To appear in "Concurrency: Practice and Experience", 1994.

Spatial Interaction Model Optimisation on Parallel Computers

Felicity George, Nicholas Radcliffe, Mark Smith

Edinburgh Parallel Computing Centre, University of Edinburgh,

Kings Buildings, Mayfield Rd.,

Edinburgh EH9 3JZ

fawg@epcc.ed.ac.uk

phone: 031-650-5962, fax: 031-650-6555

Mark Birkin, Martin Clarke

GMAP Ltd, GMAP House,

Cromer Terrace,

Leeds, LS2 9JU

In a collaborative project between GMAP Ltd. and EPCC, an existing heuristic optimisation scheme for strategic resource planning was parallelised to run on the data parallel Connection Machine CM-200. The parallel software was found to run over 2700 times faster than the original workstation software. This has allowed the exploration of complex business planning strategies at a national, rather than regional, level for the first time.

The availability of a very fast evaluation program for planning solutions also enabled an investigation of the use of genetic algorithms in place of GMAP's existing heuristic optimisation scheme. The results of this study show that genetic algorithms can provide better quality solutions in terms of both predicted profit from the solution, and spatial diversity to provide a range of possible solutions.

This paper discusses both the parallelisation of the original optimisation scheme and the use of genetic algorithms in place of this method.

1 INTRODUCTION

Geographical Modelling and Planning (GMAP) Ltd have developed a computer model that simulates the pattern and volume of business that can be expected from a geographical network of sales or service outlets. When coupled with some optimisation scheme, this software can identify business resource planning solutions that provide maximum sales, profit or service results.

Ford uses GMAP software — an optimisation scheme called the Idealised Representation Plan (IRP)[10] — to help to decide where to locate its car dealerships. The underlying spatial interaction model simulates car distribution and sales on the basis of information about the location of dealers and supply points, together with other geographic, demographic and statistical data. This allows prediction of the sales and flow patterns that result from a given configuration of dealerships. Running this software on a Sun SPARCstation 1 it would take around three months of computer time to find a network of 1000 Ford dealers covering the entire UK; thus the serial code has only ever been used to model one region of Britain at a time.

In a collaborative project between GMAP Ltd and Edinburgh Parallel Computing Centre, the GMAP IRP code has been parallelised for the CM-200. In addition, GA techniques have been studied as a replacement for the existing heuristic optimisation scheme. The major objectives of this work were:

delivery network.

- To achieve a reduction in run-time by at least two orders of magnitude, via this parallel solution, in order to allow Ford to advance their strategic planning from a regional to a national level, as well as allowing them to increasing the range and complexity of problems that can be tackled.

- To devise and implement an optimisation scheme based upon genetic algorithms — a stochastic search technique inspired by natural evolution — that will allow the computation of near-optimal network configurations and will out-perform those produced using existing heuristic techniques.

- To assess the potential advantage of parallel computing in the production of powerful, general business planning models and tools.

In the eighteen months of this work, EPCC and GMAP have achieved the following:

- We have produced a data-parallel version of the GMAP code that runs 2700 times faster than the original workstation software.

- We have optimisation results indicating that using genetic algorithms can provide qualitatively better final solutions than standard heuristic techniques.

- We have provided a tool for exploring completely new spatial problems, in terms of scale, accuracy and speed of solution, thereby improving the productivity and performance of both GMAP and their customers.

The next three sections of this paper present background information on the IRP, the spatial interaction model used by the IRP, and the Connection Machine, for which the code was parallelised. Section 5 discusses the parallel implementation of the SIM, and the performance results achieved from this implementation. The following section looks at using genetic algorithms as an alternative to the IRP optimisation algorithm, and results achieved using genetic algorithms are presented in section 7.

2 THE SPATIAL INTERACTION MODEL

Spatial interaction models are used to simulate interactions between a set of demands in a given area and the set of facilities, products or services that meet this demand [10], [1]. Principal factors influencing the degree of attractiveness of the facility and the competition from other facilities in the area.

The UK is divided into postal areas, which are in turn divided into postal districts. Postal districts are groupings of postal sectors. For example, as shown in figure 1, the EPCC postal code "EH9 3JZ" identifies the third postal sector within the ninth postal district, in the postal area "EH" (Edinburgh).

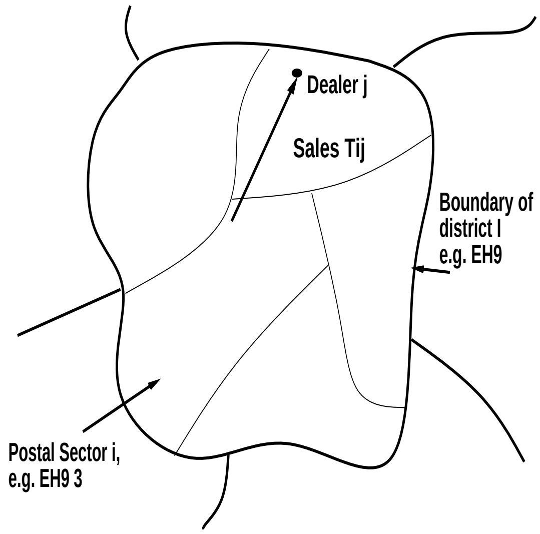


Figure 1: An Example of the Structure of Postal Districts and Sectors.

The task of this spatial interaction model is to predict for each dealer j the rate of sales T_{ij} from each postal sector i. This is computed as follows:

$$T_{ij} = \frac{O_I t_{ij}}{A_I}, \text{ where } i \in I \tag{1}$$

$$A_I = \sum_{i \in I} \sum_{j \in J} t_{ij} \tag{2}$$

$$t_{ij} = R_j W_m \exp(-\beta_i c_{ij}) \tag{3}$$

where

i indexes postal sectors,

j indexes dealers,

m indexes manufacturers,

I indexes postal districts,

J is the set of car dealers that interact with the postal sector i for which the summation is currently being evaluated.

the dealer, and location.

W_m is the average number of sales for a dealer of manufacturer m.

O_I is the number of cars registered in postal district I.

β_i is the distance deterrence parameter. This factor shows how far people are prepared to travel to purchase cars.

c_{ij} is the travel time from postal sector i to dealer j.

t_{ij} is the propensity of residents in postal sector i, to purchase a new car at dealer j .

T_{ij} is the number of cars purchased by residents in postal sector i from dealer j. The task of the SIM is to compute the sales T_{ij}.

A_I is a balancing factor which ensures that all demand is allocated somewhere within the system.

The version of the model detailed in equations 1 to 3 above only considers one competitive sector (e.g. sports cars, family cars) of the car market at a time, and relies upon the assumption that each car dealership only sells cars made by one manufacturer (which is true within the UK).

The following pseudocode solves the above equations sequentially :

For all sectors i

For each dealer j which interacts with sector i

$t_{ij} := R_j W_m \exp(-\beta_i c_{ij})$

For all districts, I

$A_I := 0$

For each sector i in district I

For each dealer j which interacts with sector i

$A_I := A_I + t_{ij}$

For each sector i in district I

For each dealer j which interacts with sector i

$T_{ij} := O_I t_{ij} / A_I$

from that dealer to the sector are estimated. Then all the sales for each postal district are summed, so that each individual sales figure can be taken as a percentage of the total sales within the district. These predicted sales are then balanced by looking at the actual demand in each postal district and weighting the predicted sales according to the percentage of sales they represent within the postal district.

3 THE IDEALISED REPRESENTATION PLAN

The Idealised Representation Plan (IRP) is the heuristic optimisation procedure developed by GMAP to find a set of locations for Ford dealerships within a given area. Given an area of Britain and a set of constraints concerning the size, number and locality of dealerships it may consider, the IRP proposes a network of Ford dealerships to place in that area. Some constraints are "hard", that is they must be adhered to, others are "soft". The soft constraints within the model are prioritised. For example, the size of dealerships being opened is not considered until the minimum number of dealers specified is reached. The objective of the IRP is to find a network which maximises sales for a given area. The network it finds, however, will probably not be that with maximal sales, as the space of possible networks is very large for any realistic problem size, and little search of this space is attempted; the algorithm simply constructs one good solution.

The constraints used by the IRP are as follows:

max_deal a maximum number of dealers in the network,

min_deal a minimum number of dealers in the network,

min_size a minimum size of dealer (where size is number of sales),

min_net a minimum drive time between dealerships,

Initially, the IRP creates a *notional* Ford dealership (which is 10% as attractive as a real Ford dealership) in every postal sector within the region being modelled. Competitor dealerships remain open with full attractiveness. Then a network of dealerships is constructed as follows:

> While fewer than `min_deal` dealers are open
>
> > Run the SIM to calculate the predicted sales for all notional dealers
> >
> > Open the notional dealer with the highest sales.

While the network is changing and sales are improving

For each of the min_deal dealers picked, in random order

Shut the dealership

Run the SIM

Open the best notional dealer

Calculate sales for new network

While sales are improving, less than max_deal dealers have been opened,

and the sales for the new dealer is over the min_size

Run the SIM

Open the best notional dealer

It can therefore be seen that the IRP makes a large number of calls to the spatial interaction model in order to build a complete solution. The aim of this project was initially to parallelise the SIM to speed up the IRP's execution time. In fact the work was later extended to cover parallelisation of the entire IRP algorithm.

4 DATA PARALLEL PROGRAMMING ON THE CM-200

Data parallel programming is a parallel programming paradigm in which data is divided between some number of processors which then perform the same operations on their portions of the data. The Connection Machine CM-200 is a data parallel computing system developed and manufactured by Thinking Machines Corporation. A data set is mapped onto a large number of simple processors, giving a nearly equal number of elements from the set to each processor. All of these processors then execute the same instructions simultaneously, each manipulating the data elements in its own local memory. Alternatively, the programmer may specify a subset of elements to be manipulated, in which case the behaviour is as before except that those data elements not selected are not altered for the duration of this instruction. A CM-200 may contain 8K, 16K, 32K, or 64K parallel processing elements (PEs). The CM-200 is controlled by one or two serial front-end computers, which operate asynchronously with the CM-200. Instructions are only passed to the CM-200 when parallel operations on data in its memory are to be carried out. Communication between the front-end and the CM-200 is via a high speed data bus.

parallel structures contain more data elements than the system has processors (the normal situation), the system operates in "virtual processor" mode. Each processor is effectively divided into several smaller processors. As the program issues parallel instructions, microcode causes it to be executed many times, once for each virtual processor per physical processor. Therefore the same program can be run on different sizes of CM-200 without any changes, subject to memory availability, and as the number of processing elements increases, the program runs faster.

The CM-200 provides the hardware to carry out parallel operations on large numbers of data items, typically arrays of more than one dimension. There are several programming languages that can be used to exploit this hardware; for this project we used C*, which is a superset of ANSI C with parallel language extensions. Details of this language can be found in [2].

The CM-200 based at Edinburgh has 16K bit-serial processors and 512 floating-point accelerators, giving a peak performance rating of 8 GFlops. Each PE has 256K of local memory; this provides a total of 0.5 Gbytes of memory over the entire machine. Additionally, there is a high performance data storage device — the Data Vault. This can store up to 10 Gbytes of data and has a peak transfer rate from the CM200 of 25 MBytes per second. Data can be visualised on a colour monitor, using a framebuffer in the CM, or remotely on a workstation running X Windows.

5 PARALLELISING THE IRP

5.1 Parallel Data Structures

The first stage in developing a parallel version of the spatial interaction model within the IRP for the CM-200 was to design an efficient data distribution mechanism. The principal data structure within the IRP contains data about the postal districts, postal sectors and the dealers with which each sector interacts. This data is arranged hierarchically, each district containing a number of postal sectors, and each postal sector containing a number of dealerships with which it interacts. Each postal sector may interact with up to 100 dealers, these being the closest dealers to the sector. Studies have shown that interactions between sectors and dealers that are further away are insignificant enough not to be worth the extra computational cost of modelling. To give an idea of the size of this data structure, within the UK there are 2625 postal districts and 8500 postal sectors.

There are three levels at which the data could be distributed. Distribution at the postal district level was rejected worst case ten. Thus distributing one postal district to each virtual processor in the CM would result in poor load balancing, as some districts would have considerably more work to do than others. Also, the CM-200 at EPCC has 16K (16384) processing elements, so a distributed array of 2625 elements would be too small for efficient parallelisation. The option of distributing the dealers with which postal sectors interact over virtual processors was seriously considered, as this would lead to a well load balanced solution, with a high degree of parallelism; however, the amount of data to be stored for each dealer would be prohibitively high. Therefore, the decision was made to parallelise the data at the postal sector level, each virtual processor within the Connection Machine manipulating one postal sector and the dealers with which it interacts. Since nearly all of the postal sectors interact with the maximum number of dealers possible, the load balance is good. Figure 2 shows the parallel and serial data structures.

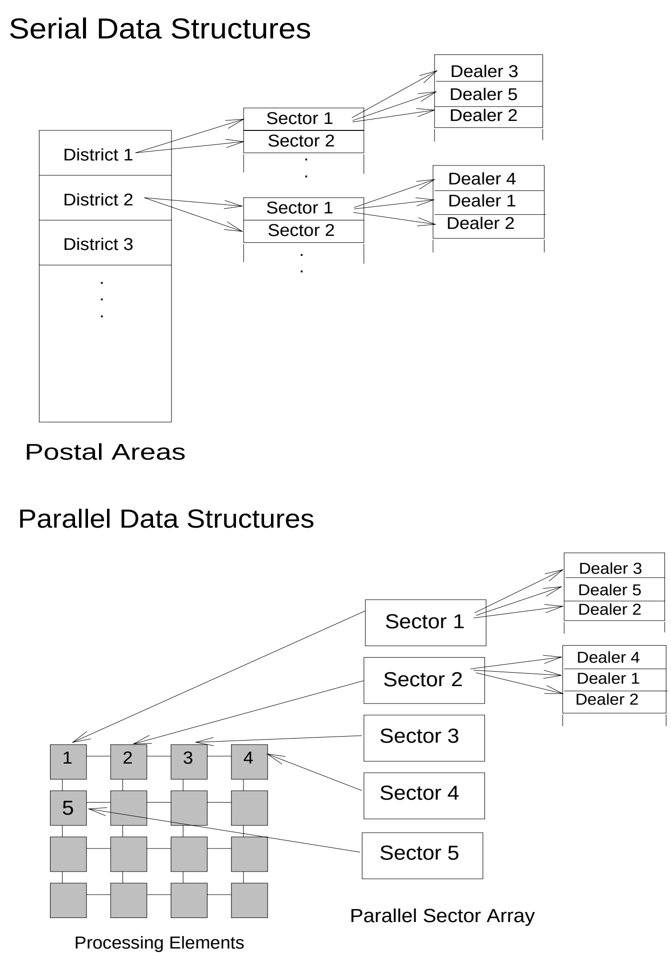


Figure 2: Serial and Parallel Sector Data Structures.

To create a parallel array of postal sectors, a CM array called a `shape` is declared, of which each element is a postal power of two, so this array has size 16384, this being the smallest power of two greater than 8500 — the number of postal sectors in the UK.

Another parallel array was used to hold information on the grouping of postal sectors into districts. This array is the same length as the parallel postal sector array. The postal sectors are arranged in their array in groups of sectors belonging to the same postal district. The postal district array then holds the value 1 where the corresponding postal sector is the first of a new district, and a 0 otherwise.

Finally a parallel array, `par_sales`, is used to collect the sums of sales for each dealer, where each element in the array corresponds to one dealership. When the parallel sector data is set up, each dealer is given an index into the `par_sales` array, then as data for dealers interacting with postal sectors is stored, the appropriate index is stored with every dealer each postal sector interacts with. Although this array holds dealerships and not postal sectors, it can still adopt the postal sector shape, as there are always few enough dealerships in any given area to fit into this shape.

5.2 The Parallel SIM

To simplify the explanation of the parallel SIM, equation 2 from section 3 is rewritten as:

$$a_i = \sum_{j \in J} t_{ij}, \tag{4}$$

$$A_I = \sum_{i \in I} a_i, \tag{5}$$

where the indices are as detailed in section 2.

The spatial interaction model is then parallelised as follows: firstly, the t_{ij} and a_i values are simultaneously calculated over all postal sectors for the first dealer each sector interacts with, then the second, and so on:

For every sector i in parallel

$a_i := 0$

For each interacting dealer j

$t_{ij} := R_j W^m \exp(-\beta_i c_{ij})$

$a_i := a_i + t_{ij}$

in a parallel array. Basically, the postal sectors are grouped into postal districts, then each district's a_i values are summed:

For all districts, I

$A_I := \sum_{i \in I} a_i$

Evaluation of the T_{ij}'s is straightforward:

For every sector i in parallel

For each interacting dealer j

$T_{ij} := O_I t_{ij} / A_I$

Once the spatial interaction model has been run, the sales for each dealer are still distributed amongst the postal sectors in the CM. These sales must be gathered, and the best notional dealership found. These sales are collected in `par_sales`, and are summed for each notional dealer:

For every sector i in parallel

For each interacting dealer j

Where j is a notional dealership

[j_{ind}]par_sales := [j_{ind}]par_sales + T_{ij}

where j_{ind} is the index into the `par_sales` array for dealer j.

Finally, the best notional dealer must be selected and opened. To find the dealer with the maximum sales, a scan is used on the `par_sales` array. The dealer found by this scan is then established as an active dealership using serial code which is not significantly different from the original implementation.

5.3 Parallel Input and Output

In the serial code, data is input from NetISAM files (NetISAM is a file management system produced by SUN microsystems). The parallel implementation loads the data it requires for the parallel array of postal sectors from be run on a new region, a preprocessing program is executed to write sector data to the Data Vault.

The serial output phase wrote postal sector and dealer data to NetISAM files. However, access to NetISAM files is fairly slow, so the parallel output phase was modified to write out data to plain ASCII files instead, from which it could be loaded into NetISAM files using a serial machine other than the CM front end.

5.4 Performance Results

Tables 5.4 and 5.4 give information on the performance of the serial and the parallel IRP codes. The serial implementation was run on a Sun SPARCstation 1, the parallel version on the full 16K processor CM-200. In table 5.4, the figures are estimated for the serial code, as the code would have taken an unreasonably long time to run in reality.

Table 1: IRP finding 100 dealers for the national data set

	Serial Implelemntation	Parallel Implementation	Speedup
Input	4 hours	68 seconds	240
IRP	150 hours	190 seconds	2842
Output	102 mins	2 mins	51

Table 2: IRP finding 1000 dealers for the national data set

	Serial Implementation	Parallel Implementation	Speedup
Input	4 hours	68 seconds	210
IRP	2520 hours (approx. 3 months)	56 minutes	2700
Output	2 hours	12 minutes	10

It can be seen from table 5.4 that using the serial code with the UK data set, choosing a realistically large network of Ford dealerships takes literally months. Having achieved a speedup of three orders of magnitude with the parallel code, such simulations can now be performed in a couple of hours. It is possible to use the serial code to simulate regions of the UK separately, bringing the results for these regions together to find a network of dealerships for the UK. This takes considerably less time than simulating the entire UK; around 20 to 30 hours. This, however, can lead to inconsistencies at the regional boundaries; thus the possibility of simulating the whole of Britain at one time has a significant advantage in terms of the consistency of results.

Another advantage of providing a fast implementation of the SIM is that the use of more compute-intensive techniques to tackle the dealer network optimisation problem can be investigated. This section discusses the application of Genetic Algorithms (GAs) to this problem.

6.1 Introduction to Genetic Algorithms

Genetic algorithms are an optimisation technique modelled on the process of natural evolution. They were first developed by John Holland in 1975 [4]. His idea was to maintain a pool of solutions to a given problem and to allocate to each solution a *fitness* which measured how good it was. Possible solutions were stored as *chromosomes* — strings of numbers, each having some particular meaning in the problem — and "mated" to produce new solutions, with *fitter* (better) solutions being selected for mating more often than less fit ones. Since only some fixed number of chromosomes were stored, those which were less fit tended to “die off”. The representation of solutions to the dealer network optimisation problem as chromosomes is discussed in section 6.2

New chromosomes are formed from existing ones in two different ways, the more important of which is *crossing over*. Crossing over involves taking two parent chromosomes, and for each value (or *gene*) in the new chromosome, using some function to determine which parent that gene should be inherited from. Traditional crossover operators never change the values of genes; they simply arrange existing gene values in different ways. They are analagous to sexual reproduction in animals, from which they draws direct inspiration. However, for more complex problems such as that tackled here, it is often necessary to use more sophisticated crossover operators that may change gene values.

The second way of producing a new chromosome is by changing one or more of the genes’ values in an existing chromosome. This is called *mutation.* It is necessary to have mutation because, as less fit members of the population are killed off, all the instances of some value of a gene may disappear. Crossover alone would only be able to explore an ever-decreasing part of the search space; mutation is needed to maintain diversity and thus allow the search to reach new parts of the space. Thus, we think of crossover as being the driving force underpinning the genetic algorithm and of mutation as being responsible for keeping the gene pool from which it draws well stocked.

A genetic algorithm generally operates as follows:

- Choose some fixed number of chromosomes, say 100, to form an initial population.

- Repeatedly:
  - pick a pair of parent solutions from the population for mating, biasing the selection towards fitter members but ensuring that every member has some chance of participating in mating.
  - produce a child from the chosen parents using crossover
  - with small probability, mutate genes in the new chromosome
  - measure the fitness of the child
  - randomly choose a member of the population to die, and replace that solution with the child.

The production of as many children as were in the initial population, in this case 100, is described as a *generation* of updates. Typically a GA may run for tens or even hundreds of generations to find a good solution.

In the case of the dealer network optimisation problem, the fitness of chromosomes is measured using the SIM. Thus the SIM will be called many times as the GA runs. Using the serial implementation of the SIM, GA techniques would be prohibitively slow for realistic problem sizes. Using the parallel SIM they are fast enough to be practicable. Although GAs have the disadvantage of being slower than the new parallel IRP, they can provide qualitatively better results. It is easy to see that even a small percentage increase in sales for Ford may be worth the expenditure of much computer time.

6.2 Data Representation

The dealer network optimisation problem looks for networks of dealerships throughout a given geographical area, placing at most one dealership in each postal sector. In fact, since coverage of the area is considered to be important in addition to finding a network of dealerships which attract high sales, over-clustering of dealerships is prevented to some extent by allowing only one dealership within each postal district. Dealerships must still be associated with a particular postal sector within a district, in order to evaluate their predicted sales using the SIM.

For the GA solution of our problem, a chromosome is defined in which every gene is a postal district. Within each postal district, the sectors are numbered $1..n$, where n is the number of sectors in the district. For postal districts with no open dealership in them, the gene is given the value zero; districts with an open dealership in some postal sector are given a number of that postal sector. Therefore, if, for instance, a network of 15 dealerships within 40 rest have some value greater than zero. Figure 3 shows the mapping of an area containing five postal districts, with three open dealerships, onto a chromosome.

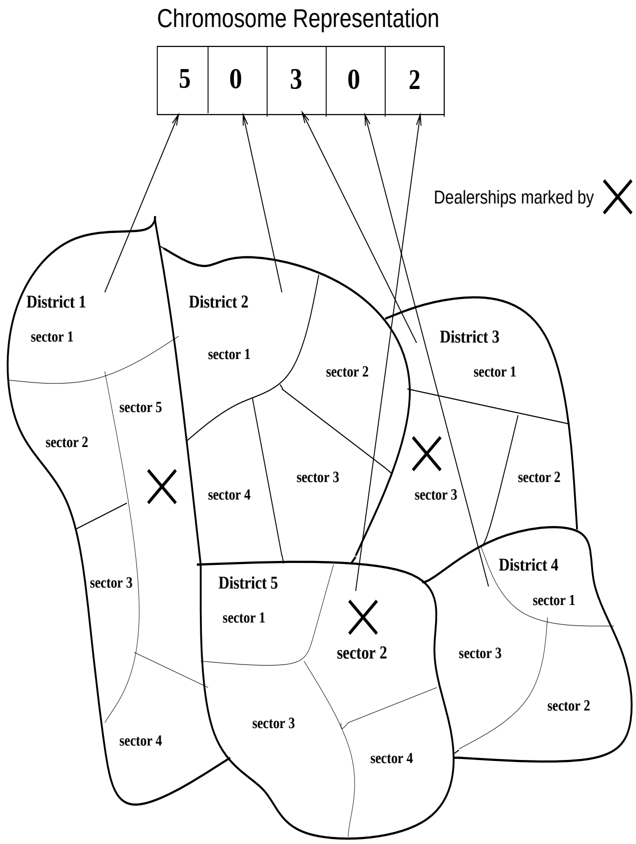


Figure 3: Mapping an area containing five postal districts and three open dealerships onto a chromosome.

6.3 GA Operators

A genetic algorithm is constructed using a series of operators to perform functions such as crossing over two parent chromosomes and evaluating the fitness of a chromosome. For this problem, six classes of operator are defined:

Initialisation These operators create a new chromosome, and are used to generate the initial population. The initialisation operators for this problem are `random-network` (section 6.3.1) and `IRP-network` (section 6.3.2).

Selection Chooses two parents to be combined; an operator called `binary-tournament` (section 6.3.3) is used.

Crossover Crossover combines two parent chromosomes to create a child. `spatial-crossover` (section 6.3.4) is the crossover operator used in this case.

changes to the chromosomes created.

Evaluation The SIM is used as the evaluation operator, measuring the fitness of chromosomes in terms of the predicted sales from the represented network.

Replacement The member of the current population to be replaced by the child is chosen, using the operator `binary-rm` (section 6.3.7).

Some of the operators used for this problem use knowledge of the specific problem area to help create better solutions, either taking advantage of techniques used by the IRP (i.e. hybridisation [5]), or exploiting aspects of the problem such as its spatial nature. Other operators use more general techniques, such as binary tournament selection and replacement operators, a random mutation operator, `local-mutate`, and a random initialisation operator, `random-network`. The evaluation operator used is the spatial interaction model, which is run on a CM-200, whilst the rest of the GA is run on the CM-200's front end. The GA code was designed to make use of a tool called the Reproductive Plan Language (RPL) [9], which simplifies writing and experimenting with GA codes. The RPL software supports arbitrary (user-defined) genome representations, and also allows the user to add functionality to the language by means of C functions, for both representation dependent and independent operations. Examples of operators provided by RPL are `binary-tournament`, `binary-rm`, and `select-best`.

6.3.1 Random-network

This operator creates a random chromosome (i.e. network of dealerships), to seed the initial GA population. For each network, it randomly selects postal districts, then sectors within districts, in which to open dealerships until a given target number of dealerships have been opened.

6.3.2 IRP-network

As an alternative to creating a population of random chromosomes, this operator was developed to allow the GA to start with a population made up from variations to networks of dealers selected by the IRP, thus providing controlled innoculation [6] of the initial population. This operator has one input parameter, i, which indicates the size of the IRP network from which the new network is to be generated, and must be greater than or equal to the size of network to be generated. A network of size i, selected by the IRP algorithm as the best dealerships in the area, is sought, then chromosomes created using this operator will be identical to the IRP solution of this size. Otherwise, they will contain a selection of dealerships which the IRP has identified as profitable. As this operator creates one new chromosome only, chromosomes within an initial population can be created from a variety of IRP network sizes, or some may be created randomly, as desired.

It is worth noting that choosing an IRP network from which to select dealerships which is larger than the network sought not only provides some diversity between chromosomes, but can also improve the quality of the networks found. The IRP selects dealerships which are profitable, but not necessarily ones which combine together optimally. Thus, by taking a larger set of dealerships chosen by the IRP and recombining them in various ways, better solutions than the IRP solution for the given network size may be found.

6.3.3 Binary-tournament

This routine is used to select parent chromosomes: to select one parent, two chromosomes are randomly chosen from the population, and the fittest of these is chosen with a probability provided by the user [7].

6.3.4 Spatial-crossover

This crossover operator considers the spatial relationship between dealerships, using geographical knowledge to build an operator specific to the problem. It has two separate functions, and will choose a method for any given crossover event according to a probability, p, provided by the user. In both cases, a number of localised geographical areas within the whole problem region are chosen (e.g. within the UK these might be an area around London, and one covering Newcastle). The number and size of the areas are chosen according to the operator parameters num and $range$. If the first operator is being used, these areas are taken from one parent, and the remainder of the solution from the other parent. For the second operator, genes within the selected areas are chosen randomly from either parent, while those outside come from one parent only, providing a localised crossover operator. Thus the first operator keeps local solutions constant but exchanges them between solutions, whilst the second keeps most of the solution constant and mixes only a localised area.

In more detail, taking two parent chromosomes, A and B, and parameters num and $range$ specifying the maximum number and size of these areas, the algorithm is as follows:

2. For each area:
    (a) Randomly choose the area size (between 1 and $range$)
    (b) Randomly choose a postal sector around which the area will be centred
    (c) Add all postal sectors within $range$ minutes drive time of the chosen postal sector to the area.
3. Add to the child elements from parent A which are outwith the chosen areas
4. Following the first method:
    (a) add to the child elements from parent B which are within the chosen areas
    (b) If there are too many open dealerships in the child, remove dealers randomly from outwith the chosen areas
    (c) Randomly add extra dealerships if necessary.
5. Following the second method:
    (a) for every element within the chosen areas, randomly chose the parent from which to inherit the value of that element
    (b) If there are too few dealerships in the chromosome, randomly add dealers within the areas only
    (c) If there are too many dealerships, randomly delete dealers within the areas only

An example of groups of postal sectors selected by this algorithm is shown in figure 4.

6.3.5 Local-Mutate

This operator takes three parameters, a chromosome, a probability p and a distance d, which is expressed as a drive time. It loops as many times as there are open dealerships in the chromosome it is given, at each step randomly choosing a number between zero and one, and if it is less than p, either randomly closing a dealership and then opening one within distance d of the closed dealership, or randomly opening a dealership and then closing one within distance d of the newly opened dealership. Thus only local changes are made to the position of dealerships.

6.3.6 IRP-Mutate

This operator is an example of hybridisation, incorporating methods used in the IRP into a GA mutation operator. It takes as parameters a chromosome, a probability p and a number of mutations, m, and functions as follows:

For $i = 1$ to m do

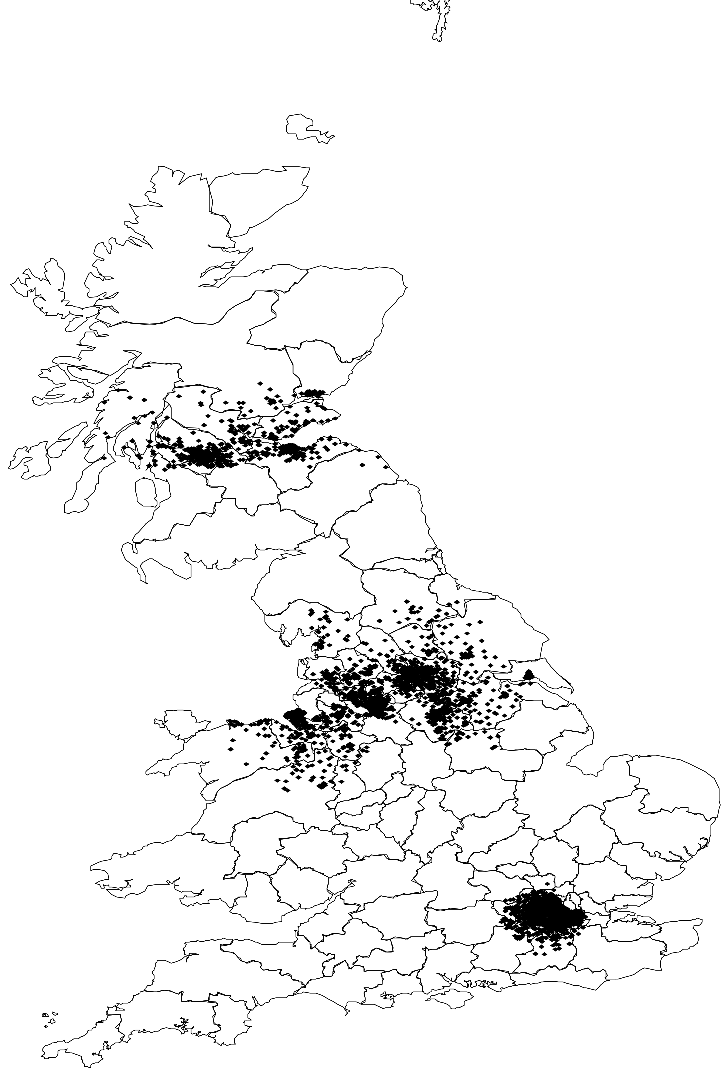


Figure 4: Spatial operator with parameters num = 5 and $range$ = 80 choosing three areas in the UK.

1. randomly choose a real number, r, between 0 and 1.

2. if $r < p$ then

   (a) randomly determine whether to open or close a dealership, using IRP techniques to select the best unopened dealer or the worst dealer currently open, respectively

   (b) if a dealership is to be opened then

      first randomly delete one, then run the SIM with notional dealerships established in every postal sector without an open dealership. Open the best notional dealership.

   else

      (a dealership is to be closed) run the SIM with only dealers in the chromosome open, and close the

As will be seen in section 7, this operator is very successful, both in complementing the performance of the spatial operator and also when used alone without any crossover operator. The disadvantage of this operator is that, while it improves the quality of solutions in terms of maximising sales, it reduces the diversity between chromosomes found by the GA.

6.3.7 Binary-rm

This operator is similar in method to `binary-tournament`. Two chromosomes are randomly picked from the population. The less fit between them is removed with a probability provided by the user. This operator also uses *elitism*, ensuring that the best member of the population is never removed.

6.3.8 SIM

The spatial interaction model, which is used as the evaluation function for the GA, is implemented in parallel as described in section 5.2, with a few alterations:

- The chromosome must be copied from the front end to the CM before evaluation
- After the SIM is run, the dealer sales are still distributed within the parallel sector data structure. To obtain the total network sales, which is returned to the front end as the network's fitness, these sales are summed for each postal sector, and then across the sectors, and the result returned.

The operators `binary-tournament` and `binary-rm` are always used, with parameter 0.6 in both cases and the evaluation function is always the SIM. The choice of other operators (to perform crossover, mutation and initialisation) and their parameters, varies for different problems. New children are generated and placed directly into the population, removing an old member of the population, rather than generating enough children for a new population and then replacing the whole population at one time.

7 GA Results

To test the suitability of GA techniques for this real world dealer network optimisation problem, a variety of different sizes and types of problem have been investigated. Initially, we used a regular triangular lattice of 91 could produce results comparable to those of other standard methods. Secondly, the GA was used to find dealership networks within a single region of Britain, for example, Strathclyde. Various sizes of network were sought within this region, to see how well the genetic algorithm coped with different problem sizes. Finally, the performance of the GA when optimising networks covering the whole UK was evaluated, again using various sizes of network.

7.1 Network Optimisation in the Strathclyde Region

The GA was used to optimise various sizes of network in the Strathclyde region of the UK, to compare the results achievable through the use of GAs to those for existing GMAP techniques.

After testing various combinations of parameters and operators, the following were selected for use in our standard test runs:

- Crossover : `Spatial-crossover`, parameters num = 4, $range$ = 50 and p = 0.5 (i.e. up to four patches with radius of up to 50 minutes drive time are chosen, and there is an equal probability of choosing either method to combine the chromosomes).
- Mutation operators: `local-mutate`, parameters p = 0.05 and d = 25 and `IRP-mutate`, parameters p = 0.2 and m = 1
- Population Size: 75

The graphs presented in figures 5 to 7 show the performance of the GA optimising networks of sizes 20, 35 and 50 dealers in Strathclyde region. All performance measures are taken as an average over at least four GA runs. For each network size, there are two graphs. The first shows the performance of a random search algorithm (instead of crossover and mutation, a random chromosome is generated at each update); an IRP solution; a GA starting with a population of IRP solutions; and a GA starting from a random population. This first graph only shows a few (50 to 80) generations, as the random search and the GA starting with IRP solutions progress very little after quite a small number of generations. In comparison, the second graph shows the performance of a GA starting from a random population over a longer time period (from 250 to 1000 generations). For these tests, the GA ran at about one generation per minute.

These tests show that the GA starting with a random population can improve on the IRP solution for networks of up to 35 dealerships within 1000 generations. For the network of 50 dealerships, it was still improving the

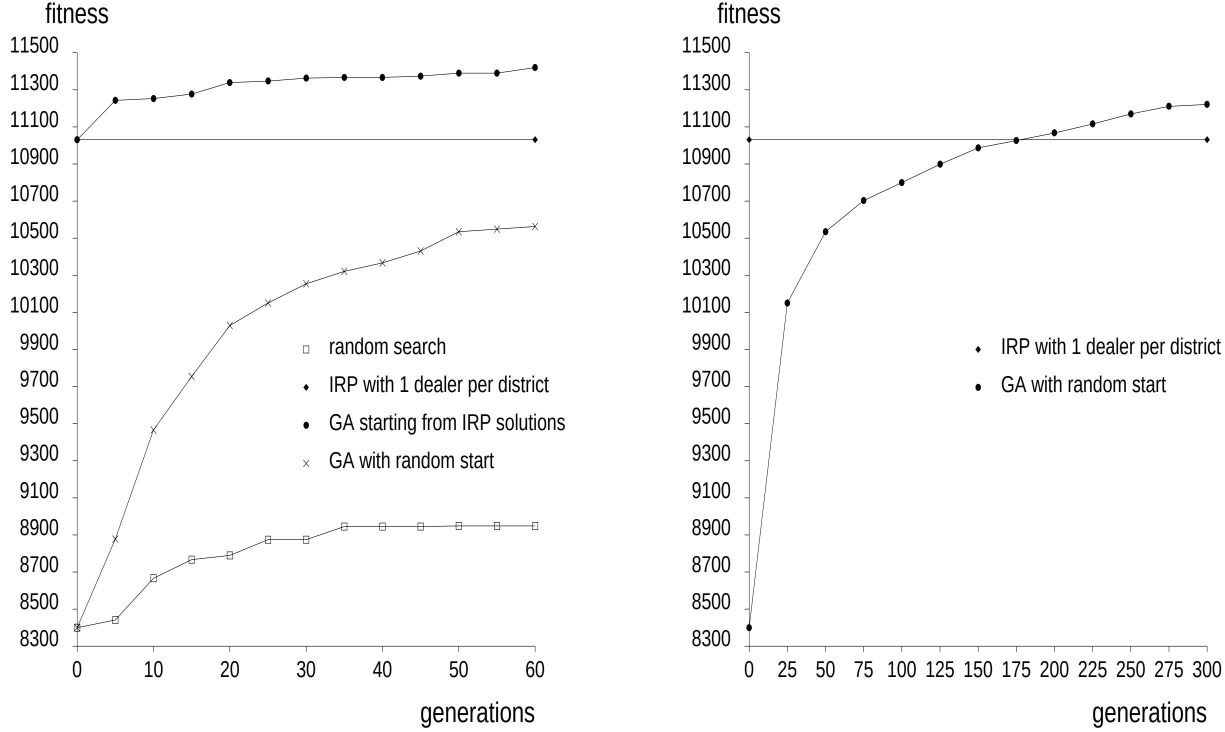


Figure 5: Finding Networks of 20 dealerships in Strathclyde.

network at 1000 generations, and so may have bettered the IRP solution eventually, but the time taken to run many more generations made this impractical. There is a clear pattern — as problem complexity increases, so does the number of generations required for the GA to find a solution better than the IRP. Starting from a population of IRP solutions, the GA finds a better solution than the IRP in all cases, generally after very few generations. Table 3 shows the average fitnesses found at the end of the GA runs and the best solutions found, for all network sizes, including networks of size ten, for which graphs are not presented.

Although the GA starting from a population of IRP solutions generally gives the best final results in terms of sales, both on average and in providing the best solutions, the GAs starting from a random population have the advantage that the solutions they provide are more spatially diverse. While the former differ in terms of spatial structure from the IRP solution by about 20% for all network sizes, the latter differ from between 20% for networks of ten dealerships, to 50% for networks of 50 dealerships. Thus if a few different good solutions are sought, starting the GA with a random population is more effective.

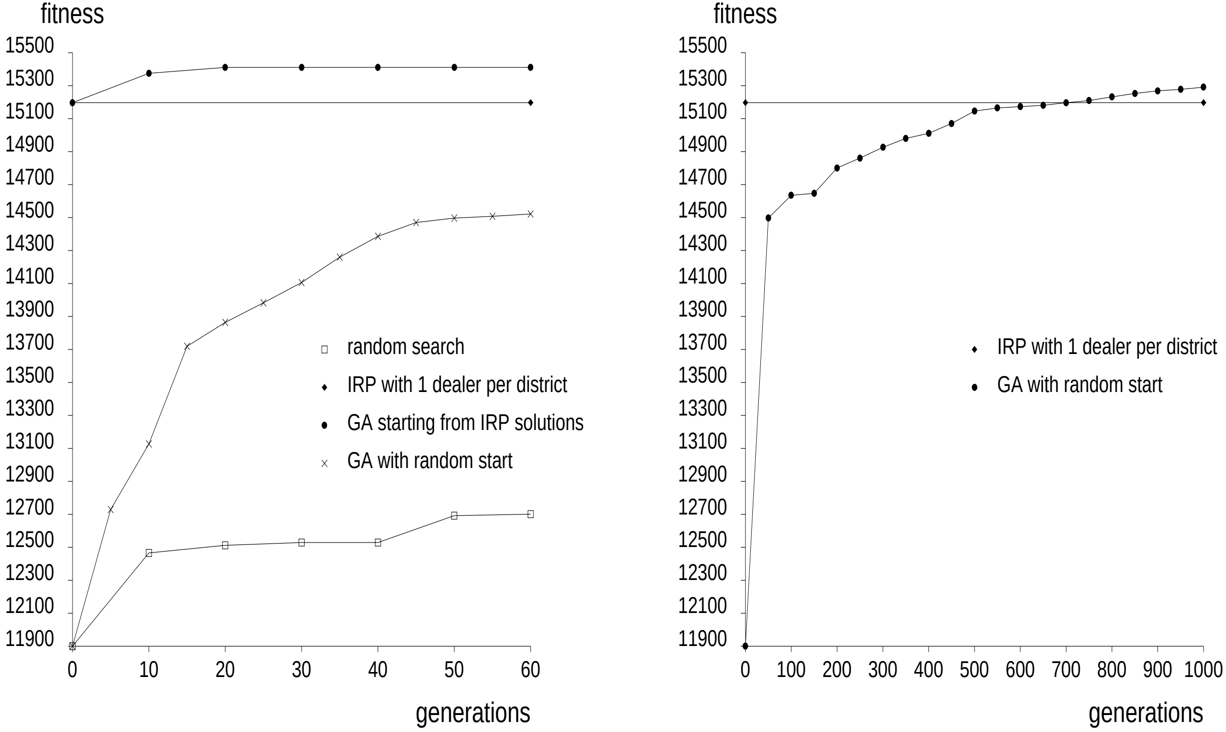


Figure 6: Finding Networks of 35 dealerships in Strathclyde.

7.2 Optimising Dealer Networks Covering the UK

The genetic algorithms used to find networks of dealerships in Strathclyde followed a standard format whether they were starting from a random population or a population of IRP solutions. However, in order to find good solutions to the much harder dealer network optimisation problem for the UK, the techniques used for the generation of different initial populations had to be adapted.

Table 3: Best and Average solutions found by GA techniques for Strathclyde.

Network size	IRP with 1 dealer per district	Best GA soln. starting with a random population	Ave. GA soln. starting with a random population	Best GA soln. starting with IRP networks	Ave. GA soln. starting with IRP networks
10	7100	7579	7562	7486	7429
20	11031	11450	11221	11470	11420
35	15197	15300	15291	15452	15411
50	18468	18364	18291	18495	18489

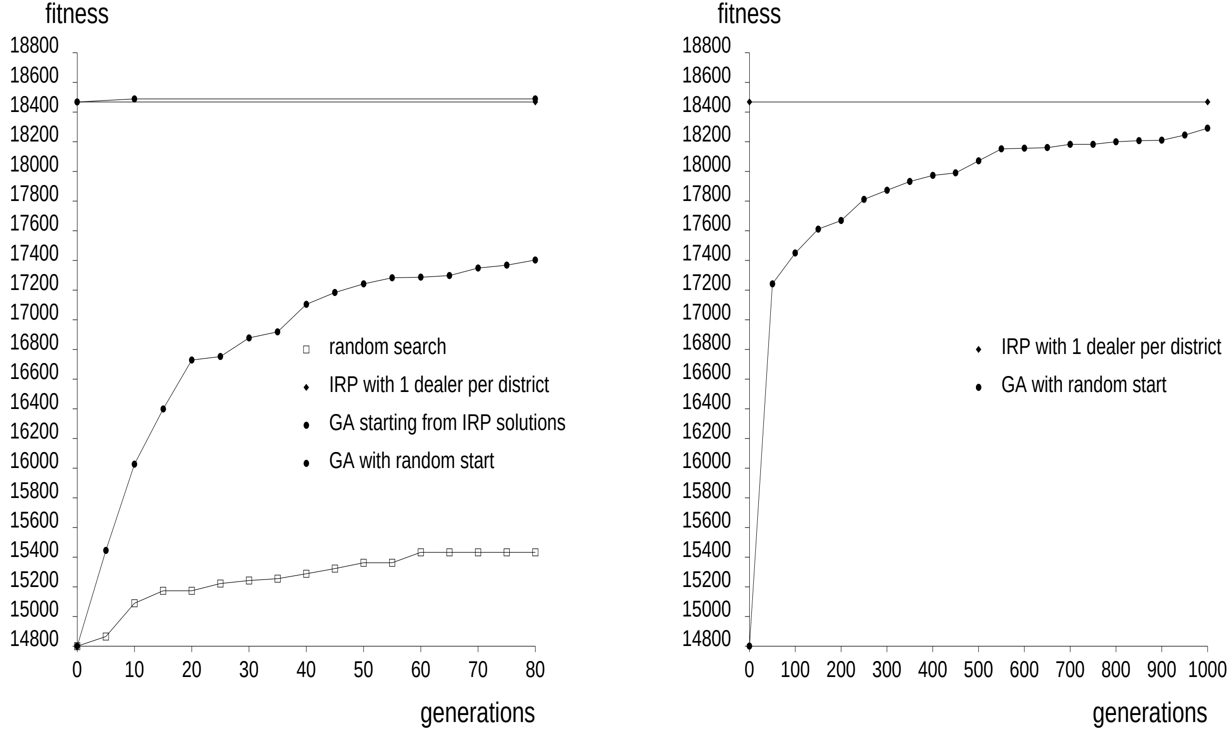


Figure 7: Finding Networks of 50 dealerships in Strathclyde.

7.2.1 Testing GAs Starting from Random Populations

The GA starting from a random population for the UK was very similar to that used for the Strathclyde data set. Various parameters were tried for the spatial crossover operator; choosing up to 10 patches of radius up to 80 minutes drive time proved to be the most effective. The population size was raised to 100. Otherwise, the operators and parameters were exactly as for the Strathclyde region.

The results for optimising UK networks of sizes 10, 50 and 75 are shown in figures 8 and 9, taking averages over three runs in each case. Only three runs were used, as the variance between results on individual runs is very low. The IRP solution, a random search technique and a standard crossover operator (RRR [8]) are shown for comparison. The GA finds networks with higher sales than the IRP for networks of up to, and slightly over, 50 dealers. The results from a single region within the UK, containing only 130 postal districts (as opposed to the 2600 districts within the UK), showed that a GA using the spatial crossover operator could not outperform the IRP when looking for a 50 dealer network in this region. The search space in this case is much smaller; roughly 10^{36} for the single region compared to 10^{106} for the whole UK. This indicates that, as would be expected, the spatial crossover operator performs better when given a larger area to work within, and also possibly that the IRP techniques do not scale well as the problem size increases. It can also be seen from the graphs that the use of the spatial crossover

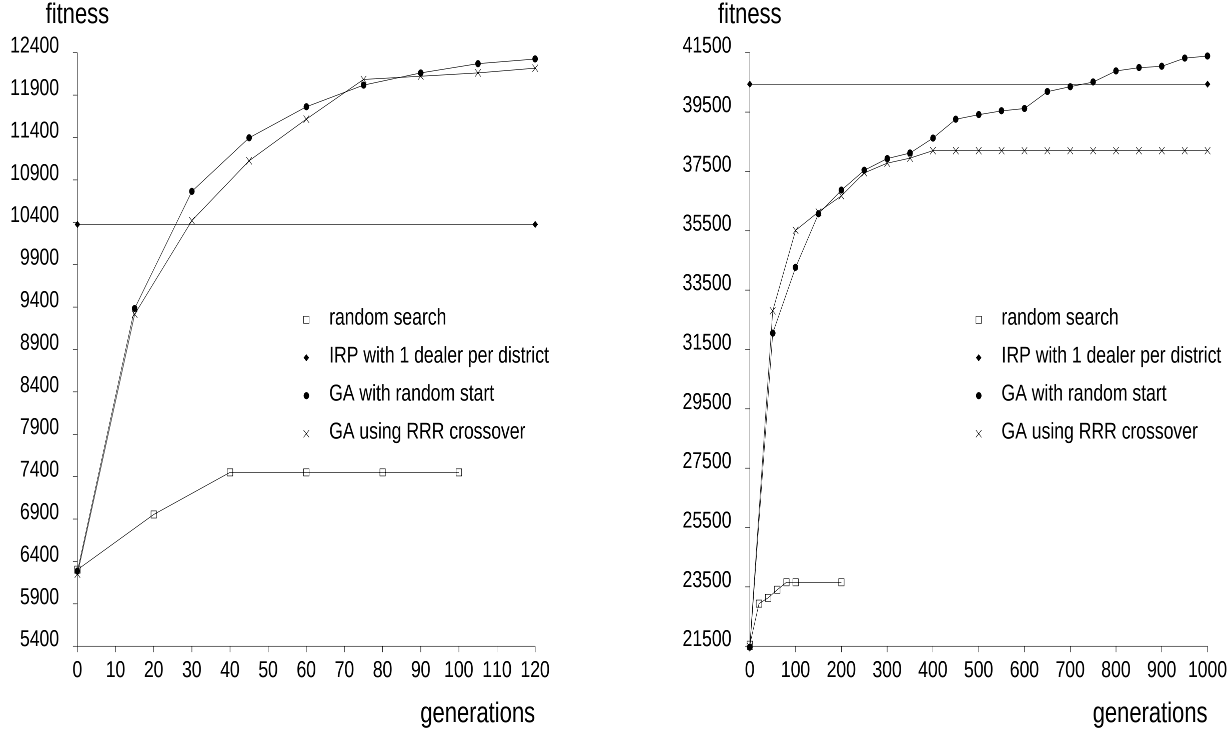


Figure 8: Finding Networks of 10 and 50 dealerships in the UK using a GA starting from a random population.

operator improves solution quality when compared to the more standard RRR crossover technique. This conforms with an expectation that incorporating problem knowledge should allow us to improve performance.

Time was too limited to test the GA for more than 850 generations for networks of 75 dealers; the GA takes about 75 seconds to complete each generation (each generation is 100 updates), running on the Connection Machine 200, so 850 generations take over 17 hours to complete. In the 75 dealer case, although the GA did not do better the IRP solution after 850 generations, it was still progressing steadily.

7.2.2 Testing GAs with Initial Populations of IRP-like Solutions

Most of the work carried out for the UK data set involved finding good initial populations and update techniques for GAs starting from a population of IRP-like solutions, since working from a purely random start requires so many generations. Firstly, initial populations generated by the `IRP-network` operator were examined. For example, for networks of 50 dealers, strategies for creating initial populations varied from mutating 50-dealer IRP solutions, to picking 50 dealers from 120-dealer networks generated by the IRP. Our results indicate that the best method for selecting an initial population from those tested is to take 50 dealerships from an IRP network of 80 dealerships, with no mutation on the resulting networks.

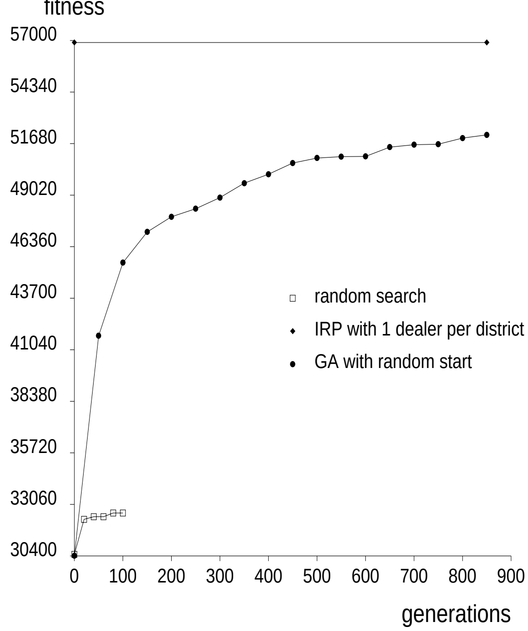


Figure 9: Finding Networks of 75 dealerships in the UK using a GA starting from a random population.

Having found a way to generate a good initial population, the next step was to try and get the GA to combine these chromosomes effectively. Various update schemes were tried, their success being measured by the average fitness of the chromosomes produced by them. It seems from these tests that the most effective update rule is to use the `IRP-mutate` operator alone; this was reinforced by comparing the fittest chromosomes found by a couple of short GA runs using crossover and mutation, and mutation alone. So the update scheme was simply to pick a chromosome from the population, duplicate it, run `IRP-mutate` on the new chromosome, then add it to the population. For different sizes of network, it was obviously necessary to alter the size of IRP network used as input, and the number of IRP mutations done in one update. The figures shown in table 4 were chosen (again by brief comparisons of likely parameters).

Table 4: Parameters chosen for `IRP-network` and `IRP-mutate` using the UK data set.

Network Size	Size of IRP Network input	Number of IRP mutations
10	20	5
50	80	10
75	110	10
100	140	15

The results of the UK GA tests, averaged over three runs, are shown in figures 10 and 11, with the IRP solutions marked for comparison. In all cases, a solution substantially better than the IRP is found in a very small number of involves a call to the SIM. The time required for one generation of updates on the CM-200, when ten IRP mutations are done in every update, is around thirteen minutes. However, only a few generations are required to improve substantially on standard heuristic techniques. Table 5 presents the average and best solutions found by both GA methods for all networks sizes tested for the UK data set.

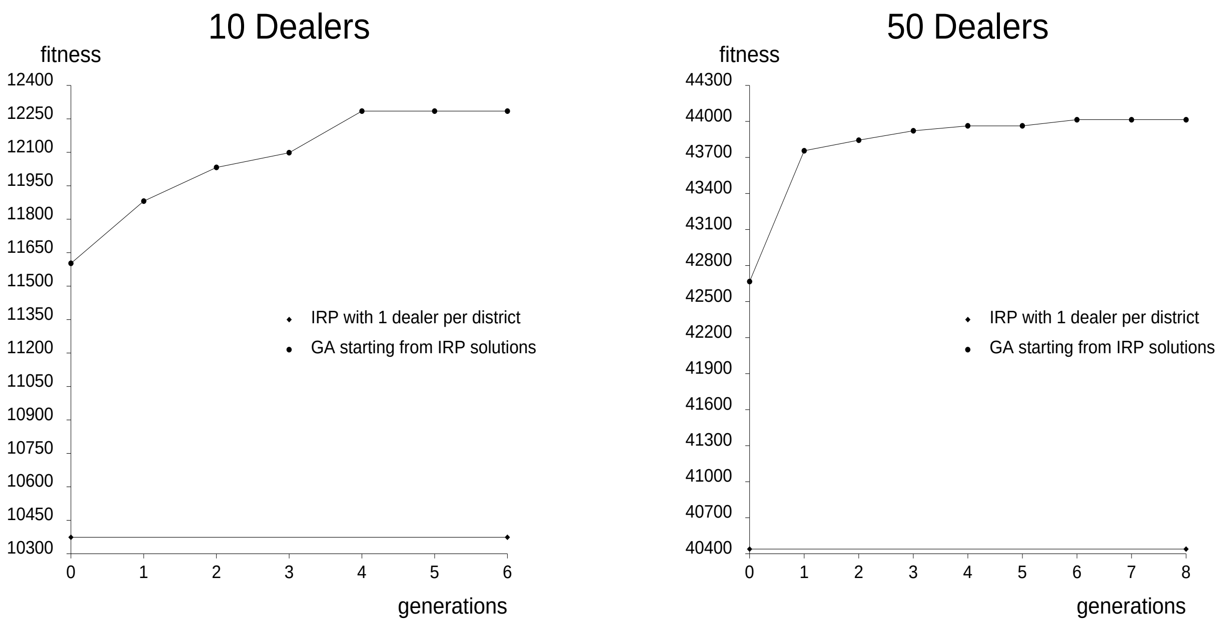
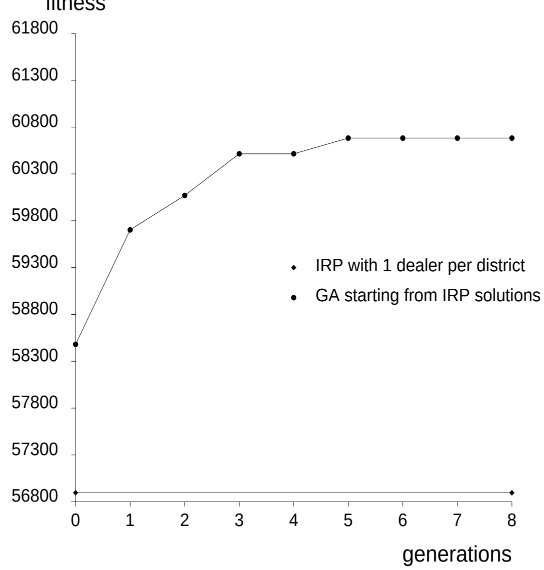


Figure 10: Finding Networks of 10 and 50 dealerships in the UK using a GA starting from a population of IRP-like solutions.

We therefore see that the GA starting from IRP-like solutions does much better in terms of sales than the GA starting from a random population in all cases. However, as was the case with Strathclyde, immunisation hinders the variation in the results; starting with a random population provides a much greater spatial diversity of solutions.

The GA solutions starting from a population of IRP-like solutions differs from the IRP solution with respect to dealership locations by an average of 35%. In comparison, the GA starting from a random population differs from the IRP solution by over 80% on average. Figure 12 show three networks of 50 dealerships in the UK. The first is generated by the IRP and has predicted sales of 40439. The second network is generated by a GA starting from a random population, and has predicted sales of 41443. The final network is generated by the GA using an initial population of IRP solutions, and has predicted sales of 44362. It can be seen that the GA starting from a random population provides a network in which dealerships are more widely distributed than they are in the IRP solution. The GA starting from IRP solutions also generates a network which has slightly less clustering than in the IRP solution. This was the case for all networks examined; in particular, fewer dealerships around the London area



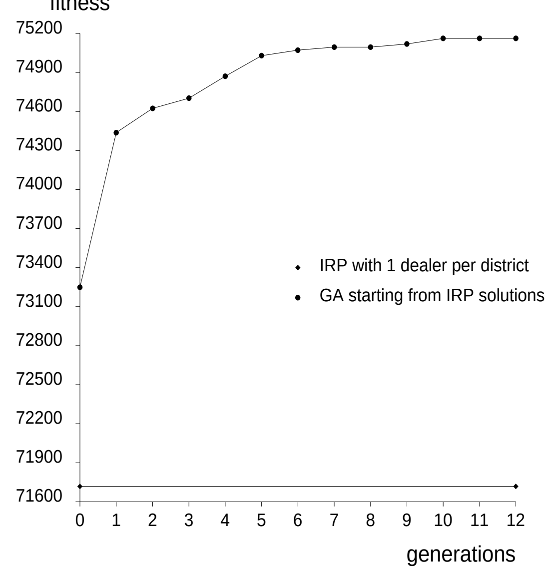


Figure 11: Finding Networks of 75 and 100 dealerships in the UK using a GA starting from a population of IRP-like solutions.

were chosen in every case.

Table 5: Best and Average solutions found by GA techniques for the UK.

Network size	IRP with 1 dealer per district	Best GA soln. starting with a random pop.	Ave. GA soln. starting with a random pop.	Best GA soln. starting with IRP-like networks	Ave. GA soln. starting with IRP-like networks
10	10374	12354	12326	12419	12285
50	40439	41443	41389	44362	44015
75	56896	52458	52130	61799	60683
100	71718	64531	62583	75256	75162

Therefore, in cases where the networks provided by the GA starting from a random population predict high enough sales to be of interest, (perhaps better than or close to the performance of the IRP), networks generated in this manner should provide a useful alternative to those formed using the IRP. Additionally, one run of the GA can supply a set of good solutions, each of which may offer substantially different spatial arrangements which can then be compared to any existing network.

The results presented in this section can be summarised as follows:

- The GA immunised with a population of IRP solutions performs better than the IRP in terms of sales in all cases tested, increasing predicted sales by up to 20%, and generally providing a slightly more widely distributed network. The solutions provided differ in their dealership locations from their corresponding IRP solutions by between 20% (Strathclyde) and 35% (UK).

- Starting the GA with a random population allows the development of more profitable networks than the IRP within short timescales for networks of around 50 dealers. In cases where it failed to find better solutions, the GA was still evolving, and might well have found a better solution given more time. The spatial diversity of these solutions in relation to the relevant IRP solution was high, from between 20% and 50% in Strathclyde, to 80% in the UK.

Thus GA techniques can produce qualitatively better results than the existing IRP technique, especially in terms of the spatial variety of different solutions. Starting the GA from IRP solutions can be used to quickly improve on a solution after the IRP has been run, while the GA starting from a random population can find alternative networks to that provided by the IRP. This could be used if, for example, a close match to an existing network was sought, or a variety of possible network structures should be explored, which may have advantages and drawbacks not accounted for by the SIM. Additionally, one run of the GA can supply a set of good solutions, each of which may offer substantially different spatial arrangements.

8 CONCLUSION

This scale of improved simulation performance, coupled with new stochastic optimisation techniques, obviously has substantial strategic impact for Ford and other businesses wishing to use this new technology. A parallel implementation of a spatial interaction model will be of interest to any organisation wishing to develop or adjust a network of sales or service outlets, since they can now simulate the effects on overall profit/turnover that different structures can produce. This technology may be of particular interest to organisations with existing outlets, who may wish to either expand their coverage, or alternatively rationalise their number of branches. For example, one could imagine that banking organisations could plan the optimal (in terms of number of customers reached) potential customers residing within a certain distance of a store; or a service company needing to cut outlet costs by a given percentage, could identify the new reduced network that still provides maximum service coverage; or retailers could select an optimal subset of stores for a new product launch.

On a wider scale, the technique of using genetic algorithms to perform stochastic optimisation, has very many other areas of applicability. Any business that can increase profitability through better organisation of their complex operational activities, should consider whether they can make use of this advanced technique to discover the best possible strategies. For example, GAs could be used to assist with planning for optimal use of all human and material resources, whether this be the scheduling of machinery and transport, the interaction of projects and staffing, or the optimisation of use of raw materials and products.

9 ACKNOWLEDGEMENTS

EPCC is a multidisciplinary centre supported by major contracts from European industry and grants from the Department of Trade and Industry, the Computer Board and the EPSRC. It is a pleasure to acknowledge substantial additional support from the University of Edinburgh and from the Industrial Collaborators and Affiliates of the centre. In particular, I would like to thank GMAP Ltd., for their support and collaboration in this project.

References

[1] A G Wilson: *Geography and the Environment, Systems Analytical Methods*. Wiley, 1981.

[2] *The Connection Machine CM-200 Series - A Technical Summary*. Thinking Machines Corporation, 1991.

[3] Felicity A. W. George : *Spatial Interaction Modelling on Parallel Computers: Implementation Document*. Project Report, EPCC, 1993.

[4] John H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press (Ann Arbor), 1975.

[5] James D. Kelly, Lawrence Davis : *Hybridizing the Genetic Algorithm and the K Nearest Neighbours Classification Algorithm*, Proceedings of the 4th International Conference on Genetic Algorithms, Morgan Kaufmann, 1991.

[7] D. E. Goldberg, K. Deb : *A Comparative Analysis of Selection Schemes Used in Genetic Algorithms*, Foundations of Genetic Algorithms, G. J. E. Rawlins, 1990.

[8] Nicholas J. Radcliffe : *Genetic Set Recombination*, Foundations of Genetic Algorithms 2, Morgan Kaufmann, 1992.

[9] Claudio V. Russo : *A General Framework for Implementing Genetic Algorithms* Edinburgh Parallel Computing Centre, Technical Report EPCC–SS91–17, 1991.

[10] M Birkin: *Spatial Interaction Modelling on Parallel Computers*. Internal report, GMAP Ltd, 1992.

Random GA Solution

IRP solution

IRP-GA solution

Figure 12: Networks of 50 dealers generated using the IRP, a GA with a random initial population and a GA with an initial population of IRP-like solutions.