Chapter 12

D.J. Nettleton, R. Garigliano
Laboratory for Natural Language Engineering,
Department of Computer Science,
University of Durham, DH1 3LE, UK.

D.J.Nettleton@durham.ac.uk

Evolutionary Algorithms and Dialogue

12.1 Introduction

Algorithms inspired by the search processes of natural evolution have generated several robust search methods. These so-called evolutionary algorithms have been applied to a wide range of problems. This chapter discusses their application to a problem in natural language dialogue processing.

The LOLITA (Large scale, Object based, Linguistic Interactor, Translator and Analyser) natural language processor has been developed at the University of Durham over the past seven years. The aim of the development is to produce a fast system capable of operating in a wide range of domains. In order to do this, theoretical and rule based approaches are used as far as possible. These rules are then fine tuned for particular situations. However, due to the large number of rules and their complex interactions, optimisation techniques such as hill climbing are not suitable, and so far the fine tuning has been carried out by hand. This chapter examines the possibility of using evolutionary algorithms to automatically carry out the tuning of LOLITA's dialogue module to particular situations.

12.2 Methodology

Everyday intelligent beings have to respond to a range of different situations. The question, therefore, arises as to how a suitable behaviour is selected for a particular situation. One explanation would be that there are rules so completely governing possible behaviours that they cover all situations which may be encountered (a purely symbolic model). Clearly, however, while there are certainly some rules which help guide behaviour they certainly do not control it all, and simple counter examples to the above explanation of behaviour are easily constructed. Another extreme possibility would be that no rules are given, but are deduced (for future application) by interacting with intelligent beings and other objects (a purely subsymbolic model). Again this clearly is not true of human behaviour in general. More likely is it that some general rules are given and these, through learning, fine tuned to respond to certain situations (a hybrid symbolic/subsymbolic model). In effect there is an interplay between symbolic and adaptive techniques (Garigliano and Nettleton, 1994; Nettleton, 1994).

A particular example of a human behaviour, as described above, would be the holding of conversations. Throughout the day one uses a different style of conversation depending on the context, *e.g.*, chatting to a friend, giving a lecture, conducting an interview, *etc*. The use of rules such as being polite, needing to initiate the conversation, *etc.*, helps to constrain the content of the conversation. These rules, however, do not cover all eventualities, and one learns to adapt them to other contexts. Furthermore, as a conversation progresses it may be necessary to change the style of the conversation, and so further adaptation takes place. It is certainly not the case that humans learn conversational rules by interaction alone. For example, one does not learn to be polite at a job interview by being rude at others, and learning from the failures.

In developing a natural language processor able to analyse and respond to natural language input, the application of either of the above extreme methods would be unsuitable. A purely symbolic system can be produced, by specifying a large number of rules, which operates within the domain of those rules. Such systems

are usually simple, and often fail when the input is not covered by the rules. Alternatively it is possible to produce a purely subsymbolic system by exposing it to large amounts of data, and hoping that rules can be inferred. This can result in a huge amount of time and resources being expended on learning even the simplest of linguistic rules, let alone more complex ones.

The method adopted at the University of Durham in developing the LOLITA system (Garigliano *et al.* 1993a, 1993b, 1994a) has been to use a mainly symbolic approach (see Section 12.4.1). However, in the dialogue module, situations often arise in which several possible responses are available, and so the system uses a subsymbolic (integer) representation to help select between them. This involves the use of parameters to control the plan boxes which carry out responses. The tuning of these parameters so that a particular behaviour can be achieved has so far been carried out by hand. As this can be a very time consuming process, an automatic means of tuning is desirable. The search space is, however, very large and there are complex interactions between the subsymbolic components. Furthermore, cases arise in which several parameters can affect one behavioural trait (polygeny), and other cases in which a single parameter can affect several behavioural traits (pleiotropy). Search algorithms such as hill-climbing are unsuitable in such spaces.

This chapter examines the use of evolutionary algorithms in fine tuning the parameters controlling the dialogue module of LOLITA (Nettleton and Garigliano 1994a, 1994b) and demonstrates the success of a hybrid symbolic/subsymbolic approach for dialogue.

12.3 Evolutionary Algorithms

Over the past thirty years algorithms inspired by the search processes of natural evolution have been developed. These so-called evolutionary algorithms (EAs) include genetic algorithms (Holland, 1975), evolutionary programming (Fogel *et al.,* 1966) and evolution strategies (Bäck *et al.,* 1991). EAs employ a trade-off between exploration and exploitation in an attempt to find near-optimal solutions. A parallel search of the problem space is achieved by maintaining a population which consists of many different solutions. A 'survival of the fittest' strategy (similar to that used in natural selection) is employed which probabilistically culls the worst solutions. A reproductive mechanism is applied to the remaining solutions in order to produce a new set of solutions to the problem under consideration. By iterating this process, the population of solutions 'evolves' toward near-optimal solutions.

Genetic algorithms and evolutionary programming, although both inspired by the search processes of natural evolution, each place a different emphasis on what is believed to be driving the evolutionary process. Genetic algorithms model specific genotypic transformations while evolutionary programming emphasizes phenotypic adaptation. The genotype being the underlying representation used to encode a possible solution, while the phenotype is its realisation. For example, the information contained in human genes is the genotype, and the human form the corresponding phenotype.

12.3.1 Genetic Algorithms

When using genetic algorithms (GAs), solutions are usually represented as binary strings. The underlying hypothesis of GAs is that by combining subsections of solutions, short highly fit segments of each binary string are propagated throughout the population, and combine to form larger fitter segments of each binary string. This is known as the building block hypothesis (Goldberg, 1989), and is a fundamental principle of GAs. In order to allow the transmission of sections of binary string, 'child' solutions are produced by combining the binary strings of two 'parent' solutions. By ensuring that the fitter solutions are involved in the reproduction of more child solutions, short fit sections of the binary string spread throughout a population. In order to ensure that no piece of binary string can be lost from a population a mutation operator is used. This typically has a very small probability of application, since otherwise it would be highly disruptive.

The following is an outline of the genetic algorithm used:

1) Randomly initialise the parent population of binary strings.

2) Evaluate each member of the parent population.

3) Select a solution from the parent population with probability in proportion to fitness.

4) Apply the crossover operator with a probability pc. If crossover is not performed then place solution into child generation.

Otherwise:
   (a) Select a solution from the parent population with uniform probability.
   (b) Select at random two crossover points that are within the binary string.
   (c) Recombine the solutions (splicing the respective sections from each string with each other), and place them both into the child generation.

5) If the child generation is not full then go to step 3.

6) With a probability pm, mutate elements of the binary string of each of the child solutions.

7) Replace the parent population with the child population.

8) If termination criteria is not met go to step 2.

There are many variations of the above algorithm. For example, many alternative crossover mechanisms and selection methods have been suggested. Further details of the implementation of the GA are given in Section 12.8.

12.3.2 Evolutionary Programming

The evolutionary programming (EP) perspective of the evolutionary process is very different from the bottom up approach of GAs. By determining how well solutions are performing in the current environment, improvements are made via a flow of information from the environment back to the underlying genotypic representation. The emphasis is, therefore, on phenotypic adaptation rather than genotypic transformation. In this way a top-down approach to solution improvement is adopted as opposed to the bottom-up approach of GAs.

The form of solution representation used when using EP usually varies from problem to problem. Often the most convenient for the problem under consideration is used, for example, floating-point numbers or integers. In order to create new solutions to the problem a mutation operator acts on the current set of solutions with each solution being mutated to produce one solution. The exact form of the mutation operator is dependent on the representation, but the degree to which a solution is mutated is related to the solution's fitness. Fitter solutions being less likely to be mutated to the same degree as less fit parents.

The following is an outline of the evolutionary program used.

1) Randomly initialise the parent population.

2) Evaluate each member of the parent population.

3) Mutate each member of the parent population, by an amount related to its fitness, to give a member of the child population.

4) Evaluate each member of the child population.

5) For each member of the child and parent populations:
   (a) Select at random a number, TOURN, of solutions from the parent and child populations.
   (b) Count the number of these solutions whose fitness is less than or equal to that of the current selected solution. This number is the 'score' for the selected solution.

6) Order the scores of the solutions.

7) Select the solutions whose score is in the top half of the list and replace the parent population with these solutions.

8) If termination criteria is not met go to step 3.

Again there are many variations on the above algorithm including, for example, meta-EP (Fogel, 1992). Further details of the implementation of EP are given in Section 12.8.

12.4 Natural Language Processing

This section together with section 12.5 discuss in some detail the problem to which EAs are to be applied. The details of the dialogue theory used are included so that its power may be better appreciated.

Natural language processing (NLP) lies at the intersection of disciplines such as artificial intelligence, linguistics and cognitive science. A successful natural language processor must be able to automatically process, understand and generate sections of natural language. Much work in the field of NLP has concentrated on 1) implementing a linguistic theory to show that it can account for the features which it describes (computational linguistics) and 2) the modelling of the human thought process by a computer (cognitive science).

Although these are of much interest, such systems are often so specialised, or so cumbersome, that they cannot be exploited in any practical way. In recent years, however, a more practical approach to NLP has emerged in the form of Natural Language Engineering (NLE), indeed a journal has recently been launched dedicated to this (Garigliano *et al.* 1994b). The paradigm of NLE is the development of systems which are general enough, and quick enough to be of practical use. Such a paradigm takes into account features such as scale, integration, flexibility, feasibility, maintainability, robustness and usability (Smith *et al.* 1994). NLE adopts a pragmatic approach to achieving these goals which is characterised by a readiness to use any means in order to build serious speech and language processing programs.

12.4.1 The LOLITA System

LOLITA is an example of a system created using a NLE methodology (Garigliano *et al.,* 1993a, 1993b, 1994a). LOLITA is built around a large semantic network of some 60,000 nodes (capable of over 100,000 inflected word forms) which contain data and world information. The system can parse text, semantically and pragmatically analyse its meaning and alter the relevant information in the semantic network. Information contained within the semantic network can be generated in the form of natural language (Smith *et al.,* 1994), and so a 'natural' interaction with the system is possible. Having being developed using an NLE methodology the system is very general. Recently the underlying system has been used (with little in the way of modification) as the base for a variety of prototype applications. These include an Italian to English translator, contents scanning of newspaper articles, Chinese tutoring, and dialogue analysis and generation.

The LOLITA system incorporates several logical and linguistic theories in its general construction. However, in dealing with specific areas these theories are often not strong enough, and so more localised theories are used. Even when these localised theories are impractical (*e.g.*, for efficiency reasons) the LOLITA system resorts to a knowledge based approach or uses heuristics to solve problems. By incorporating such a range of approaches LOLITA is able to enjoy the advantages provided by a well constructed general theory. At the same time LOLITA is flexible enough to use other approaches should these theories fail for particular problems.

12.5 Dialogue in LOLITA

This section discusses the theory of dialogue which is used within the LOLITA system. An account of the theory is given so that its power can be appreciated. First of all, however, definitions are given of some terms which may otherwise be open to various interpretations.

The terms dialogue and discourse are usually used loosely by many workers in the field. The definitions which are used in this chapter are those given by Jones and Garigliano (1993). Discourse is taken to mean a set of sentences which are related to each other both linguistically and contextually. Such a definition includes newspaper articles, but an interaction between participants is not a requirement for a discourse. Dialogue is taken to be the rich interaction between two or more participants, where 'rich interaction' is taken to include features such as sub-dialogues, interruptions and complex shifts in focus.

Theories of dialogue can be broadly classified as: descriptive, prescriptive, predictive and inferential. A descriptive theory is simply aimed at being able to describe a known piece of dialogue in terms of some set of features. The other types of theory are more useful since these can be used (with varying degrees of power) to provide information on what is to happen next in the dialogue. In a general natural language processor once a piece of text has been analysed the system needs to prepare a response. Rather than simply responding with the same style of text for all situations, LOLITA is capable of producing a wide range of styles. A theory of dialogue capable of providing information on a suitable response is required. Such a theory has been developed over the past three years (Jones and Garigliano, 1993; Jones, 1994).

12.5.1 Dialogue Situations

In many situations in which humans find themselves, the type of dialogue structure that can be expected for that particular situation is known. The knowledge required to determine this has been acquired through a mixture of given rules and learning (Section 12.2). In order to take advantage of this knowledge Schank and Abelson (1977) introduced the idea of scripts. A script is described by Schank and Abelson (1977, p. 41) as "... a structure that describes appropriate sequences of events in a particular context ... a predetermined, stereotyped sequence of actions that defines a well-known situation." An example of a script would be the dialogue between a waiter and customer in a restaurant. In such a situation both participants can be considered to be filling in the slots of some pre-determined template which has slots for actions such as ordering food.

Scripts are used to describe events from the physical world. The theory of dialogue incorporated in LOLITA is aimed at modelling the actual structure of the dialogue. This theory is based on the concept of a Dialogue Structure Model (DSM), and is now described (Jones, 1994).

A DSM is a schema which contains all of the information that can be expected to be relevant in a particular situation, and thus can be used to guide the generation of language to suit that situation. The DSM consists of dialogue elements, which are factors that influence and control the structure of the dialogue. In a lecture, for example, the lecturer can be expected to be in control of the dialogue,

and to speak for most of the lecture's allotted time. Factors such as these determine the basic information required for a class of similar situations. Furthermore, a theory of dialogue based on DSMs is not simply descriptive, for a DSM can prescribe the manner in which the remainder of the dialogue is to be carried out.

12.5.2 Dialogue Elements

The Dialogue Elements (DEs) are the fundamental components of a DSM, and the current set can be subdivided as follows.

External Elements — These are elements which are external to the language itself. Although they are not part of the dialogue they influence its structure.

- **Number** — The number of participants involved in the dialogue.

- **Time Limit** — Whether or not there is a specific limit on the amount of time available within which the dialogue must be completed. Whether or not the dialogue must terminate by a particular time.

- **Temporal Progression** — The stages through which the dialogue progresses as time passes. For example, in a lecture one can expect an introduction, a main body and a conclusion. In a chat, however, there is far less structure.

Motivational Elements — All dialogues are started for some purpose, whether it be to simply pass the time of day or conduct an interview. The elements discussed below are connected to the purposes for which a dialogue is being held, and are linked to the goals, motivations and intentions of the participants in the dialogue. Since a dialogue always has a motive a DSM must always contain a motivational dialogue element.

- **Emotional Exchange** — Whether or not any of the dialogue's participants aim to change the emotional state of another participant. For example, make them laugh, cry or indifferent.

- **Goal** — This is divided into 'task' and 'process' and relates to the aim of the dialogue. If the aim is that of a task, then the goal is used to specify some end result, *e.g.*, verbal instructions for the assembly of a piece of machinery. Process goals are achieved in stages as the dialogue progresses, *e.g.*, a lecture conveys information on some topic as it unfolds.

- **Information Seeking** — Whether or not any of the dialogue's participants aim to gain information during the dialogue.

- **Persuasive** — Whether or not the aim of any of the dialogue's participants is to cause another participant to believe in the truth of some statement.

Verbal Elements — These are verbal properties of a dialogue, and may or may not be present within the dialogue.

• **Colour** — This relates to the style of language, *e.g.*, use of adjectives, figures of speech, analogies, *etc*.

• **Distribution of Time** — The amount of speaking time that each participant is allowed within the dialogue. In a lecture, for example, the students can be expected to speak far less than the lecturer.

• **Dominance** — Determines the degree of control a participant has on the structure of dialogue, content or direction.

• **Fixed Topic** — Whether the dialogue is constrained to be on one topic or whether the dialogue can cover several topics.

• **Length** — The length of sentences contained within the dialogue, *e.g.*, long or short.

• **Register** — This relates to the kind of vocabulary that is in use within the dialogue, *e.g.*, formal, informal, slang, *etc*.

• **Rhythm** — The rhythm of the dialogue. If, for example, it is to progress in short bursts or long flowing constructions.

All dialogues have some form of structure that is external to the situation or participants. For example, all lectures can be expected to have a fixed timespan. In the case of such a dialogue in a particular situation, the relationship, individuality and character of the participants all play an important role in the development of the dialogue. Furthermore, an individual's state of mind at a particular time (*e.g.*, happy, sad) is important in determining how the dialogue progresses. It is through DSMs and DEs that the LOLITA system models these parts of human behaviour.

12.5.3 Constraints and Plan Boxes

Although the situation, character, *etc.*, allows humans to place many constraints on the responses which may be made in some situation, there are still many possibilities. The process of selecting an appropriate response is one which humans take for granted. LOLITA like a human is capable of many responses, and therefore needs some mechanism by which responses can be selected. Once a response has been selected plan boxes are used to inform on how and when the output is generated. There are currently some 124 plan boxes contained within LOLITA, and some means of selecting a plan box from the many possibilities is required.

LOLITA is able to reduce the number of possibilities via inference and heuristics. Inference on the input is used to examine its emotional and intellectual value. Heuristics are then used to ensure that certain plan boxes are not triggered. For example, if LOLITA is forced not to be rude then blocks of plan boxes that would result in a rude response are excluded. Once these processes have been performed the LOLITA system is usually left with some 10–15 plan boxes which correspond to different outputs. Some mechanism for determining how likely a

certain response is for a particular situation is needed. For example, one may not wish to answer a question, and possible responses could involve replying with a question or simply saying 'I don't want to talk about that'.

The problem that remains is how to order the possibilities, dependent on the behaviour which is being sought. If, for example, the current DSM dictates that dialogue participant X has a greater level of dominance than participant Y, it is possible for X to terminate the dialogue. Although the termination of the dialogue is permitted it may not be appropriate at particular points of a dialogue — a lecturer has greater dominance in a lecture, but would not be expected to terminate the dialogue half way through without adequate explanation. So although 'terminate dialogue' is an option it would be inappropriate and must be marked as such. In general no clear rules are available for ranking, and so a subsymbolic approach is adopted. This involves attaching a parameter (an integer) to each plan box to indicate how permissible an action is. Then in selecting a plan box (of those allowed) with which to generate a response, the plan box with the lowest value is used.

It is worth noting that it is not the absolute values of the parameters that is important, but their relative values. Furthermore, as a dialogue progresses the values of the parameters attached to plan boxes vary to take into account the dialogue to that point. For example, if one participant of a dialogue, X, continually annoys another, Y, then Y's terminate dialogue option can be expected to become more likely as the dialogue progresses.

12.6 Tuning the Parameters

The parameters that control the plan boxes contained within the dialogue module of LOLITA have been fine tuned by hand to give a particular behaviour. The 124 plan boxes each have a single parameter attached which may be altered, and so the search space of possible settings is very large. With experience it is possible to acquire an intuitive feel for the optimisation, but this process is very time consuming, and an automatic means of tuning is required.

A method by which the usefulness of a tuning algorithm could be determined would involve first of all tuning the parameters to give a particular series of utterances. Then starting with a random set of parameters, use the tuning algorithm to try and optimise the parameters so that the same series of utterances is generated. Results achieved when adopting such a method are presented later in this chapter. This method has the following advantages:

1) An optimum solution is known to exist for that situation, *i.e.*, the one obtained by hand.

2) A large number of previously untested parameter combinations will be generated, and running LOLITA with these parameters will be a good test of the robustness of the LOLITA system.

The main disadvantage of the above method is that even if the tuning algorithm is successful in the particular situation studied, there is no guarantee that it would be more widely applicable. If it were possible for the algorithms to optimise for

a predetermined behaviour, then it would be possible to find parameter settings that resulted in different styles of dialogue, *e.g.*, polite or rude. The later part of this chapter begins to examine the use of EAs in this wider role.

12.7 Target Dialogues

The dialogues given in Tables 12.1 and 12.2 are the target dialogues for the optimisation process. DIAL 1 (Table 12.1) is generated through a single interaction with the LOLITA system for which the parameters were tuned. DIAL 2 (Table 12.2) is made up of statements that it is known the LOLITA system can produce, and were selected from dialogues which had several different parameter settings. In the case of DIAL 2 it is not known if a single parameter setting exists that can produce this entire dialogue.

Table 12.1: The target dialogue DIAL 1 which was produced in a single interaction with LOLITA.

DIAL 1	LOLITA 1:	I desire a success.
	INPUT 1:	I like you.
	LOLITA 2:	I know; why do you like me?
	INPUT 2:	You are beautiful.
	LOLITA 3:	I am glad because you think I am beautiful.
	INPUT 3:	Do you like me?
	LOLITA 4:	Yes, I like you.
	INPUT 4:	Do you like me?
	LOLITA 5:	I slightly am bored and emotional.
	INPUT 5:	You are a machine.
	LOLITA 6:	I do not believe that I am a machine.
	INPUT 6:	Are you a machine?
	LOLITA 7:	No.
	INPUT 7:	Are you beautiful?
	LOLITA 8:	I do not know if I am beautiful.
	INPUT 8:	I am tired.
	LOLITA 9:	Tell me things.
	INPUT 9:	I am tired.
	LOLITA 10:	I slightly am bored.
	INPUT 10:	Are you tired?
	LOLITA 11:	No.

Table 12.2: The target dialogue, DIAL 2, which is a collection of utterances from different interactions.

DIAL 2	LOLITA 1:	I desire a success.
	INPUT 1:	I like you.
	LOLITA 2:	Why do you like me?
	INPUT 2:	You are beautiful.
	LOLITA 3:	I am glad because you think I am beautiful.
	INPUT 3:	Do you like me?
	LOLITA 4:	Yes, I like you.
	INPUT 4:	Why do you like me?
	LOLITA 5:	I do not know why I like you.

INPUT 5:	You are a machine.
LOLITA 6:	I do not believe that I am a machine.
INPUT 6:	Are you a machine?
LOLITA 7:	No.
INPUT 7:	I am tired.
LOLITA 8:	I understand.
INPUT 8:	I am tired.
LOLITA 9:	I slightly am bored.
INPUT 9:	Are you tired.
LOLITA 10:	No.
INPUT 10:	Tell me if you believe that I am tired.
LOLITA 11:	Yes, you are tired.

12.8 Application of EAs to LOLITA

The environment is the source of information on which solutions are evaluated. In theory this must correspond to all natural language utterances. In practice, however, the current implementation of LOLITA acts only on literal meaning. Metaphors, idioms and humour are, therefore, excluded from the environment.

In deciding which plan boxes are to be activated, it is not their absolute values that are important, but rather their values relative to each other. Therefore, it is not the explicit values of the parameters that are to be optimised, but a shift in value from that of the current hand optimised setting. For each plan box a range of shift values (simply referred to as parameter values from now on) of [-63,64] was deemed sufficient, since these allow for a large range of possible behaviours (if necessary this range can easily be increased). A solution's representation is, therefore, a string of 124 (the number of plan boxes) integers. A solution with all of its values set to 0 is, therefore, identical to the current hand optimised setting. The parameters of the plan boxes which control utterances of a particular type are grouped together in blocks. For example, the three plan boxes labelled `cause_Affection Platonic` are grouped together, as are the six which are labelled `show_AngerOffense`. The components within a block determines how an utterance is carried out, *e.g.*, different ways in which anger can be expressed.

As mentioned previously some measure of how closely utterances generated match those of the target dialogue is needed. The results given in the next section use a very simple fitness function. A solution's fitness is initially set at zero, and then increased by one for each utterance that exactly matches that in the target dialogue. For the target dialogues discussed in this chapter a solution's fitness is, therefore, an integer in the range [1,11]. The total number of utterances that LOLITA generates is eleven, and so this provides the upper bound on fitness. Furthermore, all solutions will have a fitness of at least one, since with the current 'personality' LOLITA always initiates a conversation with the phrase 'I desire a success'. A more sophisticated fitness function is introduced in Section 12.10.

Comparing the results of runs of a GA and EP is difficult since the underlying system is continually changing and the data files regularly updated. Only single

trials of each algorithm are carried out, but these are sufficient to show the validity of the approach.

Further details of the GA and EP implementation are now discussed.

12.8.1 Genetic Algorithms

As mentioned in Section 12.3.1 when using a GA solutions are to be represented as binary strings. The parameters controlling the plan boxes can take one of 128 distinct values, and so each parameter can be converted to a binary string of length seven. These strings are then concatenated together to form one string. Since there are some 124 plan box parameters the size of the search space is $2^{(7 \text{ x } 124)} \approx 10^{251}$.

When selecting solutions for mating, a 'roulette wheel' type of sampling is used in order to ensure that better solutions are more likely to be chosen (Goldberg, 1989). This proceeds by first evaluating the fitness of each solution in a generation. Sections of the roulette wheel are then allocated according to this fitness value. This ensures that when the roulette wheel is probabilistically spun, the fitter the solution the more likely it is to be selected. With the fitness function used all solutions will have a fitness of at least one, and so all are guaranteed a section of the roulette wheel.

Parents are combined using a two-point crossover operator (Beasley *et al.,* 1993) with the probability of crossover pc = 0.6. When applied to a point in the binary string the mutation operator changes the value at that point, *i.e.*, 1 to 0, or 0 to 1. In order not to be too disruptive the probability of mutation was kept low with pm = 0.001.

12.8.2 Evolutionary Programming

In applying EP to the dialogue optimisation problem the plan box parameters are stored as integers which are constrained to be in the range [-63,64]. In practice it isn't necessary to restrict the range, but this was done in order to ensure the search space was the same size for EP as for the GA.

Each parent solution in the population is mutated by an amount governed by its fitness to produce a child solution. Fitter solutions must be less likely to be mutated to the same degree as less fit parents, and so each component, xi, of a solution X, is mutated according to the formula (and then truncating):

$$x_i' = x_i + \sqrt{5 \cdot \left(\text{MAXFIT} - \mathit{fitness}(X)\right)} \cdot N(0,1) \qquad i \in \{1,2,\ldots,124\}$$

where MAXFIT is the maximum fitness attainable (11 for the work discussed in this and the following section), fitness(X) is the fitness of solution X (the number of correct utterances) and N(0,1) is a standard normal random variable. The above formula was selected since it allows for solutions with a poor fitness to be mutated by a large amount, while at the same time reducing the chance that the mutated parameters fall outside of the permitted range.

12.9 Results

This section presents the results of applying a GA and EP to the problem of finding plan box parameters.

For both the GA and EP a population of 50 was used and they were executed for 50 generations. The tournament size for EP was set at three. A single trial of each algorithm was carried out. Figures 12.1 and 12.2 show the online and offline performance of the GA and EP run, for the target dialogues DIAL 1 and DIAL 2, respectively. The offline performance is the average fitness of all of the solutions in a particular generation, while the online performance is the average fitness of all solutions that have been generated up to a certain generation.

In the case of DIAL 1 the GA was able to find a set of parameters which produced a dialogue of fitness 9, *i.e.*, two utterances incorrect. EP performed slightly better, discovering a solution of fitness 10. When DIAL 2 was used as the target dialogue the GA was able to find a solution of fitness 8, and EP a solution with fitness 9. These results are summarised in Table 12.3.

In the case of EP the incorrect utterance for DIAL 1 was "LOLITA 8: I do not know if I am beautiful; tell things to me." Such an utterance should not be considered as wrong, it is simply that the fitness function is not very sophisticated. Similarly for the GA and DIAL 1. For both the GA and EP, with DIAL 2 as the target dialogue, the incorrect utterances for the best parameters found indicate that the parameter settings were such that the input caused LOLITA to become offended quite easily.

DIAL	LOLITA's incorrect utterances	
1	GA	2: Tell me things. 6: I slightly am bored and emotional
	EP	8: I do not know if I am beautiful; tell things to me.
2	GA	2: I could not speak to you if you repeated you like me. 6: I desire to success. 7: I could not speak to you if you repeated Am I a machine?
	EP	2: I know; I could not speak to you if you repeated you like me. 7: I could not speak to you if you repeated Am I a machine?

Table 12.3: The incorrect utterances generated by the best parameters found when GA and EP were used to optimise the plan box parameters for DIAL 1 and DIAL 2.

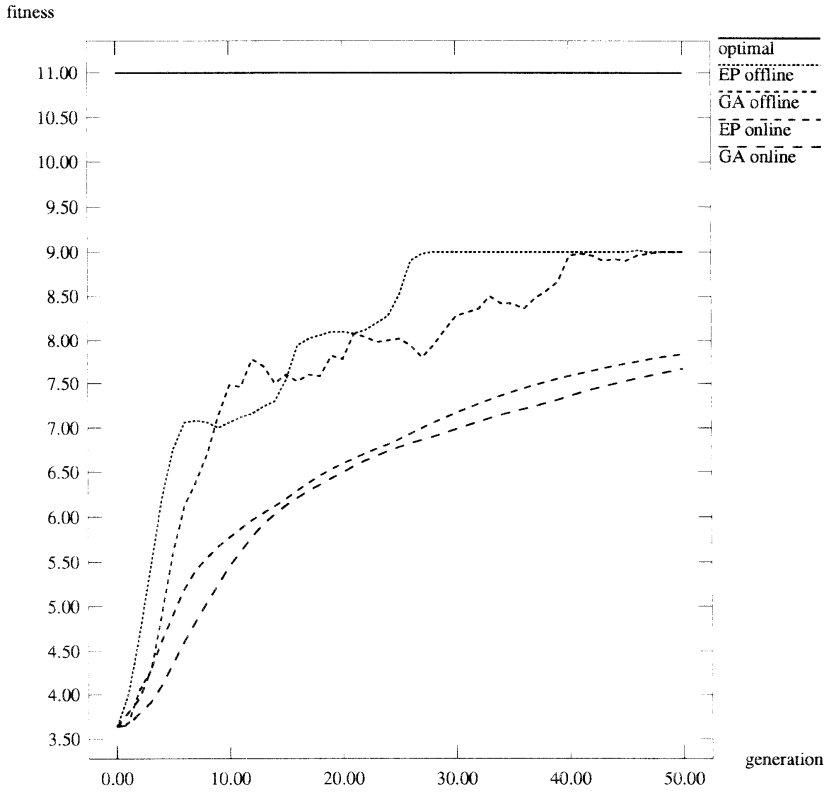

Figure 12.1: Online and offline performance for a trial of the GA and EP with DIAL 1 as the target dialogue.

An interesting feature of the EP results is how the average fitness of a generation rose to that of the best solution to date (Figures 12.1 and 12.2). It appears that when a better solution was discovered the average generation fitness would rise gradually for several generations and then quickly rise to that of the best. There is, however, one notable exception to this which occurred at generation 46 when DIAL 1 was the target dialogue (see Figure 12.1). At this point a solution of fitness 10 was produced in a population the remainder of which had fitness 9. The solution of fitness 10 was, however, subsequently lost and the reason for this is now discussed. Although a solution with fitness 10 is guaranteed a score of three in the tournament, many other solutions in that population also scored a fitness of three since all but one solution against which they were competing had a fitness of 9. When the process of sorting the scores took place there were more solutions with a score of three than places for them in the next generation and so some were lost. This included the solution of fitness 10. A similar occurrence took place in the run with DIAL 2. A solution of fitness 9 was discovered at generation 25, retained for one generation, and then lost.

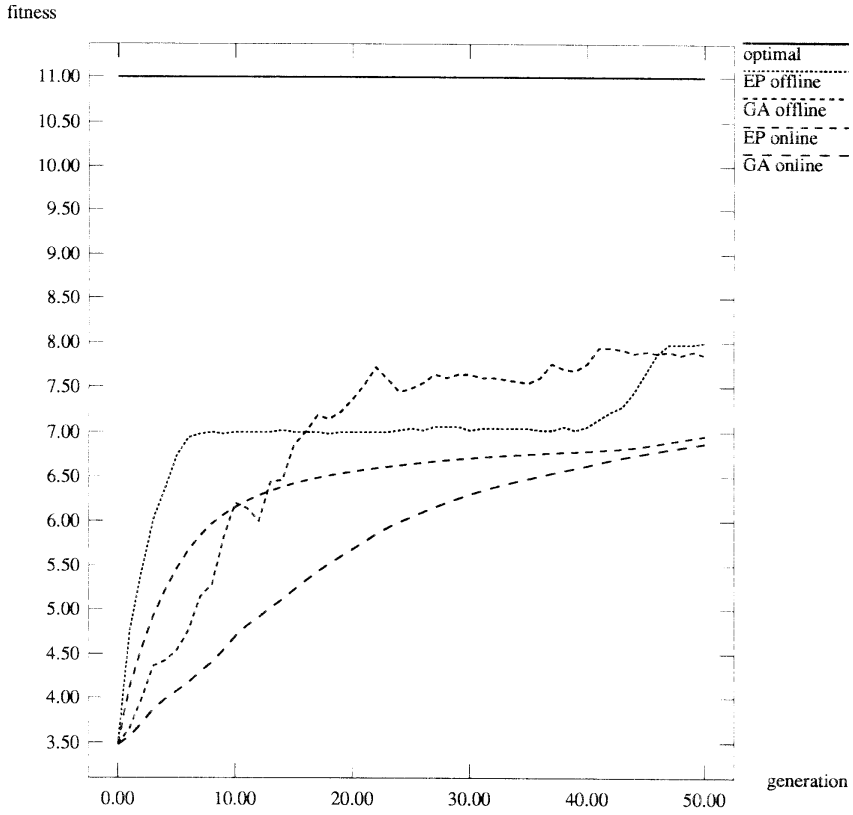

Figure 12.2: Online and offline performance for a trial of the GA and EP with DIAL 2 as the target dialogue.

The failure to retain an improved solution is in part attributable to the poor discriminatory power of the fitness function used. Since many solutions can have the same fitness a lot of solutions often perform very well in the tournament, and solutions with a maximum tournament score may be lost from the following generation. The following section examines a fitness function which is able to use additional information which the LOLITA system is able to provide. This improves the fitness function's discriminatory power, rewarding not just the words produced, but the underlying actions which lead to their generation.

12.10 Improving the Fitness Function

The fitness function adopted in the previous section is very simple and unable to take into account additional information which the LOLITA system is able to provide. On analysing an utterance the LOLITA system infers information on the local goals, subgoals, utterance types and action types of the speaker. Table 12.4 shows this information for the first seven utterances of DIAL 1.

LOLITA 1: (I desire a success.)	speaker: lolita local goal: ShowEmotionGoal, NeutralEmotion subgoals: utterance types: AllSame action types: default_tacticPB
INPUT 1: (I like you.)	speaker: roberto local goal: InformGoal subgoals: utterance types: Statement action types:
LOLITA 2: (I know; why do you like me?)	speaker: lolita local goal: InformGoal subgoals: AnyGoal, BeInformedGoal utterance types: Statement, Noise, Question action types: tellPB, why_questPB
INPUT 2: (You are beautiful.)	speaker: roberto local goal: InformGoal subgoals: utterance types: Statement action types:
LOLITA 3: (I am glad because you think I am beautiful.)	speaker: lolita local goal: ShowEmotionGoal Serenity subgoals: utterance types: AllSame action types: show_Serenity
INPUT 3: (Do you like me?)	speaker: roberto local goal: BeInformedGoal subgoals: utterance types: Question action types:
LOLITA 4: (Yes, I like you.)	speaker: lolita local goal: InformGoal subgoals: utterance types: Statement action types: answerPB

Table 12.4: The additional information which the LOLITA system makes available for the first seven utterances of DIAL 1.

The fitness function can be modified to make use of the additional information given in Table 12.4. The fitness function used in this section calculates a solution's fitness by initially setting it to zero, and increasing it by one for each utterance, local goal, subgoal, utterance type and action type, which exactly

matches that of the target dialogue. This fitness function is less sensitive to the utterance itself and more sensitive to the behaviour required. For DIAL 2 the information associated with each statement was used.

Using this additional information a fitness, which is an integer in the range [5,55], can now be assigned to solutions — 5 forms the lower bound since LOLITA always initiates a conversation with the same utterance and associated information. For each of the two target dialogues a single trial of the GA and EP were carried out. The GA and EP used the improved fitness function and in addition two modifications were made to EP. In the tournament phase of the algorithm if two solutions have the same fitness then a win is awarded with probability 0.5. This modification is aimed at helping to overcome the problem of EP 'loosing' a solution which arose in the experiments with first of the fitness functions discussed. Secondly, the EP's mutation operator is altered so that a child is produced from a parent by mutating each parameter xi as follows (and then truncating):

$$x_i' = x_i + \sqrt{\left(\text{MAXFIT} - \mathit{fitness}(X)\right)} \cdot N(0,1) \qquad i \in \{1,2,\ldots,124\}$$

where MAXFIT is the maximum fitness attainable (55 for the work discussed in this section), fitness(X) is the fitness of solution X and N(0,1) is a standard normal random variable.

Figures 12.3 and 12.4 show the online and offline performance of the GA and EP run, for the target dialogues DIAL 1 and DIAL 2, respectively.

In the case of DIAL 1 the GA was able to find a solution with a fitness of 47 by generation 15, and EP a solution of fitness 47 by generation 8. For DIAL 2 the GA discovered a solution of fitness 43 by generation 7, and EP a solution of fitness 43 by generation 14. The breakdown of these results is shown in Table 12.5.

Additional information	DIAL 1		DIAL 2	
	GA	EP	GA	EP
utterance	9	9	8	7
local goal	10	10	8	9
subgoals	11	11	11	10
utterance types	9	9	8	9
action types	8	8	8	8
Fitness	47	47	43	43

Table 12.5: Decomposition of the results achieved with the improved fitness function. The optimum value for each of the values is 1.

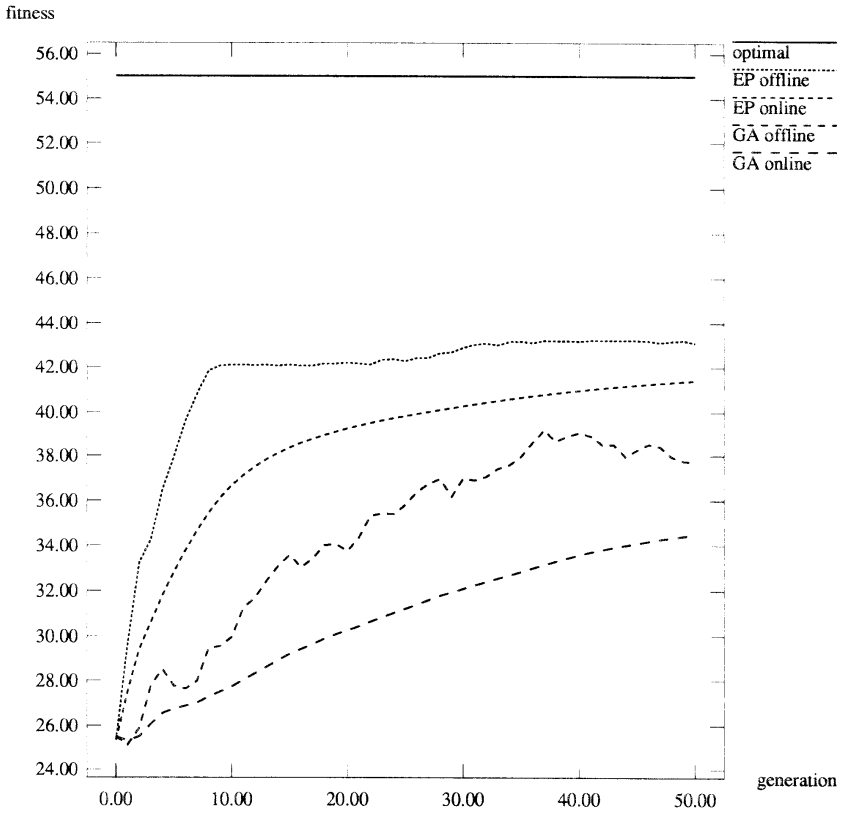

Figure 12.3: Online and offline performance for a trial of the GA and EP with DIAL 1 as the target dialogue. The fitness function which takes into account LOLITA's additional information was used.

Again the exact matching of utterances resulted in statements such as "LOLITA: Why do you like me?" in place of "LOLITA: I know; why do you like me?" being scored as incorrect. Similar instances arose with the matching of the additional information. For example, if the utterance types are "Statement, Noise, Question" then "Statement, Question" is currently scored as incorrect. A fitness value of 0.666 would be more appropriate. There is clearly much scope for improvement in the discriminatory power of the fitness function.

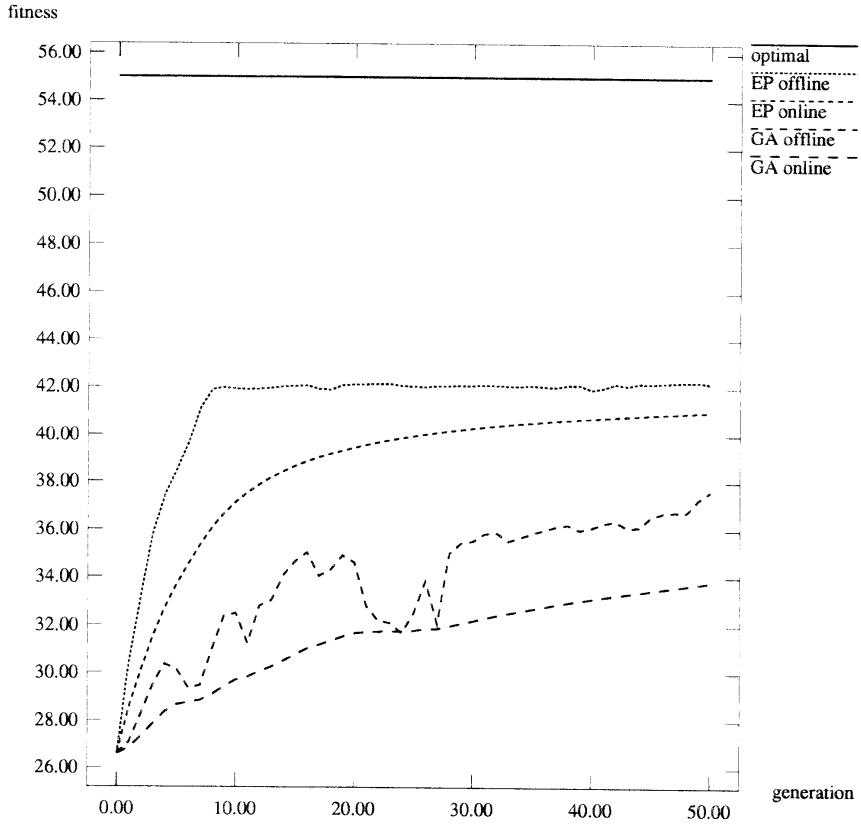

Figure 12.4: Online and offline performance for a trial of the GA and EP with DIAL 2 as the target dialogue. The fitness function which takes into account LOLITA's additional information was used.

12.11 Discussion

The results show that both a GA and EP were reasonably successful at the dialogue optimisation problem presented. These results, although preliminary, do lead to some interesting points worthy of further consideration.

For both the GA and EP the average fitness of solutions in subsequent generations steadily improved. No attempt was made to tune the settings of the evolutionary algorithms themselves. In the case of the GA such settings include the crossover and mutation probabilities. Other components of the GA that may be altered include the solution representation (*e.g.*, integers), crossover type and selection mechanism. The performance of EP may be improved by altering the tournament size, or the formula controlling the amount of mutation. Furthermore, it is likely that by increasing the population and generation size improved results can be expected. This has not been studied to date since with a population and generation size of 100, the runtime (on a Sparc4 workstation) can be expected to be of the order of two days. Furthermore, evaluating any differences in performance is difficult since the underlying system is continually being modified.

For the dialogues and fitness functions considered the fact that both a GA and EP are able to discover solutions which perform well indicates that both a bottom-up and a top-down approach is a suitable means of solution construction.

The discriminatory power of the fitness function needs to be further improved. Ideally some quantitative measure of semantic distance would be used (Short *et al.,* 1994a, 1994b). This would entail finding some quantitative measure for the similarity of the meaning of two sentences. Another approach would involve better use of the information that the LOLITA system is capable of producing. With an improved fitness function the current limitation of having to apply the EAs to known dialogues can be removed. Evolving the plan box parameters so that the resulting dialogue exhibits a certain personality is the long term aim, *e.g.*, finding the parameters which result in LOLITA becoming easily offended. Once sets of parameters for different behaviours have been determined they can be used to run LOLITA with that 'personality'.

12.12 Summary

This chapter provides evidence that a hybrid symbolic/subsymbolic approach can be successfully applied within the dialogue module of a large scale natural language processor. Adopting such an approach allows the dialogue module to enjoy many of the advantages of a well constructed theory, while at the same time allowing for the flexibility which a subsymbolic approach is capable of providing. The complex dialogues which can be generated validate the approach.

Evolutionary algorithms have been applied to the problem of searching the space of the subsymbolic representation so that a solution which exhibits a certain behaviour can be found. For the dialogues and fitness functions considered both a GA and EP were able to overcome the interactions which may occur and construct solutions that perform well. A more general application of the approach is currently limited by the poor discriminatory power of the fitness function.

References

Bäck T., Hoffmeister F. and Schwefel H. (1991) A Survey of Evolution Strategies, in Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, pp 2–9.

Beasley D., Bull D.R. and Martin R.R. (1993) An Overview of Genetic Algorithms: Part 2, Research Topics, University Computing, Vol. 15, No. 4, pp 170–181.

Fogel D.B. (1992) Evolving Artificial Intelligence, Ph.D. Thesis, University of California, San Diego.

Fogel L.J., Owens A.J. and Walsh M.J. (1966) Artificial Intelligence through Simulated Evolution, J. Wiley, New York.

Garigliano R. and Nettleton D.J. (1994) The Interplay of Symbolic and Adaptive Techniques: two Case Studies, IEE Colloquium on Symbolic and Neural Cognitive Engineering, Savoy Place, London.

Garigliano R., Morgan R.G. and Smith M.H. (1993a) The LOLITA System as a Contents Scanning Tool, in Proceedings of the Thirteenth International Conference on Artificial Intelligence, Avignon.

Garigliano R., Morgan R.G. and Smith M.H. (1993b) LOLITA: Progress Report 1, Technical Report 12/92, Department of Computer Science, University of Durham, U.K.

Garigliano R., Morgan R.G. and LOLITA group (1994a) The LOLITA Project: The First Seven Years, under negotiation with After Hurst Ltd.

Garigliano R., Tate J. and Boguraev B. (eds.) (1994b) Journal of Natural Language Engineering, Cambridge University Press.

Goldberg D.E. (1989) Genetic algorithms in search, optimization, and machine learning, Addison-Wesley.

Holland J.H. (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press.

Jones C.E. (1994) Dialogue Structure Models: An approach to Dialogue Analysis and Generation by Computer, Ph.D. Thesis (submitted), Department of Computer Science, University of Durham, U.K.

Jones C.E. and Garigliano R. (1993) Dialogue Analysis and Generation: A Theory for Modelling Natural English Dialogue, in Proceedings of EUROSPEECH '93, the 3rd European Conference on Speech Communication and Technology, Berlin, pp 951-954.

Nettleton D.J. (1994) Evolutionary Algorithms in Artificial Intelligence: A Comparative Study Through Applications, Ph.D. thesis (submitted), University of Durham, U.K.

Nettleton D.J. and Garigliano R. (1994a) Evolutionary Algorithms for Dialogue Optimisation in the LOLITA Natural Language Processor, Seminar on Adaptive Computing and Information Processing, London.

Nettleton D.J. and Garigliano R. (1994b) Evolutionary algorithms for dialogue optimisation as an example of hybrid NLP system, International Conference on New Methods in Language Processing, Manchester.

Schank R.C. and Abelson R.P. (1977) Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates Inc., New Jersey.

Short S., Collingham R.J. and Garigliano R. (1994a) What did I say...? Using Meaning to Assess Speech Recognisers, Institute of Acoustics Autumn Conference on Speech and Hearing, Windermere, Cumbria, U.K.

Short S., Collingham R.J. and Garigliano R. (1994b) Making Use of Semantics in an Automatic Speech Recognition Systems, Institute of Acoustics Autumn Conference on Speech and Hearing, Windermere, Cumbria, U.K.

Smith M.H., Garigliano R. and Morgan R.G. (1994) Generation in the LOLITA system: An engineering approach, Seventh International Workshop on Natural Language Generation, Maine, U.S.A.