Chapter 4

Review of Related Work

This chapter examines the support already available for fulfilling GA users' visualization requirements. Section 4.1 describes a range of visualizations available for showing the key characteristics of GAs suggested in the overview of EC (Chapter 2). Section 4.2 presents a brief overview of some systems which exemplify these key characteristics. Finally, Section 4.3 concludes this chapter with a summary of the contributions made by these systems.

4.1 Visualizing a GA’s Key Characteristics

Section 2.3 identified a set of seven key characteristics of GAs that were considered potentially useful for understanding a GA’s search behaviour; namely: showing the operation of the GA, the quality of the solutions found, the chromosomes’ genotypes and phenotypes, the GA’s sampling of the search space, the user’s ability to navigate through the GA’s execution, and editing the GA’s population or algorithm configuration. These were used to inform the design of the GA user questionnaire used in Chapter 3, and are used again here to structure this section, where their relevance to the GA user’s visualization needs is discussed along with the visualization support currently available; each subsection concludes with a summary of the contributions made.
CHAPTER 4. REVIEW OF RELATED WORK

Figure 4.1: The internals window available in Giga for showing the actions of the reproduction operators of a GA.

This example was taken from [Dabs and Schoof, 1995, page 8].

4.1.1 The Operation of the GA’s Component Parts

Visualizing the operation of the GA’s component parts, i.e. the actions of the algorithm’s selection and reproduction operators, can be done either by using a static or dynamic illustration. The static illustration benefits from being easy to present on paper as well as the computer screen, although viewing a dynamic illustration (i.e. an animation) is often a more effective and more engaging representation.

An example of a static illustration of a GA’s components is the “internals window” available in the Graphical user Interface for Genetic Algorithms (“Giga”) [Dabs and Schoof, 1995]. This view illustrates the internal operations of the GA such as the crossover and mutation operators (see Figure 4.1). A sample dynamic algorithm animation of a GA was produced by David Brogan using an SV system called “Tango” [Stasko, 1989]. Brogan’s illustrative example is included in the example visualizations supplied with the X windows version of Tango, available via ftp from per.cc.gatech.edu (see directory /pub/xtango). A screen view is shown in Figure 4.2 depicting both phenotype and genotype visualizations. An algorithm animation is shown at the bottom of the view which illustrates the actions of the GA’s genetic operators.

Contribution

The actions of the GA’s operators drive the GA’s search in the problem space. Therefore, it would be reasonable to assume that illustrating the execution of the operators would provide some insight into GA’s search behaviour. However, both the Giga and Tango visualizations of the GA’s genetic
Examining the actions of the GA’s selection and reproduction operators was one of the GA characteristics that the study respondents were not directly interested in other than as an educational or debugging aid. Therefore, the visualization of the GA’s operators is not pursued further within this thesis, although provision for such support could be made in the future using the visualization framework presented in Chapter 6.

### 4.1.2 The Quality of the Solutions Found by the GA

Examining the quality of the solutions found by a GA is an important part of applying a GA. Monitoring the GA’s progress can be used to inform the user’s decision to end the GA’s run, or as a post-mortem technique for illustrating the GA’s run. This subsection presents a variety of techniques for showing the quality of a GA’s solutions, including both summaries and complete accounts of the
entire run’s results, as well as the results in individual populations.

The standard method for presenting a summary of the entire run’s results is to plot some aspect of the population’s fitness ratings for each generation. These visualizations are commonly referred to as “fitness versus generation number” or “fitness versus time” graphs. Fitness versus time graphs first appeared in one of the earliest papers on simulated evolution written by A. S. Fraser in 1957 [Fraser, 1957]. Fraser used 2D line graphs to illustrate the changes in the population’s average phenotype fitness value over successive generations. An example taken from Fraser’s paper is shown in Figure 4.3.

A variety of fitness versus time graphs are commonly used today, examples include “online” and “offline” fitness ratings (i.e. the mean fitness rating, and mean current-best fitness rating across all generations [De Jong, 1980]), as well as the best and worst fitness ratings in each population [Goldberg, 1989], [Davis, 1991], [Baeck, 1996].

Although the fitness versus time graph is the most commonly used representation of the GA’s entire run, it is incomplete in that it only provides an indication of the chromosomes’ fitness ratings in each population rather than the actual chromosomes’ fitness ratings. The 3D fitness graph presented in [Harvey and Thompson, 1996] provides a more complete view, showing the fitness ratings of every chromosome in a fitness-ordered population (see Figure 4.4).

The 3D fitness graph presents all the chromosomes’ fitness ratings, but if presented as a static view some sections of the lines may be hidden by earlier and fitter line sections. A solution to this
Figure 4.4: An example of a 3D fitness graph. The fitness rating of each individual in the population is plotted over each generation. The fitness ratings are plotted on the y axis ($y = 0$ to $2.5$), the position of each chromosome in the fitness ordered population is plotted on the x axis ($x = 0$ to $50$), and the generation number is plotted on the z axis ($z = 0$ to $522$). This figure was taken from [Harvey and Thompson, 1996].

Another point to be noted regarding the 3D fitness graph is that the individual lines do not refer to the same chromosomes, rather they refer to chromosomes at the same position in the fitness ordered population across different generations.

Rather than examining the quality of the solutions found during the course of the GA’s run, a number of visualization techniques for illustrating the fitness ratings of the chromosomes in a single generation were proposed in a previous project [Collins, 1993]. The techniques explored included block diagrams, colour maps, bar charts, radial line graphs and radial point plots.

“Hinton diagrams” are used in the study of artificial neural networks to illustrate the strengths of the links between the nodes in a network (see [Rumelhart and McClelland, 1986, page 103]). A Hinton diagram is made up of a series of coloured blocks used to indicate the network weights, the size of the block indicates the magnitude of each link’s weighting, and the colour; black or white, indicates whether the weight is positive or negative. A diagram based on the Hinton diagram illustrates the fitness values of the chromosomes in a population (see Figure 4.5). The size of each block indicates
CHAPTER 4. REVIEW OF RELATED WORK

Figure 4.5: A Hinton style block diagram. This figure illustrates each chromosome in the population as a square block; the size of the block indicates the chromosome’s fitness rating. Colour is used to highlight the chromosomes’ level of fitness, here the fitness ratings are split into four bands corresponding to the four sizes of blocks used in the figure. This figure was taken from [Collins, 1993a] where texture was used to indicate colour on a black and white printer.

A colour map shows the fitness rating of every chromosome in a population. These are similar to Hinton-style diagrams but use colour only to indicate each chromosome’s fitness rating, the size of each square remains constant (see Figure 4.6). The ordering of the individual squares in a colour map can be used to illustrate different aspects of the population, ordering by fitness emphasizes the frequency of individuals with similar fitness ratings, whilst ordering by a similarity measure can emphasize the diversity of the population and the possibility of multiple solutions.

Coastline fitness diagrams show the fitness rating of each chromosome in the population as a long
Figure 4.7: A Coastline Fitness Diagram showing the chromosomes in two populations (a) and (b), here a fitness ordered view is shown on the left and a similarity (i.e. Hamming distance) ordered view is shown on the right for populations (a) and (b). Both views are ordered from left to right for increasing fitness and similarity ratings. This figure was taken from [Collins, 1993a].

Figure 4.8: A radial plot of the fitness ratings in a single generation. The radial line trace shows the fitness ratings of the chromosomes in a fitness ordered population the distance \(m\) from the centre to the line indicates the magnitude of the fitness rating.

vertical bar; the height of each bar indicates each chromosome’s fitness rating. Like colour maps, different ordering methods can be applied in order to illustrate different features of the population. For example, the fitness rating could be used to illustrate the diversity in fitness, or a similarity rating (such as Hamming distance to the fittest) can be used to indicate the diversity in the chromosomes’ values. Figure 4.7 shows the coastline fitness diagrams of two populations, one for an unfit population (a) and one for a fit population (b), the two views in each case illustrate alternate ordering methods; by fitness (shown on the left) and Hamming distance to the fittest (shown on the right).

In a radial fitness diagram a single radial line trace is used to illustrate all of the chromosomes’ fitness ratings in a fitness ordered population. The angular position indicates each individual chro-
Figure 4.9: Three radial fitness plots illustrating three different stages during an algorithm’s execution.

Figure 4.10: A radial fitness plot. Each individual’s fitness rating is represented as a dot. The fitness ordered position of each chromosome is represented by the angular position of each dot, the distance (m) of a dot from the origin indicates the magnitude of the fitness rating, and the dot’s colour indicates the generation number in which it last appeared.

The final fitness plot suggested in [Collins, 1993] was a Fossil fitness diagram. These can be used to present either the fitness ratings of the chromosomes in a single generation, or the fitness ratings of all the chromosomes in every population across a number of generations (see Figure 4.10).

In both cases each chromosome is represented as a dot. The angular position of each dot indicates the chromosome’s position in the fitness ordered population, the distance from the centre of the display to each dot indicates the chromosome’s fitness rating, and the colour of the dot indicates the generation in which that chromosome last appeared, ranging from red for generation 0 to blue for the
final generation. The overall result is a series of coloured markings, similar in shape to an ammonite (i.e. a spiral fossil). The number of dots at each angular position illustrates the diversity in the chromosomes' fitness ratings, in the fitness ordered population, over an entire GA run. Again like the 3D fitness graph, the same angular position does not indicate the same chromosome in different generations, rather each angular position shows all of the chromosomes at the same position in a fitness ordered population.

**Contribution**

None of the fitness plots described in this subsection suffer from any scaling problems, all of these plots are applicable to any size of population and any form of EA. Table 4.1 summarizes the defining characteristics of each fitness visualization.

Although there are a range of visualizations available for showing the quality of the solutions found during a GA's run, the results of the GA user study indicated that the traditional 2D fitness versus time graph was by far the most popular (see Section B.3, Question 7.3). However, several respondents also indicated a need for a more detailed understanding of the GA's run. These responses refer to a need to understand the local structure of the search space and the relationship between the local structure and fitness ratings, rather than a more complete understanding of the chromosomes' fitness ratings in each population. The provision of this is discussed in subsection 4.1.5.

### 4.1.3 The Chromosomes' Genotypes

Viewing the chromosomes' genotypes is usually carried out either for an entire population or a subset of the population, for example by displaying the best chromosome or top five chromosomes in each generation. Although displaying the genotype of a few chromosomes per generation gives the user an indication of the solutions currently being considered it is impossible for the user to view every chromosome from every generation and grasp the GA's behaviour - there is simply too much information for the user to deal with. As a result, several systems have been developed using visualization techniques to represent this information in a more manageable form.

Three chromosome icons were introduced in [Collins, 1990] for illustrating the chromosomes' genotypes; the "trace icon," "DNA strip" and "colour strip" (see Figure 4.11). A "trace icon" is a 2D
Table 4.1: The defining features of a range of EA fitness visualizations.

<table>
<thead>
<tr>
<th>VISUALIZATION</th>
<th>GRAPHIC</th>
<th>CONTENT</th>
<th>PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>2D Fitness vs time graphs</em></td>
<td><em>2D line graph</em></td>
<td>Summary of fitness ratings</td>
<td>per generation for every generation</td>
</tr>
<tr>
<td><em>3D Fitness vs time graphs</em></td>
<td><em>3D line graph</em></td>
<td>Every chromosome’s fitness rating</td>
<td>per generation for every generation</td>
</tr>
<tr>
<td><em>Hinton style block diagrams</em></td>
<td><em>2D extended point plot</em></td>
<td>The chromosomes’ fitness ratings</td>
<td>for a single generation</td>
</tr>
<tr>
<td><em>Colour maps</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Coastline fitness diagrams</em></td>
<td><em>2D radial line graph</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Radial fitness diagrams</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Fossil fitness diagrams</em></td>
<td><em>2D radial point plot</em></td>
<td></td>
<td>for a single generation, or per generation for every generation</td>
</tr>
</tbody>
</table>
Figure 4.11: Three example chromosome icons showing the design of line trace, DNA strip, and colour band icons. This figure was taken from [Collins, 1993a], where texture was used to indicate colour on a black and white printer.

The trace image construction of the allele held at each locus in the chromosome. The variation in the vertical position of the trace at each line segment indicates the allele’s position in the coding alphabet for each locus. A “DNA strip” is a 2D line plot showing each allele as a vertical bar, the horizontal position of the bar indicates the allele’s position in the coding alphabet. Thirdly, a “colour strip” icon shows the allele held at each chromosome locus as a coloured block, the colour of each block indicates the allele’s position in the coding alphabet.

Bill Spears at the US Naval Research Lab has also explored the use of visualization within GAs [Spears, 1994]. In order to illustrate the chromosomes in a specific population Spears suggested illustrating the alleles in a population of binary chromosomes as black and white pixel dots. The resulting pixel-oriented visualization shows a random set of black and white pixels for the initial population with patterns of vertical black and white lines forming during the GA’s run indicating common genes between neighbouring chromosomes (see Figure 4.12).

Although developed separately, the pixel-based genotype visualization proposed by Spears is similar to the color strip icon proposed by Collins. Spears’ representation can illustrate bigger genotypes
CHAPTER 4. REVIEW OF RELATED WORK

Figure 4.12: A high dimensional visualization showing a population of 100, 1008 bit binary chromosomes as black and white pixels. The entire population is shown here in 100 rows each chromosome is shown as a single row of 1008 pixels.

Figure 4.13: An example of a Vis "run window," illustrating the best individual from each generation using a "zebra" representation.

than the color strip icon in the same amount of screen space, but the legibility of each pixel point would be poorer than the legibility of each coloured block. For any specific application the purpose of the visualization should be used to determine the balance between screen economics and image legibility. The purpose of Spears' pixel-oriented visualization is to help people spot emerging patterns within the population, where as the purpose of the colour strip icon was to directly illustrate the alleles in each chromosome's genotype.

Another more recent project at the US Naval Research Lab has been exploring the use of GAs for modelling viruses (the "Virtual Virus" project [Grefenstette et al., 1997]). As part of this project an offline (post-mortem) visualization tool called Vis has been developed to support the detailed analysis of a GA's run [Wu et al., 1997], [Wu et al., 1998]. Vis presents three different perspectives on a GA's run. Run windows display information on the entire run (typically showing one entry per generation, see Figure 4.13). Population windows display single individuals from a single generation (see Figure 4.14). Thirdly, Individual windows display information about a single individual (see
CHAPTER 4. REVIEW OF RELATED WORK

Figure 4.14: An example of a Vis “population window” illustrating all the individuals in a single generation using a “zebra” representation.

Figure 4.15: Within Vis multiple windows can be viewed simultaneously.

Five different genotype representations are available in Vis, namely: “text,” “zebra,” “neapolitan,” “colour coded” and “gene location” representations. The representation used within any of the windows can be changed at any time via the “Views” menu. The text representation simply displays the individuals using a fixed width type font. The zebra representation displays binary chromosomes as strips of black and white bars, like a zebra’s stripes. The neapolitan representation displays every pair of binary alleles as a coloured bar, where 00 = black, 11 = white, 01 = magenta, and 10 = orange. The colour coded representation is used to illustrate multi-letter alphabets (i.e. coding alphabets with more than two symbols), where each unique letter is shown by a different coloured bar (e.g. A = blue, C = red, G = yellow, and T = green). Finally, the gene location representation can be used to highlight the occurrence of building blocks (i.e. groups of symbols or partial solutions), different coloured strips are used to identify different building blocks.

Although it is easier to identify trends within the population using a chromosome icon representation rather than printed text, both printed text and chromosome icons present the same amount of information and therefore, suffer from the same drawback i.e. when applied to large populations they both contain too much information for the user to deal with. As a solution to this [Collins, 1993] proposed three composite representations for summarizing the chromosomes’ genotypes; “overlaid line trace icons,” “population bar charts,” and “allele versus locus frequency matrices” (see Figure 4.16).
CHAPTER 4. REVIEW OF RELATED WORK

Figure 4.15: An example of a Vis “individual window” showing the data held on a single individual.

Figure 4.16: Three example genotype visualizations; “overlaid chromosome icons” (left), a “population bar chart” (middle) and an “allele versus locus frequency matrix” (right). These three chromosome visualizations are taken from [Collins, 1993a].

The overlaid line trace icons representation is produced by plotting an enlarged version of every chromosome’s line trace icon on the same set of axes. The composite image indicates the allele diversity at each locus within the population, by the number of vertically aligned separate line segments. For large (i.e. most practical) population sizes the overlaid chromosome icon representation becomes overloaded and difficult to read (see Section 5.2.3, Figure 5.4 on page 142 regarding the graphic density and angular separation of legible images). Although the line trace icons identify each chromosome and its alleles, they do not indicate the frequency of each chromosome (or chromosome building block). Therefore, the user cannot see when the population is completely converged at a specific locus, but they cannot see the diversity of the population prior to that point. For example, a population containing equal numbers of two different chromosomes would look the same as a population
that contained 90% of one chromosome and 10% of an other.

The population bar chart summarizes the alleles that are present within the chromosomes in the population. Each bar indicates the alleles present at each locus, the height of the bar is used to indicate the most frequent allele at that locus. Lines are added to indicate the minimum and maximum allele values at each locus for the current population. Although this gives an indication of the population's diversity, like the overlaid chromosome icon representation it does not illustrate the distribution of the alleles. As a result, the user is no better informed about the diversity of the chromosomes in the population.

Thirdly, allele versus locus frequency matrices illustrate the distribution of the allele within a population. By viewing the allele versus locus frequency matrices of subsequent generations the user can see how the allele's distribution varies during the GA's run. This shows both the convergence and diversity of the alleles. However, it does not show any information regarding the local structure of the alleles within each chromosome. The allele versus locus frequency matrix gives a clear summary of the distribution of alleles and is perhaps the clearest of the three genotype summary representations proposed in [Collins, 1993].

Contribution

Although exploring the fine-grained details of the individual chromosomes can be very useful for examining the solutions found, like the visualization of the genetic operators, it is at too fine-grained a level of detail to help people follow the overall search behaviour of the algorithm. The responses given in the GA user study (Section B.3, Questions 7.1, 7.2, and 8.1) indicated that the respondents also believed that displaying all the chromosomes would present too much information for their purpose. They also considered the selection of a subset of the chromosomes in each generation to be a difficult task, resulting in an un-representative, or possibly misleading, visualization. Therefore, genotype visualizations must be used carefully to complement the user's exploration of the the GA's search behaviour. Perhaps if used in tandem with a visualization of the GA's sampling of the search space, the fine-grained focus that genotype visualization provides could be directed toward the more significant and interesting chromosomes within the GA's run.
4.1.4 The Chromosomes' Phenotypes

Visualizing the chromosomes' phenotypes is a very effective way of illustrating the solutions being considered by the GA (see Figure 4.17). Several education-oriented GA tools illustrate the GA's phenotypes, specifically for the traveling salesperson problem. Examples include the “best individual window” in GIGA, the phenotype view presented in the XTANGO sample GA visualization and the “Best-So-Far” window available in the Genetic Algorithm Software Development Package produced by EvoNet, the European Network of Excellence on Evolutionary Computation (available from http://www.dcs.napier.ac.uk/evonet/coordinator/html/software.html).

Although visualizing the chromosomes' phenotypes can produce a very salient illustration of the solutions being considered they are specific to the problem being solved and therefore, as new problems are attempted new views must be produced. If the effort involved in producing the view is perceived to be greater than the benefit achieved through its use then the user will be disinclined to produce new views.

This “ease of production” threshold is a serious problem for SV. Producing any new visualization requires some form of programming. The important issue here is to ensure that the programming involved is sufficient to fully express what the user needs, whilst remaining at a sufficient level of abstraction such that the user does not get deterred by technically demanding graphics programming.

One of the primary goals of producing an SV development environment, such as

---

1The traveling salesperson problem is a problem in which the GA attempts to find the shortest route linking a set of cities.
CHAPTER 4. REVIEW OF RELATED WORK

Balsa [Brown and Sedgewick, 1985], Tango [Stasko, 1990], Zeus [Brown, 1991], or Viz [Domingue et al., 1992], is to facilitate the development of new views. Although a great deal of work has been done in SV, establishing a sufficient level of expressiveness whilst maintaining ease of use is a difficult trade off (see [Repenning and Ambach, 1996]). As a solution to this problem John Stasko, author of the Tango and Polka SV environments, developed “Samba,” an interpreted, interactive animation front-end to Polka [Stasko, 1996]. Samba is used by students in an undergraduate algorithms class at the Georgia Institute of Technology to produce algorithm animations from recorded data files or the output of a program piped directly to Samba.

Contribution

Producing problem-specific visualizations of the chromosomes’ phenotypes is a very salient illustration of the GA’s solutions. Such views explicitly illustrate the link between the chromosomes’ genotype and phenotype. This is why phenotype visualizations are so useful when illustrating the GA’s operation within an educational context. However, visualizing all of the chromosomes’ phenotypes in a typical GA produces too much information for the user to digest easily, yet like the genotype visualization described in the previous subsection selecting a representative subset can be problematic. Again, perhaps such detailed views are best used selectively to illustrate the more important chromosomes in the GA’s run.

4.1.5 The GA’s Sampling of the Search Space

The term “search space” is used repeatedly in this thesis to refer to the complete set of all allele combinations available within any given coding alphabet. Exploring the GA’s sampling (i.e. searching) of that space is one way of viewing the GA’s behaviour. This subsection describes some of the available visualizations.

In addition to his genotype visualization tool, Bill Spears also produced two visualization tools to illustrate the GA’s sampling of the search space; one for one-dimensional problems and a second for two-dimensional problems [Spears, 1994]. The first tool uses a 2D line graph to illustrate the fitness rating (plotted on the y axis) of each chromosome (plotted on the x axis). The second tool adopts

\footnote{The term “pipéd” is used here with reference to the UNIX pipe command “|” e.g. “% yourprog | samba”}
Figure 4.18: A 3D surface plot showing the fitness surface for a two-dimensional search space. The chromosomes from old generations shown as blue dots and the chromosomes in the current generation shown as red dots, this figure was taken from [Spears, 1994].

A similar approach but uses a 3D plot to show the variation in fitness for two-dimensional fitness functions. In the 3D visualization the individual chromosomes are shown as points on a 3D fitness surface, as the GA evolves old chromosomes from previous generations are drawn as blue points and chromosomes created in the current generation are drawn as red dots (see Figure 4.18). Both of these tools illustrate the GA’s sampling of the search space by explicitly plotting a line or surface showing the fitness landscape (i.e. the complete search space with its associated fitness ratings) and highlighting the population’s sampling points. However, this approach is not possible for real problems in which the fitness landscape (i.e. the fitness rating for every possible chromosome) is unknown.

Around the same time Nassersharif, Ence and Au from the University of Nevada, Las Vegas were working on another 3D visualization of a GA’s fitness landscape [Nassersharif et al., 1994]. As with Spears’ second tool, Nassersharif et al. visualized GAs solving two-dimensional problems. In this case the problem space is plotted as a three-dimensional scatterplot in which the two problem dimensions are plotted on the x and z axes, with the corresponding fitness ratings plotted on the y axis (Figure
Rather than illustrating the entire fitness surface and then highlighting the GA’s sampling of it, Nassersharif et al. used 3D scatterplot visualizations to show only the population’s sample points i.e. the population’s chromosomes without the fitness surface.

As noted in both [Spears, 1994] and [Nassersharif et al., 1994], GAs are not typically applied to one or two dimensional problems, they are more often applied to high-dimensional problems whose search space cannot be directly illustrated in two or three dimensional space. Therefore, a number of people have explored similarity metrics for illustrating the GA’s sampling of high dimensional search spaces.

In [Collins, 1993] a suggestion was made to use 2D scatterplots to illustrate the distribution of a population’s chromosomes. Each chromosome in the population can be represented by a dot in a 2D scatterplot, the coordinate of each dot indicates some problem-specific data measure, for example the chromosome’s fitness rating versus its similarity measure (such as the chromosome’s Hamming distance to the fittest). Selecting an informative similarity measure is the key to this view’s effectiveness. As noted by several of the respondents, Hamming distance is not a very effective similarity measure. It is in fact a non-unique measure (i.e. 0000 is equidistant from 1100 and 0011).

In addition to using a dot to illustrate each chromosome [Collins, 1993] also used chromosome icons to represent each chromosome, an example visualization is given in Figure 4.20.
Since the time the above visualization was first proposed, further work on GA similarity metrics has been carried out as a means for judging the problem complexity and population diversity. For example, Terry Jones and Stephanie Forrest have explored the correlation between the fitness values of all the chromosomes in a GA's run and the chromosomes' similarity to the final solution (measured either by the Hamming distance for binary chromosomes or the Euclidean distance for non-binary chromosomes). The resulting measure of problem complexity is referred to as the "fitness distance correlation" [Jones and Forrest, 1995]. Simon Ronald's work on distance functions for order-based encodings (as used for representing the traveling salesperson problem) measures the genotypic or phenotypic similarity between the chromosomes in a population. These measures are then used as a means for preserving the population's diversity during a GA's run (see [Ronald, 1995], or [Ronald, 1997] and [Ronald, 1998]).

**Contribution**

The working practices of the surveyed GA users indicated a strong interest in the GA's sampling of the search space. When asked about visualizing the rate of change in the populations' fitness values, six of the nineteen respondents indicated that they wanted to know more about the solutions considered by the GA than the fitness versus time graph could give (see Section B.3, Question 7.3). Furthermore, the respondents were strongly in favour of visualizations illustrating a similarity rating for each chromosome in the population, such as the Hamming distance to the fittest chromosome (see Section B.3, Question 8.4). The only doubts expressed were with regard to the quality of the
CHAPTER 4. REVIEW OF RELATED WORK

similarity measure used. The choice of similarity measure was generally considered to depend on the specific problem domain and representation used in the GA.

Showing a 2D or 3D visual representation of the search space enables the user to judge the diversity of the population and identify the formation of chromosome clusters. Although a similar impression can be gained from similarity measures of the population’s chromosomes, measures based on a specific search space sample (i.e. the chromosomes in a specific population or GA run) rather than the complete search space lack a consistent scale and therefore, comparisons across different populations or different runs can be difficult. However, if a consistent representation for high-dimensional problem spaces could be found then salient search space visualizations such as those proposed by [Spears, 1994] and [Nassershahif et al., 1994] could be produced for GA’s solving high-dimensional problems.

4.1.6 Navigating the GA’s Search

_Navigating a GA’s execution_

GAMeter [Kapsalis et al., 1993], Giga [Dabs and Schoof, 1995], and the Genetic Algorithm Software Development Package produced by EvoNet, are just three example systems that enable the user to “play” the GA’s run like a movie, “pause” the execution of the GA, and “step” forward a single step (i.e. one generation). Using these controls the user can pause the execution of their algorithm, make a change to the algorithm’s parameters and restart it, or step forward generation by generation in order to examine the GA’s execution.

Within the field of SV a number of systems support the bi-directional control of the program’s execution. These first appeared in systems like Henry Lieberman’s “ZSTEP” system [Lieberman, 1984], Marc Eisenstadt and Mike Brayshaw’s Transparent Prolog Machine (“TPM”) [Eisenstadt and Brayshaw, 1987], and Thomas Moher’s PR0cess Visualization and Debugging Environment (“PROVIDE”) [Moher, 1988]. Bi-directional navigational control over the program’s execution is usually achieved by periodically recording the program’s current state and then producing the visualizations using the recorded history of events. As a result, the user can navigate forwards and backwards through the program’s recorded history and the resulting visualizations will show the forwards and backwards execution of the program.
CHAPTER 4. REVIEW OF RELATED WORK

Navigating a GA’s Fitness Landscape

Another form of navigation that may prove useful within EC is the navigation of the fitness surface. Although generally used to navigate a program’s execution a similar approach could be used to identify regions of interest within the range of fitness values from a GA’s run, e.g. to identify the best chromosomes found by the GA. The navigation of the GA’s execution and the discovered regions of the fitness landscape both require immediate visual feedback.

“Dynamic Queries” [Shneiderman, 1994] incorporate the use of direct manipulation and immediate feedback to query databases. An “AlphaSlider” [Osada et al., 1993] is an example of a dynamic query interface. The AlphaSlider enables users to select an item or range of items of interest within a dataset. A range-defining alphaslider looks like a regular scroll bar, except that rather than identifying a single point in a range as a small square, the alphaslider identifies a range within a range as a bar with draggable arrow buttons at both ends. These arrow buttons define the start and end of the range of interest within an ordered data set. The rectangular bar itself can also be dragged to pan across the data set. Continuous feedback keeps the user informed of their current position within the data set.

Contribution

Within the GA user study the proposal for a bi-directional control mechanism was strongly supported (see Section B.4, Question 10.1). Using a similar approach to that commonly applied within SV, a bi-directional navigation controller could be introduced for the user to navigate the GA’s run, generation by generation. In addition to a movie-player styled controller for exploring the GA’s execution by generation, alphasliders can be used to define ranges of fitness ratings and generation numbers to be displayed. For example, an alphaslider could be used to control the displayed content of a search space visualization, displaying the top 5% of all the generations chromosomes would show the user how many good solutions the GA had considered during its run.

Within this project GA users appear to consider their algorithms in two ways; as a series of evolving generations and as a search technique for exploring problem spaces. By enabling GA users to query a GA’s execution in terms of its generation-based execution and its exploration of the problem space, both forms of understanding can be supported.