.

Special Services covers non-fire-fighting activities where there is still a risk to life, such as road traffic accidents. The Special Services module considers nine types of incident, including road

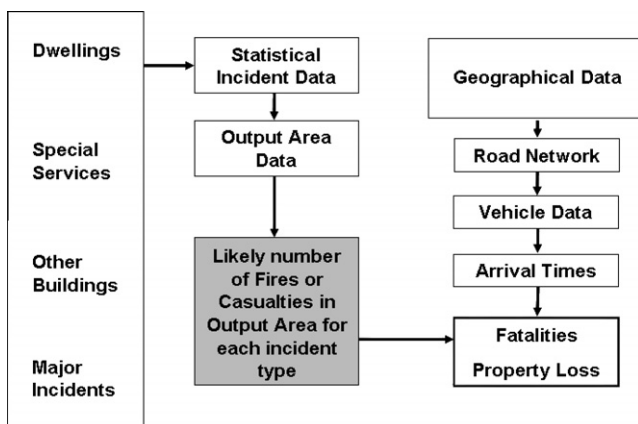


Fig. 2. FSEC toolkit model schematic.

Table 1
Fatality rates per dwelling fire casualty.

Vehicle arrival time (min)	Fatality rate F_i
$t \leq 5$	0.02037
$5 < t \leq 10$	0.02596
$10 < t \leq 15$	0.04227
$15 < t \leq 20$	0.04787
$t > 20$	0.08725

Table 2
Partial benefit factors.

Partial benefit factor	Number of vehicles in planning scenario			
	4	3	2	1
PB1	0.72	0.72	0.72	1.00
PB2	0.25	0.25	0.28	0.00
PB3	0.015	0.03	0.00	0.00
PB4	0.015	0.00	0.00	0.00

traffic accidents, lift and line 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 calculation procedure is the same as for Dwellings, with fatality rates calculated separately for each of the nine types of Special Service incident.

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. These calculations are performed for 17 major sub-types of other building, such as hospitals, hotels and schools.

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 result of the initial calculations of each module is a predicted rate of fires and casualties for each output area in each of the four incident categories (and sub-categories). FSEC then calculates the effect of a Fire Service response to incidents on saving lives (i.e. preventing casualties becoming fatalities) and preventing property damage.

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 FRS. 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. The road network is represented as a series of nodes (road junctions) and links (roads) within a Geographical Information System. This software uses a proprietary road-spanning algorithm which calculates travel time from each fire station location to every node in the road network assuming that the vehicle is travelling at the permitted road speed. This travel time is then modified using the percentage of road speed at which each vehicle can travel – for example larger aerial platform vehicles travel far more slowly than smaller pump vehicles. To this modified travel time is added the vehicle's turn-out time (the time taken from the alarm being raised to a vehicle leaving the fire

station) to give an overall arrival time for each vehicle. From this data, FSEC identifies the order in which vehicles arrive at each output area (each output area has a centroidal road junction allocated to it).

From the calculated response times, FSEC uses the previously described mathematical 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 can then be performed which compare the results of the strategy being considered with the base-case results obtained using the current Fire Service configuration. These taken into account a range of factors, such as the cost of making changes to Fire Service cover (additional vehicles, re-organisation costs, etc.) and the ongoing staffing cost implications, and compare these to the financial benefit to society over the remainder of a fire victim's life, had they not died in a fire. This process uses data and values supplied by the UK government, and is explained in more detail in Gros et al. [16].

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.

4.2. Fitness function development

The core of FSEC has been re-written for this work as a more computationally efficient Fortran 90 software code, which is suitable for use with an evolutionary algorithm. The strategy employed is explained in Fig. 3.

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 dwellings, special services, other buildings and road network data is transferred to the core model, written specifically for this project in Fortran 90. The Other Buildings data requires further pre-processing as it is exported from FSEC for each building. A Fortran 90 code has been written to perform some pre-processing on this data to aggregate it to the Output Area level to reduce the necessary calculations.

The core model is based on the same methods and mathematical 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. The use of road network data exported from FSEC avoids the need to repeat road network calculations at each run of the fitness function which is time consuming. From this, the FSEC mathematical relationships are used to calculate the predicted fatalities and property loss levels based on the calculated vehicle arrival times at each output area. All calculations which depend on the particular resource/fire station configuration being examined are contained within the core model.

Finally, the cost-benefit analysis (based on procedures already developed by the Department of Communities and Local Government) is performed to determine the “fitness” of the particular resource configuration being considered.

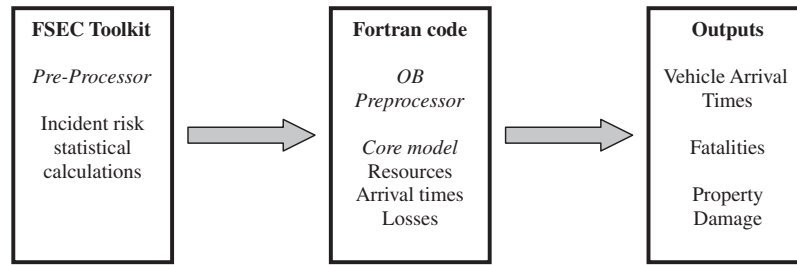


Fig. 3. New model strategy.

The execution times for the improved Fortran model are significantly reduced when compared to the original FSEC times, given the careful attention paid to code optimisation. For example, a typical FRS FSEC model takes approximately 27.2 min to execute when tested on a 2.8 GHz Pentium 4 PC. The Fortran core model takes around 2 s 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.

In order to validate the software, comparisons were made between the results of a typical analysis of a Fire Service's data using the original FSEC software and the new Fortran model. Results were compared across all the modules of the software, and the Dwellings results are examined here as an example. The calculated fatalities were compared at output area level and found to be in good agreement. The percentage error between Fortran and original FSEC results was calculated for each output area. For the data set considered, which contains 7569 Output Areas, the error values were found to have a mean of 0.002% and a standard deviation of 0.104%. Table 3 shows the error distribution for the output areas.

Inspection of Table 3 shows that some 99.6% of output areas had errors of less than 2% between the Fortran and original models. The maximum error is between 8% and 10%. Investigation of the output areas concerned showed that this error was related to different rounding levels within the two software packages causing high percentage errors where the predicted fatalities are very small. The authors believe the Fortran-derived values to be the more accurate estimates of the fatalities in this case.

In order to make a further comparison, the total fatalities for the Fire Service area under consideration were calculated using both models. FSEC reported a total of 7.9070 fatalities in dwelling fires per year in the validation data set, whereas the Fortran Model reported fatalities of 7.9068, which demonstrates the high level of agreement between the two calculations.

4.3. Integration with genetic algorithm

Fig. 4 shows the integration of the FSEC-based model with the genetic algorithm, which has been coded by the authors from first principles in Fortran. The Fortran 90 code forms the fitness function of the algorithm. The genetic algorithm is broadly canonical in structure, following the basic structure shown in Fig. 1, although it employs integer encoding. This leads to a number of advantages

Table 3
Percentage error distribution for dwelling fatalities.

>	≤	No. of output areas	% of output areas
0	2	7540	99.62
2	4	27	0.36
4	6	0	0
6	8	0	0
8	10	2	0.03
10	100	0	0

– not least the simplicity and ease of interpretation of the solution. Encoding is discussed further in the case study in Section 5.

It may be seen from Fig. 4 that the original FSEC is run once as a pre-processor, followed by the additional Other Buildings pre-processing as described in Section 4.2. This non-configuration-specific data is then input to the GA loop. An initial random population of solutions is then selected, and their fitness evaluated. Further generations of solutions are then bred from the better solutions of the previous generation, and the loop repeats until near-optimal resource configurations have been identified.

The GA has been developed so that various parameters can be selected for optimisation. For example, an FRS may wish to optimise the crewing and vehicle allocations for a fixed set of fire station locations. Alternatively, the optimisation may consider a complete optimisation where fire station locations are allowed to change in addition. For example, in the case study presented here, the dwellings and travel network parts of the Fortran FSEC-based model were used.

5. Case study

As an example of the use of the optimisation methods developed in this paper, a case study is now presented which details the initial part of an ongoing investigation into Fire Service resource optimisation being conducted by the lead author in conjunction with South Wales Fire and Rescue Service (SWFRS). South Wales is a large FRS with 50 fire stations as outlined in Section 3.

The purpose of the initial project carried out with SWFRS was to investigate the appropriate location of vehicles to optimise the response to dwelling fires. These are a significant part of the Fire Service's work. The current SWFRS response to a dwelling fire is to send two pump vehicles (a “standard” fire vehicle), and thus this case study considers the appropriate locations for pump vehicles. It was decided to keep the fire station locations fixed as per the

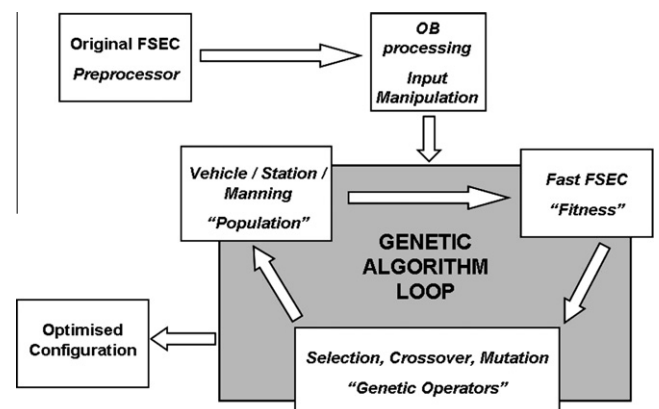


Fig. 4. FSEC model and optimisation routine integration.

current fire station network of South Wales, and only allow the optimisation algorithm to modify the location of vehicles within the fixed station network model.

In the current configuration, each of the 50 fire stations has at least one pump allocated and 14 fire stations also have an additional second pump allocated to them, giving a total of 64 pump vehicles within SWFRS. It is essential to ensure that vehicles are located so that they respond to an emergency call in such a way that the risk to both public and fire-fighters is minimised. SWFRS consider that the time between the first and second vehicles arriving at an incident is highly important for both public and firefighter safety. The arrival of a second vehicle at a fire incident means that there are more fire-fighters available to rescue casualties from the fire, thus improving public safety. In addition, fire-fighter safety is improved by having backup fire-fighters available to support those entering a burning building. Thus, in this optimisation, each fire station was assumed to have one pump allocated to it, and these were fixed and not part of the optimisation. The optimisation algorithm was used to modify the location of the 14 additional pumps (which could be allocated to any of the 50 fire stations) in order to minimise the time lag between the first and second vehicles arriving at an incident. In addition, cases with less than 14 additional pumps were investigated, and optimum locations of the available number of additional pumps determined.

In order to calculate the average time lag across the SWFRS area, parts of the FSEC-based Fortran core model were used as a fitness function (specifically the Travel Time and Dwellings modules). The calculation was based on the FSEC output areas. The travel time calculation was used to estimate the arrival time of the first two pump vehicles to each output area. The difference between these two arrival times was used to calculate the time lag for an individual output area, t_i . The casualty rate (c_i) in each output area per annum (statistics used within the FSEC calculations) was then used as a weighting factor to calculate the overall average time lag t , as shown in Eq. (4), for a total of a output areas:

$$t = \frac{\sum_{i=1}^a c_i t_i}{\sum_{i=1}^a c_i} \quad (4)$$

The use of casualty rates to weight the calculation of average time lag is important for two reasons. From a public safety point of view it ensures that the optimisation algorithm directs resources to those areas where, based on Fire Service incident statistics, there is a higher probability of fires occurring. Considering fire-fighter safety, it also ensures that the optimisation gives most improvement in lag times in those areas where fire-fighters respond to more emergencies.

In order to formulate a fitness function for the optimisation algorithm, a special case was simulated where only the fixed 50 pump vehicles (one per station) were used, giving a “base case” time lag, t_0 . The fitness function f (Eq. (4)) was therefore given by the improvement over the base case when up to 14 additional pumps were used:

$$f = \left(t_0 - \frac{\sum_{i=1}^a c_i t_i}{\sum_{i=1}^a c_i} \right)^3 \quad (5)$$

The fitness function uses the cube of the time lag in order to more widely differentiate between solutions which are reasonably closely spaced in terms of time lag. The aim of the optimisation algorithm is to maximise the value of Eq. (5).

Considering the general case of there being n additional pump vehicles (where $1 \leq n \leq 14$ was considered in this case study), integer encoding was used to represent the station to which each of the n pump vehicles was allocated. For example, if the optimisation was being run for $n = 5$, then a typical member of the population would be: [01, 42, 31, 29, 13] meaning that the five pumps

were allocated to stations 1, 42, 31, 29 and 13, where the stations are numbered from 00 to 49. There are a number of advantages to integer encoding, principally that it avoids Hamming cliff problems [4] and is much more simple to understand in the context of this problem than binary encoding.

Thus, the optimisation code proceeds along typical GA lines as described earlier. For this particular problem, three genetic operators were used – single point crossover, creep mutation (where the value of a gene is varied by no more than 5%) and jump mutation (where the value of a gene may be varied to any value within its valid range). Mutation probabilities of 0.015 were found to give rapid convergence of the algorithm. Bradshaw [17] gives a full discussion of these operators and extensive trials of a range of operators. The authors have based their selection of appropriate operators on Bradshaw’s work, to which the reader is referred for further elucidation.

The single-point cross-over operator does not produce invalid vehicle locations, since the use of station numbers 00–49 precludes this. However, it is possible for duplicate vehicle locations to be produced – for example more than one second pump being allocated to the same station. However, when one considers the practicalities of the problem being optimised, it is clear that a solution where the second pumps are located at the same points will be a poorer-performing solution than one where the second pumps are more widely spread around the area. Thus, these duplicate locations are not automatically removed, but are naturally rejected by the optimisation algorithm due to their poor performance and hence low fitness.

A population size of 200 was used for the case where $n = 14$, with proportional roulette wheel selection being employed to select the mating pool for the new population. Elitist selection of the top 10% of solutions was used, with these solutions being automatically carried over to the next generation. Smaller populations were possible as n was decreased, hence giving a smaller search space.

For each number of additional vehicles, the algorithm was run multiple times (typically 10 runs) to ensure that the true optimal area of the search space was being consistently found. Fig. 5 shows the results for the case where $n = 14$. In each run, the fitness of the best individual within the population was logged. In Fig. 5, the average of the best individual fitness values across the 10 runs is plotted, together with the single run which resulted in the highest fitness.

The population is clearly evolving to a more optimal area of the search space, as shown by the increasing average and best individual run fitness levels. Although the trends are somewhat stepped, this is actually a function of the way in which genetic algorithms search for solutions using mutation and crossover to explore new

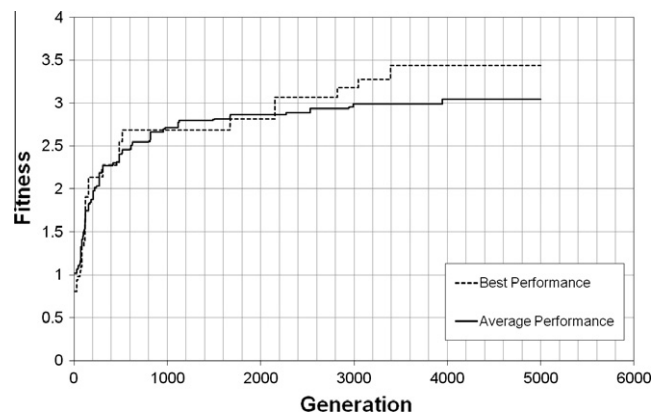


Fig. 5. Fitness evolution for $n = 14$.

regions of the search space. For the largest problem considered here, with $n = 14$, the algorithm takes (averaged across ten runs) 2.53 min for 5000 generations, run on a standard Pentium IV pc with 4 GB of memory, using a Linux operating system. This is clearly a vast improvement over attempting to determine optimum locations for vehicles using the standard FSEC software!

Having run all cases for $1 \leq n \leq 14$ it is possible to compare the performance of the optimised solutions, as shown in Fig. 6. This shows the improvement in time lag over the 50-vehicle base case achieved by the optimal solution for each number of additional vehicles.

The horizontal line in Fig. 6 shows the improvement over the base case achieved by the current locations of the 14 additional

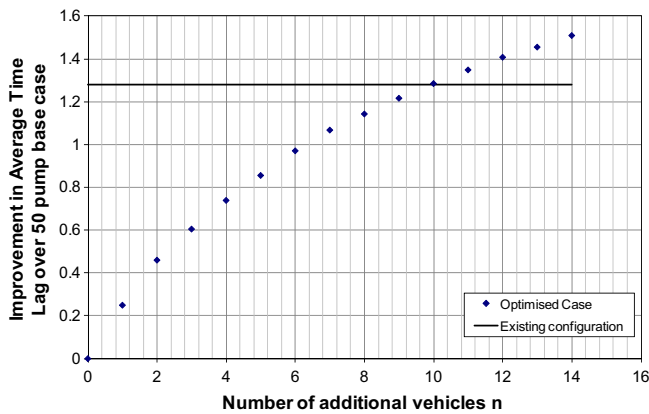


Fig. 6. Improvement in time lag over base case for $1 \leq n \leq 14$.

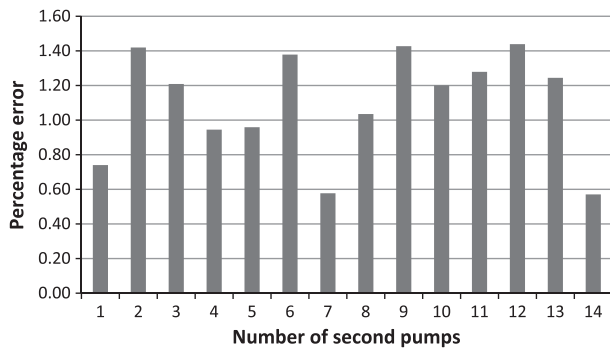


Fig. 7. Errors between FSEC software and optimisation algorithm.

pumps currently used by SWFRS. Thus it may be seen that, by using the results of the optimisation algorithm and relocating the 14 additional pumps currently in use, the time lag may be reduced by approximately 0.2 min. Alternatively, the current level of performance can be achieved by using 10 pumps in optimal locations, offering the potential to release resources to meet other operational needs of the Fire Service. Of course, it is essential to consider the cost implications of any changed to the current FRS configuration.

For each case, as well as the most optimal solution (shown in Fig. 6), the other solutions identified by the algorithm which performed within 5% of the most optimal solution were stored and made available to the Fire Service. This is because these solutions would deliver near-optimal results but may, for operational reasons, be simpler to implement.

In order to validate the results, the optimal configurations of second pumps was re-constructed in the original FSEC software and analysed. The error between the average weighted lag time calculated using FSEC results and that obtained using the optimisation algorithm was found to be minor. The error for each case is shown in Fig. 7.

Table 4 shows the optimal stations at which to locate each of the n pump vehicles, for each value of n which was investigated. It also shows the current locations of the 14 existing additional pump vehicles. It should be noted that these fire station numbers are not the SWFRS station numbers, but are coding numbers for each fire station within the genetic algorithm.

It is pleasing to note that the results build on one another, i.e. the optimal locations for ten vehicles is the same as the optimal locations for nine vehicles plus one additional station. It is interesting that the current location of the 14 additional vehicles used by SWFRS contains eight vehicles which are already at optimal locations. This explains the reasonably high performance of the existing vehicle locations.

6. Discussion and conclusions

A validated model (based in part on the calculation methodology used in existing software) has been developed by the authors to rapidly evaluate the effectiveness of different Fire Service configurations of vehicles, stations, manning levels and response types. This has been coupled to a genetic algorithm (also developed from first principles by the authors) to produce flexible optimisation software which allows Fire and Rescue Services to investigate high performance strategies for using their available resources. The software has been demonstrated on a real optimisation problem in conjunction with a UK Fire and Rescue Service. Although the case

Table 4
Optimised vehicle locations.

n	Station number													
14	19	24	22	00	38	16	48	14	49	42	18	33	01	13
13	19	24	22	00	38	16	48	14	49	42	18	33	01	
12	19	24	22	00	38	16	48	14	49	42	18	33		
11	19	24	22	00	38	16	48	14	49	42	18			
10	19	24	22	00	38	16	48	14	49	42				
9	19	24	22	00	38	16	48	14	49					
8	19	24	22	00	38	16	48	14						
7	19	24	22	00	38	16	48							
6	19	24	22	00	38	16								
5	19	24	22	00	38									
4	19	24	22	00										
3	19	24	22											
2	19	24												
1	19													
Current	19	24	22	00	38	16	48	14	39	09	07	27	30	41

study shown here has only considered response to dwelling fires, further work will apply the software to a wider range of incidents, and to also consider high performance strategies for locations of specialist Fire Service vehicles.

The work is novel in that it provides the UK Fire and Rescue Services, for the first time, with an optimisation tool which is developed to a stage where it can be used to solve practical, real-world problems. This capability has not previously been available to the Fire Services and, as such, this software represents a useful and potentially life-saving application of optimisation methods.

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 with 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 novel optimisation tool for Fire Services has been developed which couples a genetic algorithm with a Fire Cover model.
- A practical optimisation problem has been solved as part of an ongoing collaboration with South Wales Fire and Rescue Service.
- The model uses currently available data sets and does not require significant development on the part of the Fire Service.
- The software performance is such that a complete optimisation can be performed in a similar amount of time to that which would be required to analyse a small number of configurations with other tools currently available to the Fire and Rescue Services.
- This problem has clearly demonstrated the applicability of genetic algorithms to Fire Service optimisation by identifying areas for potential performance increase (by re-allocating current resources) or areas for cost saving whilst maintaining current performance. As such it is a very powerful planning tool for emergency services.

Acknowledgements

The authors wish to acknowledge the support of EPSRC for the initial development of this software 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 within the UK who co-operated with the research. In addition, the first author wishes to thank South Wales Fire and Rescue Service for ongoing support to the development and practical application of this research. In particular he wishes to thank Nigel Marchant and Carol Davies for their assistance.

References

- [1] Audit Commission. In the line of fire. London: The Stationary Office; 1995.
- [2] Fire and Emergency Planning Directorate. Out of the line of fire. London: Home Office; 1998.
- [3] Goldberg DE. Genetic algorithms in search, optimisation and machine learning. New York: Addison-Wesley; 1989.
- [4] Parmee I. Evolutionary and adaptive computing in engineering design. Berlin: Springer-Verlag; 2001.
- [5] Koza JR. Genetic programming: on the programming of computers by means of natural selection. Cambridge, MA: MIT Press; 1992.
- [6] Kennedy J, Eberhardt R. Particle swarm optimization. In: Proc of IEEE int conf on neural networks, Piscataway, NJ; 1995. p. 1942–8.
- [7] Coello Coello C, Van Veldhuizen D, Lamont D. Evolutionary algorithms for solving multi-objective problems. New York: Kluwer Academic; 2002.
- [8] Kapelan Z. Some recent developments in water distribution systems rehabilitation. In: Parmee I, editor. ACDM2008; 2008.
- [9] Miles JC, Sisk G, Moore CJ. The conceptual design of commercial buildings using a genetic algorithm. *Comput Struct* 2001;79:1583–92.
- [10] Shaw D, Miles J, Gray A. Determining the structural layout of orthogonal framed buildings. *Comput Struct* 2008;86:1856–64.
- [11] Coello Coello C, Rivera D, Cortes N. Job shop selection using the clonal selection principle. In: Parmee I, editor. ACDM VI; 2004. p. 113–24.
- [12] Stalker ID, Bates RA. Informing the design of special section roll sets. In: Parmee I, editor. ACDM VII; 2006. p. 81–90.
- [13] Bovenkamp E, Dijkstra J, Bosch J, Reiber J. Multi-agent segmentation of IVUS images. *Pattern Recog* 2004;37(4):647–63.
- [14] Fire Research Report 01/2008. Fire service emergency cover toolkit: executive summary. Department for Communities and Local Government; May 2008.
- [15] Wright M, Antonelli A, Marsden S. Development of the fire service emergency cover planning methodology – final report. Greenstreet Berman Consultants, report C645; 2003.
- [16] Gros S, Spackman M, Carter S. A cost benefit analysis of options to reduce the risk of fire and rescue in areas of new build homes. Department for Communities and Local Government Fire Research, report 1/2010; 2010.
- [17] Bradshaw JP. Deriving strategies for reducing pollution damage using a genetic algorithm. PhD thesis, School of Engineering, University of Wales, College of Cardiff; 1996.