Local Halting Criteria for Stochastic Diffusion Search Using Nature-inspired Quorum Sensing — Chapter 2: History of AI

Andrew Owen Martin

a.martin@gold.ac.uk, andrew@aomartin.co.uk

Thesis submitted for the degree of Doctor of Philosophy

of the

University of London

Department of Computing

Goldsmiths College

April 2020

Chapter 2

History of AI

It is only a question of cards and time.

— Major-General H. P. Babbage (1888) [2]

2.1 Historical background

In this section we chart development of diversity in a machine's behaviour, from Clepsydra which can perform a single behaviour, through machines which include configurable mechanisms to determine a specific routine, to computers which are hugely diverse due to defining algorithms in the same manner as their input data. Though people have been questioning the potential of machine behaviour at least as early as 1842 when Ada Countess of Lovelace wrote "[the Analytical Engine] might compose elaborate and scientific pieces of music of any degree of complexity or extent", the modern study of achieving intelligent behaviour in machines was greatly influenced by Alan Turing's pioneering work following his definition of a model of computing, and subsequently named Artificial Intelligence in 1955 [33].

2.1.1 Machine behaviour

Ancient automata — Clepsydra

In Ancient Greece engineering was sufficiently advanced to make artefacts, such as water clocks and automata, that would exhibit a single pre-determined behaviour. The water clock or clepsydra was originally a simple container with an aperture through which water flowed, emptying the container over a known time span. The Greek engineer Ctesibius is known for adding water wheels and floats to drive a further mechanism that would indicate the passage of time [28] (Figure 2.1). Though simple, this behaviour was of great practical use, enabling clepsydra to be used as alarm clocks or stop watches and was only made obsolete by the 17th Century invention of the pendulum clock.

Instilling artefacts with a certain specific behaviour was therefore within the reach of engineers of antiquity. A greater challenge was to instil a single artefact with a range of complex behaviours.

Hero's mechanical theatre

A more complex example, given by Hero of Alexandria in a work dated around the second half of the first century AD, posited a mechanism of an automatic theatre (Figure 2.2), itself first described by Philon of Byzantium in the late third century BC. The resulting display showed five scenes with animated characters, scrolling backdrops, timed sound effects and main theatre doors which opened and closed to hide the scene transitions [32, 3]. The energy for the actions was provided by a series of cords attached to a weight which sat in a cylinder of sand that was slowly emptying. Each cord was attached to a single mechanism with an appropriate length such that the action occurred at the intended time. The principle could be used to display any number of scenes of arbitrary complexity, as long as a cord-driven mechanism could be designed for each action.

Hero's theatre could be considered to be a single artefact that exhibited a rich range of behaviours but each individual action required its own separate mechanism. Even two instances of similar action, such as in first scene where two shipbuilders perform a hammering and sawing action, required two separate mechanisms. The range of actions of a single artefact was therefore tightly limited by considerations of space and complexity.

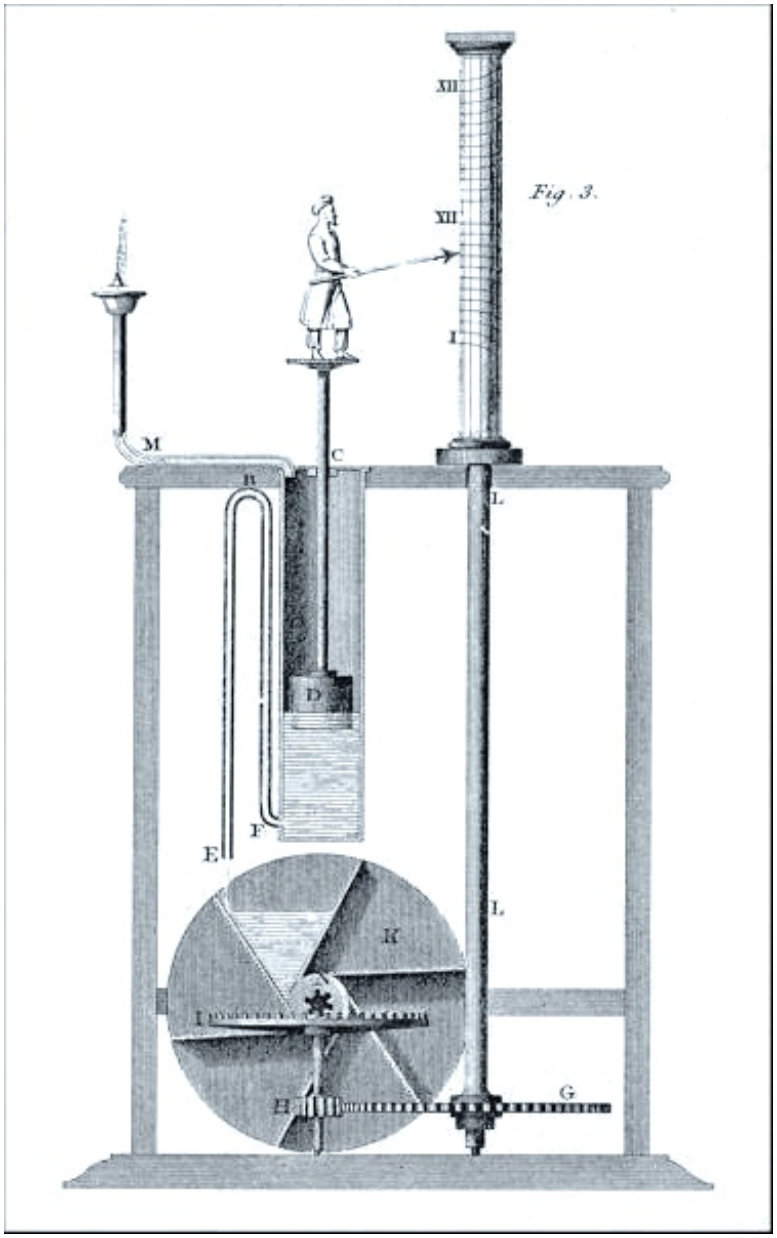

Figure 2.1: An early 19th-century illustration of Ctesibius' (285–222 BC) clepsydra from the 3rd century BCE. The hour indicator ascends as water flows in. Also, a series of gears rotate a cylinder to correspond to the temporal hours. Image in the public domain, via Wikimedia Commons `https://commons.wikimedia.org/wiki/File:Clepsydra-Diagram-Fancy.jpeg`

Figure 2.2: The fifth and final scene. The figure of Athena has been raised up from where it lay on the floor, to then move on an elliptical track around the stage. The backcloth shows Ajax in the sea. At the same moment when the flat panel, depicting the thunderbolt, falls from above, and disappears into a slot cut in the stage floor, a small second backcloth (painted so that it exactly matches a portion of the seascape) suddenly descends to cover the figure of Ajax, as if he has vanished into the sea. [3]

To stage a different production, the theatre would have to be significantly modified in a way that required the skill of the original builder, some mechanisms may be reusable but a change in behaviour would effectively necessitate the construction of a new theatre. The theatre therefore, while able to exhibit a range of actions, could only ever perform them in a single sequence.

L'Ecrivain

By the 18th century, clockwork engineering had advanced such that automata were built whose behaviour was configurable by a user. Pierre Jaquet-Droz, a Swiss watchmaker is known for a number of exquisite works, of which L'Ecrivain (The Writer) is a sophisticated example [56, p.23].

Inside the automaton is a disc containing forty notches, these notches can each contain one of a set of tabs. Each tab is engraved with a letter of the alphabet so a human operator can tell which letter of the alphabet this tab will induce the automaton to write. All tabs engraved with the same letter are of a specific height, such that when they are rotated to the top of the disc, they offset an arm in the mechanism by a specific amount. The offset of the

arm causes an internal cylinder to raise or lower such that the section encoding the chosen letter is aligned with the mechanism for driving the automaton's arm. The automaton's arm is driven by three internal levers, each one causing movement in one of the three spatial dimensions, these levers are bought into contact with the appropriate section of the internal cylinder, which is composed of a series of cams, such that each lever will be in contact with a cam which produces the appropriate movement as the cylinder rotates about its axis. As with Hero's theatre, the number of available actions was strictly limited, one letter could be written for each set of three cams in the cylinder, but in L'Ecrivain they could be arranged into as many unique sequences as there were unique combinations of tabs on the disc, without requiring any fundamental modification of the system.

The Jacquard Loom

A critical development in the determination of machine behaviour was first demonstrated in 1801. Designed by Joseph Marie Jacquard, the Jacquard loom responded to patterns of holes punched into cards that passed through a device which would convert the pattern into an appropriate internal configuration. In weaving, a cloth can be considered as composed by a series of parallel "warp" threads interlaced with a series of perpendicular "weft" threads. In plain cloth the weft lies on alternate sides of each warp thread; patterned cloth can be produced by modifying which sides of the warp threads each subsequent weft thread lies. In the Jacquard loom, each card had a space corresponding to each warp thread, the presence or absence of a hole corresponded to the position of a hook which each held a single weft thread. The position of the hook would determine which side of the warp thread the weft thread would be placed [45]. Each card therefore fully described a line in a woven pattern.

Comparison of mechanical automata

A tab in L'Ecrivain would correspond to one of the character patterns encoded on its central cylinder but a "punched card" in the Jacquard loom would itself define a single line of a woven pattern. The critical difference is that each line of the pattern was symbolically represented on the chain of cards and that the Jacquard loom was therefore capable of weaving as many patterns as there were unique representations on a series of cards of

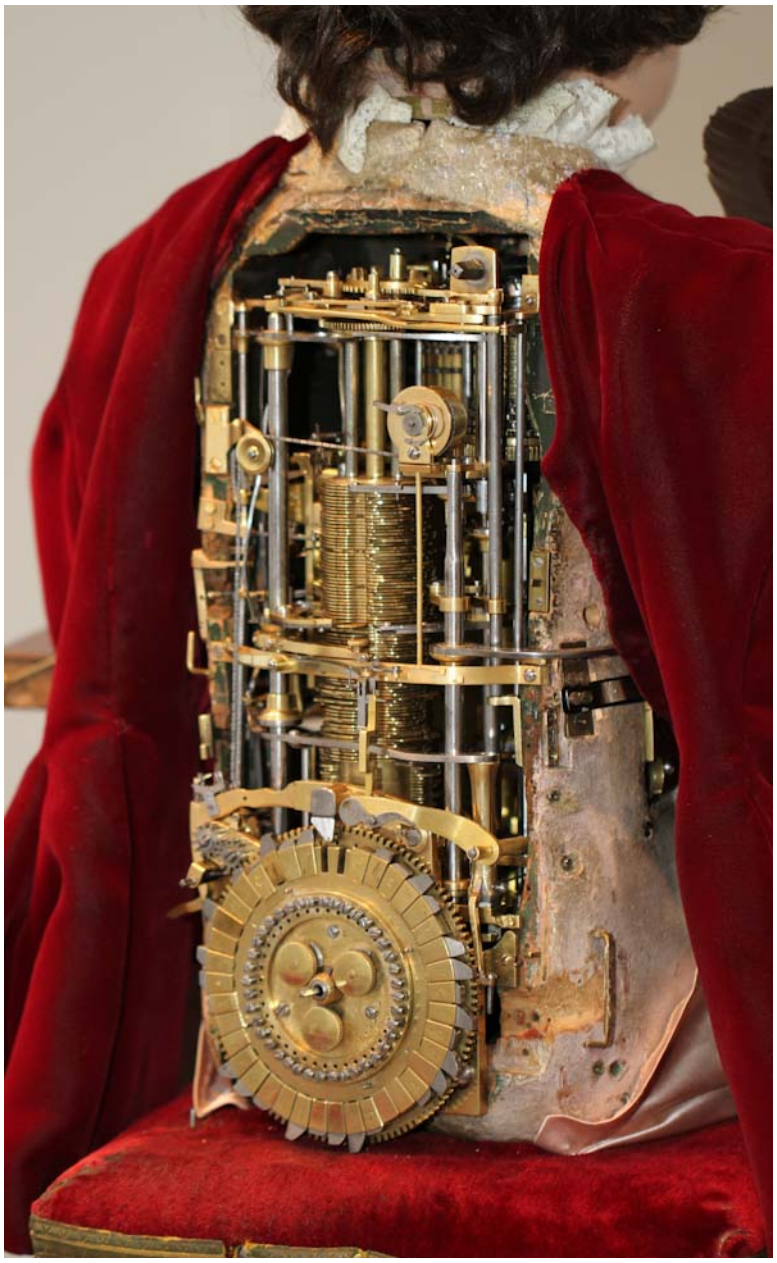

Figure 2.3: A view of the rear of L'Ecrivain, showing the mechanism and the configuration wheel holding 38 of 40 tabs. Used with permission of Ariel Adams, from http://www.ablogtowatch.com/jaquet-droz-the-writer-automata-awesome-antique-android/

Figure 2.4: This portrait of Jacquard was woven in silk on a Jacquard loom and required 24,000 punched cards to create (1839). Charles Babbage owned one of these portraits; it inspired him in using perforated cards in his analytical engine [1]. It is in the collection of the Science Museum in London, England. Image in the public domain, via Wikimedia Commons `https://commons.wikimedia.org/wiki/File:A_la_m%C3%A9moire_de_J.M._Jacquard.jpg`

arbitrary length.

The range and richness of the behaviours of artefacts clearly grew in superiority since the invention of Greek clepsydra but they were equivalent in their inability to react to events in the world. L'Ecrivain would continue to write its predetermined message just as Hero's Theatre's show must go on regardless of external influence, except in the case where the conditions for the pre-determined behaviour were fatally disrupted.

2.1.2 The question of intelligent behaviour selection

Investigations in to the determination of appropriate behaviours is also traceable as least as far into the past as Ancient Greece. Both Plato and Aristotle wrote on the subject.

Plato

Plato, in the Republic describes the soul as being comprised of three parts, the smallest and most critical part for a "just soul" was the thinking part — the *logisticon*. In the allegory of the cave, prisoners see shadows of objects, which are recognised by comparing their shape to ideal "forms" [30]. The description of a thinking part of the soul that makes just decisions guided by recollection of representations of worldly objects can be seen as an early description of intelligent action selection.

Aristotle

Aristotle, questioning "how the soul moves the body, and what is the origin of movement in a living creature" posited that the heart was the seat of the rational soul. At the heart, the sensations from the various sense organs arrived in a "central sensorium" allowing perception to occur. Aristotle posits a single location for collection of, and action upon, the senses arguing that the subject that perceives, thinks, and moves is one and the same.

Aquinas

Aquinas in translating and interpreting the work of Aristotle discussed intelligent behaviour as directed action defined and chosen by the actor. Physical stimulations from the

world are carried into the body as "phantasms" which are then acted upon as a result of comparison with generalisations of internal phantasms formed through passed experience. Aquinas used a metaphor of the intellect lighting the phantasms within the self; in contrast to Plato, the stimulations from the world were compared with a set of forms, internal, imperfect, and unique to the individual [20].

Descartes

Landmark work by René Descartes bought the question of intelligent behaviour into the Age of Reason, adopting a position of scepticism in order to avoid any unfounded assumptions [12]. The Cartesian Mind was in some sense a departure from the ideas of antiquity; where Plato distinguished between material "known to us through sensation" and forms, Descartes distinguished between "res extensa" that which is extended (in size and shape) in the world and "res cogitans" that which thinks. This position is known as Substance Dualism.

Aristotle required that there was a similar dichotomy, positing "prime matter" could not exist on its own, and must be "informed by a substantial form" which directs its characteristic behaviour. Therefore Aristotle placed the determination of behaviour of both humans and elements in their substance. Significantly, Descartes placed the behaviour of elements purely in their physical extension, describing it as "matter in motion" and the behaviour of humans was determined by the mind existing in non-corporeal "res extensa". Some aspects of antiquity remained in Cartesian thought, however, agreeing with the Platonic tradition in holding the intellect in higher standing than sensory knowledge [22].

As described by Wheeler [66], "[Cartesian] intelligent action the form of a series of sense-represent-plan-move cycles", in this framework nerves in the sense organs receive stimuli from the external world which sets in motion small fibres that connect the nerves with ventricles of the brain, in turn stimulating the flow of "animal spirits". The pineal gland, considered by Descartes to be the "seat of the soul", in interacting with the animal spirits was the conduit by which the mind was informed of the world by the body. On reception of the senses, the mind could represent the state of the world and reason upon it, finally commanding the body to move.

Wheeler also identifies that sensing and action were "theoretically disassociated" by being

separated by reason. As such the *inner state* was the determining concept for intelligent behaviour in the Cartesian Mind.

2.1.3 Artificial inner states

The number of actions definable in a Jacquard loom was vastly superior to that of any autonomous artefact that had been invented before, but the sequence of steps, though represented symbolically, still had to be represented explicitly. The requirement for one card per line of woven pattern meant that the loom was limited by the number of warp threads the loom could manipulate at once and the practical problem of managing card chains as patterns grew in length.

The Analytical Engine

The first machine which could surpass this limitation was the Analytical Engine [37] designed, though never built, by Charles Babbage a contemporary of Jacquard. Babbage's innovation was to have the machine capable of performing a number of operations and to have the performed operation determined by punched cards. Whereas a single pattern of holes will always result in a certain woven pattern in the Jacquard loom, a punched card may be used as an operator or operand for one of a number of operations in the Analytical Engine. The range of available operations included — but was not limited to — elementary arithmetic, conditionally feeding forward or backwards by a certain number of cards in the chain, and halting the machine.

Procedurally defined behaviour

It was the ability to determine the subsequent card to be actioned as a condition of the state of the machine which allowed procedures to be described implicitly, rather than explicitly. Consider the procedure for calculating the factorial of 6, explicitly it can be written as the results of $6 \times 5 \times 4 \times 3 \times 2 \times 1$ but as a procedure it could be written in a general form "accumulate the product of all the positive integers less than or equal to n" followed by an input "$n = 6$" which would apply to all positive integers. The Analytical Engine could therefore compose a large number of complex procedures from the small set of operations

which it was able to perform.

The defining feature of all the machines described so far is that their operation may continue entirely without human intervention once the action has started, and hence the state of a machine at each instant defines the state in the next instant. There is no capacity for intuition, rumination or inspiration in the tasks given. This type of behaviour is not exclusively limited to machines, there are certain similarities with slavery but the closest analogy amongst humans is with the human computer, who were required to process numbers in accordance with a given procedure that limited any requirement for insight. Alan Turing used the term "(mechanical) process" [65, p.262] to describe this kind of behaviour that could theoretically be carried out by a machine without human intervention.

Turing computability

Arguably the most important behaviour for which a corresponding mechanical procedure has been sought is that of determining whether "a given formula $\mathfrak{U}$ of the functional calculus K is provable", in a paper by Alan Turing in 1936 [65]. In investigating the possibility of such a mechanical procedure Turing invented a notation for defining procedures that could be simulated in a physical machine or by hand on paper. This was necessary as to give formal definition of a mechanical procedure, and so allow Turing to propose that an informal term as "a behaviour achievable purely through mechanical methods" may by replaced by the term "a behaviour which may be implemented on a Turing machine".

Related work was published by Alonzo Church, a contemporary of Turing, hence the proposal that all such mechanical computations can be carried out by a Turing machine became known as the Church-Turing thesis [9].

An important proof of Turing's was that certain numbers will not be representable in his notation, though large sets of numbers were [65]. The proof depended on the facts that there could not be infinitely many types of symbols printed on a tape as the difference between types was required to be greater than a lower bound, and the complexity of a symbol needed to be less than an upper bound before it could be identified at a glance. Turing gave the example of the decimal representations of the two numbers 999999999999999 and 9999999999999999 which have to be compared in a series of steps. Turing could identify

some large sets of numbers which could be represented in his notation and significantly showed how each Turing machine could correspond to exactly one computable number.

The significance of this observation was that a symbolic reference to a Turing machine could be made upon the tape of another Turing machine. Turing then showed that there existed a Turing machine which could encounter this reference and perform the action of the referred Turing machine. The importance of this discovery was that a single machine implemented thus, known as a Universal Turing Machine, could perform any definable mechanical procedure without any reconfiguration other than modifying symbols on the tape.

Babbage had designed an architecture whereby actions could be symbolically defined as procedures but once running the procedure would output to another medium, the Analytical Engine was equipped with a number of output methods including decimal dials, a bell and a curve drawer [37]. The Universal Turing Machine, in contrast, read input and wrote output to the same tape, therefore if the output was the number of a further Turing Machine and control moved to the location of that number the referenced Turing machine would be executed. In having no distinction between input, output, and instruction, the Universal Turing machine had access to the means of modifying its own behaviour.

Building upon the description of Turing Machines, computable numbers and the Universal Turing machine, Turing gave a further proof; there is no solution to the Entscheidungsproblem. Formally described thus, "there can be no machine which, supplied with any one $\mathfrak{U}$ of these formulae, will eventually say whether $\mathfrak{U}$ is provable". At once Turing's work both introduces a formalisation of computation, demonstrates a remarkable property, and describes an important limitation in its power.

Now machines could be defined in a way that the behaviour of one machine was multiply realisable on any implementation of a universal machine, as the behaviour was now inseparable from the data, what remained was to find the procedure by which a machine could write further procedures of the machine's own volition, the ability to *learn*.

Here we chart the development of a loosely related set of ideas from engineering, philosophy and mathematics into an explicit field of scientific investigation. As a subject, *Artificial Intelligence* more clearly describes the area of study, being largely defined by (i) the principles of intelligence, (ii) the design of artefacts to exhibit those principles, and (iii)

methods to indicate whether those artefacts are intelligent.

2.1.4 Turing's introduction of machine intelligence

1948 — Intelligent Machinery

Turing first treated the creation of a learning system as a serious scientific investigation in a report for the National Physics Laboratory in 1948 entitled *Intelligent Machinery* [64], however it was described as "not suitable for publication" although demonstrating "rather fundamental studies" [25, p.489]. In the report Turing describes two programmes towards the creation of an intelligent machine. The first approach was to "take a man as a whole and try to replace all the parts of him by machinery", while this was considered a "sure" method it was also admitted to be "too slow and impracticable". The more feasible approach was to "see what can be done ... more or less without a body", which included intellectual tasks such as game playing, language acquisition and mathematics.

This is an early example of a recurring dichotomy in the approach to building intelligent machines. Firstly, physically duplicating an intelligent being to an arbitrary degree was seen as a guaranteed method for success, but one that was clearly infeasible and threatened to grant no insight into intelligence itself. Secondly, investigating the abstract processes which characterise intelligence was immediately actionable, but required that intelligence was characterisable in such a way that the discovered processes were testable in an existent architecture even if it was, as in Turing's case, a paper machine implemented by hand.

In discussing intelligent behaviour in computable terms Turing must have hypothesised that an appropriate Turing machine existed, and hence could be discovered by a human searching for it intelligently. His emphasis in the 1948 report however was on the process by which a machine could reach the state of a universal machine implementing an intelligent machine with as little human interference as possible.

Turing broadly set out an architecture that could be "rewarded" or "punished" for its actions from outside the machine, causing its future behaviour to be respectively repeated or modified. The behaviour of rewarding and punishing was also within the grasp of another machine, extending Turing's investigation to include not only machines that could learn but also those that could *teach*.

Critical to the scientific field being conceptualised by Turing was the notion of determining whether or not any behaviour was intelligent. Even at this early stage Turing was sceptical about an objective measure, claiming that two people with a differing "state of mind and training" and knowledge of the system may commonly disagree on whether the expressed behaviour was intelligent. A scientific approach to determining intelligence could still be conceived of however, in the form of "a little experiment on these lines".

> It is not difficult to devise a paper machine which will play a not very bad game of chess. Now get three men as subjects for the experiment *A*,*B*,*C*. *A* and *C* are to be rather poor chess players, *B* is the operator who works the paper machine. (In order that he should be able to work it fairly fast it is advisable that he be both a mathematician and chess player). Two rooms are used with some arrangement for communicating moves and a game is played between *C* and either *A* or the paper machine. *C* may find it quite difficult to tell which he is playing.
>
> > (This is a rather idealised form of an experiment I have actually done).

One role of this experiment is to show that mechanical processes exist which produce behaviour that can be hard to distinguish from characteristically intelligent behaviour. The role of *B*, the intermediary, is to ensure that *C*, the player, only receives information about the behaviour of *A*, which may be a mechanical process. Turing does not specify exactly whether a mechanical process exists that will *always* be *indistinguishable* from intelligent behaviour, though his admission of having conducted a similar experiment is suggestive that the scenario described is not intended to exist solely as a thought experiment.

A number of arguments against the use of words such as "teach" and "learn" with regards to machine behaviour were anticipated and addressed in the same paper. Most pertinently, the criticism that any intelligent behaviour was merely an expression of the intelligence of the human programmer was described as being "rather similar to the view that the credit for the discoveries of a pupil should be given to his teacher".

1950 — Computing Machinery and Intelligence

Where *Intelligent Machinery* was seminal work, the philosophical investigation of the creation and recognition of intelligence in a machine was most fully developed, and with most impact, in *Computing Machinery and Intelligence*. Turing outright rejects the question "Can machines think?" as being formed of terms whose definition are too poorly defined. An alternative question is proposed, dependent on the results of a new experiment, apparently inspired by the previously described Chess experiment, called "the 'imitation game' ".

> It is played with three people, a man (A), a woman (B) and an interrogator($\tilde{C}$) who may be of either sex. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either 'X is A and Y is B' or 'X is B and Y is A.' The interrogator is allowed to put questions to A and B thus:
>
> > C: Will X please tell me the length of his or her hair?
>
> Now suppose X is actually A, then A must answer. It is A's object in the game to try to cause C to make the wrong identification. His answer might therefore be:
>
> > 'My hair is shingled, and the longest strands are about nine inches long.'
>
> In order that tones of voice may not help the interrogator the answers should be written, or better still, typewritten. The ideal arrangement is to have a teleprinter communicating between the two rooms. Alternatively the question and answers can be repeated by an intermediary. The object of the game for the third player (B) is to help the interrogator. The best strategy for her is probably to give truthful answers. She can add such things as 'I am a woman, don't listen to him!' to her answers, but it will avail nothing as the man can make similar remarks. We now ask the question, 'What will happen when a machine takes the part of A in this game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?'

This question, equivalently phrased as; "Are there imaginable digital computers which would do well in the imitation game?"; "Are there discrete-state machines which would do well?"; and "Is it true that by modifying [one particular digital computer *C*] this computer to have an adequate storage, suitably increasing its speed of action, and providing it with an appropriate programme, *C* can be made to play satisfactorily the part of *A* in the imitation game, the part of *B* being taken by a man?"; marks the creation of a process for indicating intelligence. It is not claimed to be a truth statement about the objective presence of intelligence, just a suitable working definition of a goal for the research field.

Turing goes on to say later that he firmly believes that it will be possible to build a machine "in about fifty years' time" that would "play the imitation game so well that the average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning". With this predicted progress in technology, and similar development in the "the use of words and general educated opinion", Turing also predicted that "one will be able to speak of machines thinking without expecting to be contradicted". Turing accepted that thinking as implemented on a digital computer may be "very different from what a man does" but suggested that whilst this objection is "very strong" it is one that "we need not be troubled by" should such a machine be constructed.

There were no explicit limits to the question of which principles founded intelligence, but only "digital computers" were permitted to "take part in our game". In allowing only digital machines to participate in the imitation game, explicitly to avoid any candidates being simply "men born in the usual manner" Turing also prohibited, implicitly, any method of intelligence that does not fall within Turing computability. Here again we see an example of the dichotomy between either building an intelligent being through duplication or simulating intelligent behaviour on a determined architecture.

2.1.5 Definition of the Artificial Intelligence project

Taken together, the delineation of the suitable approaches and goals as described by Turing formed the foundation upon which a full body of scientific study was to be built. The focus was on being able to describe mechanically the things intelligent humans do, the process was designing digital computers to implement these descriptions, and their validity was determined chiefly by evaluation of the resulting behaviour.

1956 — The Dartmouth conference

At a conference in held in Dartmouth; New Hampshire, specifically to formalise the field, and coin the name, John McCarthy set out a plan, thus:

> We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer. [33]

This proposal therefore was an explicit declaration of the fundamental assumption. Not only that every feature of intelligence can be simulated but also that a machine can and will be made to do so. At the conference itself some seminal work was presented, including Simon and Newell's Logic Theory Machine (LT).

Even in this proposal, the dichotomy of approaches was apparent, in discussing the requirement for some sort of originality, it was decided that something more sophisticated than simply adding randomness was required and that two procedures were available, "One of these is to find how the brain manages to do this sort of thing and copy it. The other is to take some class of real problems which require originality in their solution and attempt to find a way to write a program to solve them".

An echo of the Turing's imitation principle of intelligence is also present, as McCarthy also avoids defining a formal, objective test for intelligence, stating "For the present purpose the artificial intelligence problem is taken to be that of making a machine behave in ways that would be called intelligent if a human were so behaving." [33]

2.2 Making a Mind versus Modelling the Brain

One approach, fundamentally grounded in the mind, was searching for the mechanisms through which inner representations could be manipulated to result in representations of actions to be actuated in the body. It was mind prior, as it assumed a mind could be built, then a body could be bolted on like a marionette.

The other branch, most distinct by its foundation in the body, attempted to model the action of the brain. Hypothesising the phenomena of the mind would emerge, or arise from the replication of the physical systems. It was brain prior as it assumed the structures of the brain, entirely defined and properly constructed was a sufficient condition for the emergence of a mind.

Notably, at this stage, the approach of duplicating a physically intelligent system was conceptualised in a way that was also Turing computable. The intelligent being was largely equated with the activity of the brain, the brain was largely equated with actions of neurons, and neurons in turn were being equated with the action of the abstract, computable, model developed by Pitts and McCulloch. Either approach, therefore, though distinct in their origin could be carried out through design, implementation and observing the behaviour of programs.

The dichotomy in these approaches was described by Dreyfus as Making a Mind vs. Modelling the Brain [16].

2.2.1 Making a Mind

The founding idea behind Making a Mind was that the mind was an emergent phenomenon that resulted from the formal manipulation of symbols, patterns of which were representative of features of the outside world. There was no upper limit to what could be represented, complex or subtle, object or concept, but there was assumed to be a lower limit, which Dreyfus likened to the ultimate 'simples' described by Leibniz, in terms of which all complex concepts can be understood. A large aspect of the approach was the identification of an appropriate set of such simple terms with which all aspects of human intelligent behaviour could be understood.

Given a fully defined context such as in the game of chess, the simple terms can be clearly seen. All that would be needed to fully describe all aspects of a game of chess can be captured in terms of spaces, pieces, moves and so on. Further problems such as the clothing of one of the players, or the minuscule positioning of a piece within its square e.g. whether it exactly centred, or slightly overlapping an edge, can be ignored.

The solubility of identifying the simple terms of a well defined context allowed the projects founded on the representation of the situation and manipulations thereof to have some early success that was, in appearance, startling and impressive.

1956 — The logic theory machine

A project at the RAND Corporation produced the Logic Theory Machine (LT) [44] which would attempt to prove that a given theorem could be proved from a given set of axioms. The significance of LT was that the procedure used was not simply an exhaustive search of the possible solution space, but extra techniques were implemented to apparently guide the search. These techniques, named heuristics, were intended to demonstrate that a machine could reach a solution using methods similar to those used by an insightful expert, as opposed to one merely following an effective method.

In the case of an expert proving a logical theorem, one process by which investigation was deemed to be guided was the recognition when theorems were similar and hence LT was designed with a method of evaluating the similarity of two theorems. Another heuristic employed by LT was to store any proven theorems and use them alongside the initially given axioms to guide further investigation, described by Simon and Newell as "learning".

This approach achieved some success, with LT being able to prove a number of elementary theorems, with some elegance. A novel proof produced by LT of Russell and Whitehead's theorem 2.01 from *Principia Mathematica* was submitted to the *Journal of Symbolic Logic*. Publication was refused on the grounds of the proof being co-authored by a program [17, p. 184].

1958 — The Physical Symbol System Hypothesis

The initial success led to two significant beliefs, firstly that progress would continue largely unabated, and secondly that the success of the approach was grounded in its capturing of the nature of intelligence. The first belief is aptly demonstrated in a quote from Newell and Simon relating to the predicted progress in heuristic-based systems "[I]n a visible future ... the range of problems [computers] can handle will be coextensive with the range to which the human mind has been applied." [62]. The second belief is more formally defined in The Physical-Symbol System Hypothesis [43].

> A physical-symbol system has the necessary and sufficient means for general intelligent action.
>
> By "necessary" we mean that any system that exhibits general intelligence will prove upon analysis to be a physical-symbol system. By "sufficient" we mean that any physical-symbol system of sufficient size can be organized further to exhibit general intelligence.

The significance of the thesis is hard to overstate, the claim that a physical-symbol system may exhibit general intelligence (not just intelligent behaviour) and that any system exhibiting general intelligence must be a physical-symbol system implies that a formalism for the intelligence exists, and any other conception of intelligence will be reducible to an equivalent formalism. This is described as an "empirical hypothesis" which therefore largely described how research may progress to work towards finally discovering and implementing a generally intelligent system.

Thus the research community was ready to start embarking on the projects that would exhibit general intelligence.

1959 — The General Problem Solver (GPS)

Buoyed by the success of LT, Simon and Newell set out to design a similar system that would be able to solve any well defined problem presented to it [42], the project was something of a success, being practically implemented and leading to a successor, the Soar architecture. Unfortunately the GPS could only solve problems that defined a small context,

as more complex problems quickly ran into a combinatorial explosion of intermediate states.

1964 — Dendral-64

Considered the first *expert system* as it attempted to reproduce the behaviour of experts in the same field, Dendral-64 would receive spectrometry data as input and respond with a number of candidates for the chemical that produced the data [29]. Notably, Dendral-64 used heuristic processing in which rules-of-thumb used by human experts were encoded into the behaviour of the machine, rather than simple, comprehensive searches of the entire search space. This approach traded a guarantee that an answer would be found for processing times that were not so long as to be impractical. This meant the output was likely to be fallible, so Dendral-64 was designed to return a number of likely candidates that a relatively junior expert would be able to review to determine the most likely candidate. Even with the greatly reduced processing times and internal data store requirements of a heuristic process Dendral-64 required to be only applied to a very tightly defined context, such as spectroscopy, and would not ever branch out to other problems.

1972 — SHRDLU

Winograd's SHRDLU, could react to and respond with a subset of English which allowed surprisingly coherent conversations and complex behaviour [67]. Its behaviour was determined by the translation of sentences of plain text into a formal declaration of action, or propositions about the environment. This could then be compared against SHRDLU's representation of the world and either the truth of the proposition could be determined logically, or the difference between the required state and the current state could be determined and hence an effective sequence of actions could be determined to perform the requested task.

2.2.2 Critiques of Making a Mind

1973 — The Lighthill Report

By the mid-seventies, requests for funding in Artificial Intelligence projects had reached a level of frequency that it came to the attention of the government which had no official stance on the value of related research. A report was commissioned to evaluate the current and potential progress. Professor Sir James Lighthill FRS was chosen as someone who was not actively involved in the field but with the required experience in mathematics, biology and engineering. The report drew a distinction analogous to the dichotomy previously described, splitting work into the categories of; "Advanced Automation" (category A), for any work whose goal was to produce useful behaviour which would be considered intelligent if seen performed by a human; "Computer-based [Central Nervous System] research" (category C), for any work that wanted to further the understanding of the natural central nervous system through study upon models of artificial neurons. A third category, "Building Robots" (category B) was placed theoretically between the other two categories. Category B contained all work which aimed to make progress towards the goal of the creation of an artefact that exhibited behaviour from category A through methods developed by work in category C, it was therefore also referred to as the "Bridge category" [31].

The tone of the Lighthill report was more cautious than much related literature, often referring directly to admissions of naïvety and subsequent disappointment from distinguished researchers. Categories A and C were therefore hailed as promising endeavours within the contexts that individual projects in category A remained within a tightly defined context, and work in category C remained tightly aligned to merely computationally supporting work that would have otherwise been performed in psychology and neurobiology. The tone was largely critical, though not unfair, of category B which was unable to unite categories A and C into a single research field by citing successful projects that drew heavily on both categories.

The central critique of category B was that no projects existed that were both (i) sufficiently related to the other two categories and (ii) successful. This critique can be interpreted as an example of the "No True Scotsman" informal fallacy in that any successful project suggested as a counterexample could be immediately claimed as being not sufficiently in either

category A or C, and any project that drew significantly from both categories A and C when suggested as a counterexample could be claimed to be insufficiently successful. This can be seen in the report as any projects that are inarguably successful, are inevitably described as using "general computational theory" which, given that the roots of computational theory lay in Turing's work in decidability and intelligence is not obviously distinct from category A research. Similarly, Winograd's SHRDLU project which is admitted to draw deeply on work from both other categories was claimed to be essentially not successful enough as "suggestions for possible developments in other areas that can properly be inferred from the studies are rather discouraging". Lighthill also noted that "one swallow does not make a summer", though this was "banal" by his own admission.

The Lighthill report had a significantly negative effect on AI research funding, though it was not entirely unfair. It was broadly compatible with the view of Dreyfus and potentially saved the reputation of AI from a more disastrous period which may have occurred in the future had overconfidence and overpromising been allowed to continue unchecked.

1980 — The Chinese Room Argument

Two stances on the ultimate potential of the Artificial Intelligence project were identified by John Searle [61]. "Strong" AI held that "the appropriately programmed computer really is a mind" by virtue of instantiating its program alone. "Weak" or "cautious" AI held that artificial intelligence was a useful tool with certain practical applications and academic value, but allowed that there may not exist any program which would instantiate a mind.

Searle described a thought experiment, to argue in favour of weak AI. In its original form the target of the argument was a claim that Schank and Ableson's *Script Applier Mechanism* (SAM) could understand the stories that were presented as input, it should be noted that the claim was not attributed to Schank or Ableson. Searle is clear to state that the argument applies to any formal system, such as any Turing computable process.

Described in work published in 1975 [59], SAM could receive input in the form of a script, a formal syntax for describing a "story" (Figure 2.5) and subsequently respond to further input in the form of "questions" in natural English with "answers to the questions" in natural English (Figure 2.6).

```
script: restaurant
roles:  customer, waitress, chef, cashier
reason: to get food so as to go up in pleasure
           and down in hunger

scene 1: entering
         PTRANS self into restaurant
         ATTEND eyes to where empty tables are
         MBUILD where to sit
         PTRANS self to table
         MOVE sit down

scene 2: ordering
         ATRANS receive menu
         MTRANS read menu
         MBUILD decide what self wants
         MTRANS order to waitress

scene 3: eating
         ATRANS receive food
         INGEST food

scene 4: exiting
         MTRANS ask for check
         ATRANS receive check
         ATRANS tip to waitress
         PTRANS self to cashier
         ATRANS money to cashier
         PTRANS self out of restaurant
```

Figure 2.5: "A sketch of a script for a restaurant from the point of view of the customer." Actions are specified in terms such as MTRANS for mental information transfer and PTRANS to move a physical object. [59]

```
Input:
     John went to a restaurant. The hostess seat-
     ed John.  The hostess gave John a menu. The
     waiter came to the table.  John ordered lob-
     ster.  John was served quickly.  John left a
     large tip.  John left the restaurant.

Questions and Answers:

Q: What did John eat?
A: LOBSTER.
Q: Who gave John the menu?
A: THE HOSTESS.
Q: Who gave John the lobster?
A: PROBABLY THE WAITER.
Q: Who paid the check?
A: PROBABLY JOHN.
Q: What happened when John went to the table?
A: THE HOSTESS GAVE HIM A MENU AND JOHN SAT DOWN.
Q: Why did John get a menu?
A: SO HE COULD ORDER.
Q: Why did John give the waiter a large tip?
A: BECAUSE HE WAS SERVED QUICKLY.
```

Figure 2.6: Example session with the Script Applier Mechanism [59]

The experiment asks one to imagine whether manually processing the input that would be received by SAM and hence producing the same output will give the human processor an understanding of the story described in the script. To avoid the objection that the human processor would be able to read the script as they would a normal text, Searle defines that all text is written in a language he cannot read, Chinese in this case, so all symbols are identified naïvely.

> Suppose that I'm locked in a room and given a large batch of Chinese writing. Suppose furthermore (as is indeed the case) that I know no Chinese, either written or spoken, and that I'm not even confident that I could recognize Chinese writing as Chinese writing distinct from, say, Japanese writing or meaningless squiggles. To me, Chinese writing is just so many meaningless squiggles.
>
> Now suppose further that after this first batch of Chinese writing I am given a second batch of Chinese script together with a set of rules for correlating the second batch with the first batch. The rules are in English, and I understand

> these rules as well as any other native speaker of English. They enable me to correlate one set of formal symbols with another set of formal symbols, and all that 'formal' means here is that I can identify the symbols entirely by their shapes. Now suppose also that I am given a third batch of Chinese symbols together with some instructions, again in English, that enable me to correlate elements of this third batch with the first two batches, and these rules instruct me how to give back certain Chinese symbols with certain sorts of shapes in response to certain sorts of shapes given me in the third batch. Unknown to me, the people who are giving me all of these symbols call the first batch "a script", they call the second batch a "story", and they call the third batch "questions". Furthermore, they call the symbols I give them back in response to the third batch "answers to the questions", and the set of rules in English that they gave me, they call "the program". [61]

The argument demonstrates that an instantiation of SAM is not sufficient for understanding as Searle could *be the instantiation* himself, and would not understand the story. This is taken by adherents to the Chinese Room Argument to show formal symbol manipulation alone does not give rise to understanding, in other words, syntax is not semantics.

In discussing what may give rise to understanding, if not symbol manipulation, Searle distinguishes between computers and machines, stating that the brain is a type of machine, but that understanding must arise from its causal properties, rather than the instantiation of a formal syntax.

Domain specificity

Two problems were encountered relating to domain specificity of an AI system. The first was that there was no general procedure for taking two symbolic solutions, such as SHRDLU and GPS and combining them to form a system that could behave intelligently in the union of the two contexts. The second was that as a situation became gradually more complex, the set of combinations and required manipulations became exponentially larger, eventually becoming unmanageable. These problems could be managed initially by manually combining existing systems on a case by case basis, and by increasing the complexity of a system to handle a more detailed domain. For example, SHRDLU's blocks

world relied on the combination of "the vision project of David Huffman (1971), the vision and constraint-propagation work of David Waltzi (1975), the learning theory of Patrick Winston (1970), the natural language understanding program of Terry Winograd (1972), and the planner of Scott Fahlman˜(1974)" [58] and could act in a non-trivially complex context of a physical simulation of blocks and a subset of natural English. The process of integrating further systems into SHRDLU or expanding them to handle a more detailed physical simulation would have eventually become so complex as to be infeasible to implement, either by limiting the processing power of the utilised hardware, or by pushing the human designers past their capabilities.

The Common Sense Problem

A single game of chess can be broadly captured by recording which space each remaining piece was occupying after every move. To capture a more complex, or less well defined context, such as the entirety of a chess championship, it is not always immediately clear which facts not immediately pertinent to a game may become features that are critically important to explaining further occurrences, such as a judgement about an illegal move. Imagine, for example, that a player's dangling sleeve knocked a piece from one square to another.

An approach that was a potential solution to the domain specificity problems of system integration and combinatorial explosion was to design systems that had a common sense, and therefore had access to all knowledge that may become important. Marvin Minsky has been associated with a claim that "representing a few million facts about objects including their functions" would solve this problem [15], but this simply exacerbated the problem of determining which fact was relevant at any one time. Consider again the example of the chess championship: a system may have sufficient data about invalid moves in chess to know to take account of a players loose clothing, but never acts appropriately as it is taking too long in considering whether to act on a vast array of other facts such as the impossibility of whistling whilst chewing gum, and common misspellings of the word scissors.

The first step fallacy

Artificial Intelligence projects that attempted to produce intelligent behaviour through formal manipulation of symbols quickly demonstrated encouraging results, notable examples are Winograd's SHRDLU which could be interacted with through a subset of the English language and appeared to exhibit comprehensive knowledge of its environment "blocks world". There were also impressive early results from projects based around language acquisition, and theorem solving.

Many projects proudly publicised their achievements, and this would often be accompanied by predictions of steady progress until a system had a capability at the level of an expert human. However, the progress inevitably slowed. This became known as a "first step" fallacy, where impressive early results were interpreted as a promise of similarly impressive progress when trying to expand a system into broader or more advanced functionality. The cause of the fallacy was identified by Dreyfus [14] as the combinatorial explosion of complexity encountered as a system was expanded into more varied domains. The problem being that a formal system has to objectively process all symbols and input patterns, where as a human experiences an environment in a way better described as having significant things present themselves to you.

2.2.3 Modelling The Brain

As the brain was taken to be the seat of intelligence, an approach to creating intelligent behaviour was to simply reproduce, at a given level of abstraction, the action of the brain, under the assumption that the same manipulations of stimuli seen in the brain would produce the same responses in artificial systems.

Associationism

Philosophically, the character of Connectionism can be traced back at least as far as work by John Locke, named "associationism" and founded on the idea of co-occurrences and correlations of physical events and mental states. Commonly summarised by the epithet *items that "go together" in experience will subsequently "go together" in thought*, it was a typically empiricist approach, explaining where possible learning and behaviour as a result of past

experience [21].

Ideas are associated in the mind through experience and ideas are derived from sensations. The sensations themselves are caused by something outside the head, so there is still a concept of "inner" as distinct from the outside world.

This allowed the study of intelligent behaviour to be grounded in visible processes in the world. Experiences will effect someone in such a way that they will be more likely to act in the same way again. The advantage was that behaviour selection could have a causal explanation. This had implications for the notion of free will in intelligent behaviour, as identified by Nietzsche in 1887, "seeking the truly effective and directing agent ... where the intellectual pride of man would least desire to find it (in the vis inertiae of habit ... or in a blind and chance mechanistic hooking together of ideas ... passive, automatic, reflexive, molecular and thoroughly stupid)" [46].

1943 — McCulloch and Pitts model of the neuron

Connectionist AI is a fundamentally distinct approach to producing intelligent behaviour. In a seminal paper by W. S. McCulloch and W. H. Pitts [36] and drawing strongly on an "all-or-none" interpretation of the activity of a neuron, a formal model of nervous activity was defined. Various features of physical neurons were declared irrelevant, such as the speed of propagation of signals. The brain was therefore defined by the activity of a network of interacting formal neurons. All logical calculations were theorised to have at least one net with equivalent behaviour and all nets were describable by a formal statement. The advantage of encoding a behaviour in a net, rather than a series of statements was that the process of discovering the appropriate configuration could be mechanised, in a way analogous to Turing's suggestion in *Intelligent Machinery*.

1949 — The Organization of Behaviour

The process for discovering the neural network of an appropriate configuration was grounded in the work of Hebb who postulated that repeated co-occurrences of activity of two cells, most significantly in the case of a neuron whose firing is determined by the presence of a certain stimulus, firing at the same time as a neuron which is instrumental in

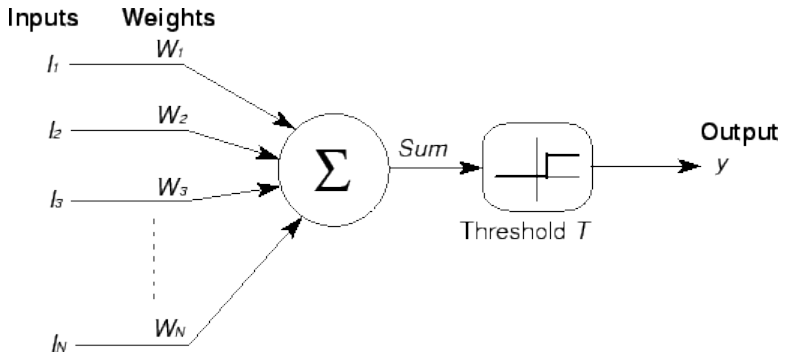

Figure 2.7: The McCulloch and Pitts model of the neuron, showing the weighted sum of inputs 1 to N being passed through a threshold function.

---

the production of a suitable behaviour in response to the stimulus, increases the probability that activation of the cells will co-occur in the future [23]. Quoting Hebb "the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability". A colloquial description of the same phenomenon is to say "neurons that fire together, wire together", capturing the spirit of Hebb's observation, if not the precise letter.

1958 — Mechanisation of thought and the Perceptron

In 1958 many of the delegates present at the Dartmouth conference met again in Teddington, London at an annual symposium hosted by the National Physical Laboratory; the theme for that year was "The Mechanisation of Thought Processes" [35]. Many of the subjects that would come to characterise much of AI were present such as medical expert systems, image recognition, and neural nets for pattern discrimination.

At this event [34] Rosenblatt made a statement that although the brain makes decisions and hence controls the body based on logic rules, it also interprets the external environment. In this recognition of the behaviour of the brain being coupled with the environment Rosenblatt made clear his concern not only with what the brain did, but how it did it. Developing an internal model of logical laws from experience was a distinct approach from the symbolic approach favoured by Minsky and McCarthy as they were looking for a set of logical functions that, statically implemented, would be able to produce the full range of required behaviour. Rosenblatt was clearly aware that an artificial neural network could be simulated on a Universal Machine, and indeed Rosenblatt's belief was that a computer

used any other way would be inadequate to express intelligence.

Rosenblatt united the theoretical model of the with biological observations in a biologically plausible model of computation that could be used for both prediction of neuronal activity as well as synthesis of behaving systems in a model he named the Perceptron [55]. Various training methods were applicable to the network which would be initialised with random connections, of random strength, between the neurons. Training would take the form of providing stimuli with known correct responses and in the event of an incorrect output, inhibiting the weights of any units that fired incorrectly, and enhancing the weights of any units that incorrectly failed to fire. If the problem was linearly separable then the network was guaranteed to converge on the set of weights that would perform the specified mapping, should it exist [47].

1969 — Perceptrons

A critical event in the Connectionist approach to Artificial Intelligence was the publication in 1969 of *Perceptrons* by Minsky and Papert [39]. The aim of the book was to discover the capabilities of the type of parallel computing seen in artificial neural networks, by formulating various mathematical theorems on the capabilities of the Perceptron. The Perceptron was considered a suitable candidate to represent the entirety of the approach as its characteristic of "adding up evidence obtained from many small experiments" was shared with most, and perhaps all, Artificial Neural Networks, and in recognition of the pioneering work of Frank Rosenblatt. The analysis was not intended to be critical of the Perceptron, merely mathematically objective, reiterating the claim in the 1988 reprint that "romantic claims [of Perceptrons' capabilities] have been less wrong than the pompous criticisms", though important limitations were described [38]. It was shown that some problems were unsolvable within the given architecture, whereas other problems were "surprisingly tractable". Minsky and Papert make clear that their intuition was that research into multi-layer Perceptrons would be 'sterile', but they encouraged research into elucidating, or rejecting, their judgement. Nevertheless, *Perceptrons* has earned the reputation of dealing a near-fatal blow to the attitude of academics and funding bodies towards Perceptrons, and Connectionism in general.

The canonical example of a non-linearly separable problem, hence one unsolvable by the

In_1	In_2	Out
0	0	0
0	1	1
1	0	1
1	1	0

Table 2.1: Minimal training set for defining the "exclusive or" function. A single layer perceptron *will not* converge to a solution when trained against this set.

In_1	In_2	In_3	Out
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 2.2: Training set for defining the "exclusive or" function, augmented with a third input. A single layer Perceptron *will* converge to a solution when trained against this set.

Perceptron, is the XOR function. The mapping from two inputs to one output which defines the XOR function are shown in Table 2.1. Minsky and Papert observe that a Perceptron could be trained reproduce the XOR function by augmenting the training set with a third input column which, so while training a Perceptron on the input-output pattern mappings in Table 2.1 would fail, training against the set in Table 2.2 would succeed. In this case the value of the unit in column In_3 represents whether the value of both In_1 and In_2 is 1. The same effect can be achieved by training a Perceptron against the set in Table 2.1 if it has a hidden unit that accepted input from In_1 and In_2, though a procedure for finding the requisite connection weights, of similar power and utility to the Perceptron convergence procedure was not widely known until the publication of *Parallel Distributed Processing*.

1981 — Hinton mapping

In 1981 [24] a technique was introduced to address the problem of stimulus equivalence which by 1992 had been named Hinton mapping [6]. This problem, which is the invariance of symbolic information independent of transformation within a search space, is a problem

present in neural networks when applied to visual identification tasks. The occurs when many different patterns of stimulation can identify the same object. For example, a printed letter 'A' should be identified as such regardless of its orientation, size or position but would present a different stimulus to a neural network under such transformations.

Hinton mapping is based in the theoretical assumption that visual processing depends on a representation of a scene in terms of three-dimensional features relative to a point of view. The representation of these features is enabled be groups of retina-based units, object-based units and mapping cells. Each retina-based unit represents the presence of a feature, which in the case of detecting possibly rotated and resized letters in a 2D plane could be a horizontal line of a certain length, or a slightly angled line of another length. Each object-based unit represents a canonical instance of an object to be detected. Mapping cells read from collections of these retina-based units, and hence represent the detection of the features that define a certain presence, for example, the presence of a slightly rotated capital A may be represented for a mapping cell by reading from the retina-based units which are active in the presence of a two-tilted lines and one shorter tilted line. This process requires one mapping cell per valid hypothesis. The mapping cell which represents the best mapping of an object-based unit to multiple retina-based units will become the most active [24]. The detected object is therefore the object cell associated with the most active mapping cell. Hinton mapping can be computationally expensive, as the number of mappings increases combinatorially when each feature is presented in all possible combinations of rotation, position, size, and any other transformations that may be applicable [4].

1986 — Parallel Distributed Processing

Parallel Distributed Processing [57] noted that *Perceptrons* justified a critique that was limited to single layer Perceptrons by explaining that the set of multi-layer Perceptrons was so large as to be vacuous; one could implement a Universal Turing Machine with sufficient linear threshold units; and that there was no strong training procedure for multi-layer Perceptrons analogous to the *Perceptron convergence procedure* for single-layer Perceptrons. These arguments were countered by firstly explaining that the focus was on the investigation of multi-layer Perceptrons that were "as parallel as possible" and hence far from implementing a Universal Turing Machine, and that a number of strong training proce-

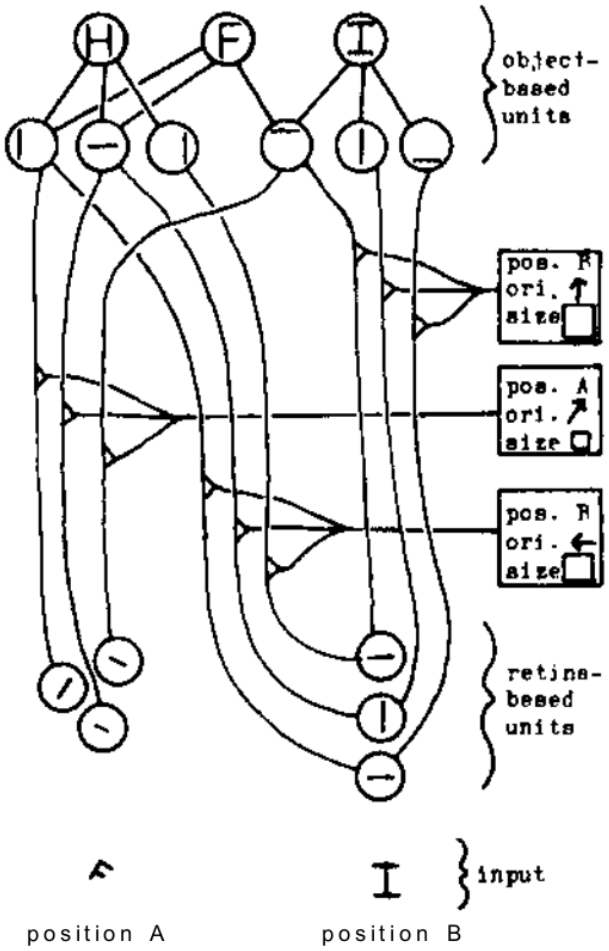

Figure 2.8: A diagram of Hinton Mapping from [24]. The retina based units are sensitive to basic features in a visual scene. Each retina-based unit is joined to an object-based unit by a mapping unit. The activity of an object-based unit is the multiple of the activity of a retina-based unit and the mapping unit which joins them. The activity of a mapping unit is the multiple of the activity of an object-based unit and the mapping unit which joins them. Many more retina, object, and mapping units are required than are depicted.

dures had since been discovered for systems with hidden layers, going so far as to describe a generalised procedure that could be applied to "arbitrary networks with multiple layers and feedback among layers".

Comparison of symbolic AI and neural networks

Fundamentally distinct from Symbolic AI, an approach emerged where the process by which intelligent behaviour could be investigated was grounded in the study of mechanisms and biological structures that may may give rise to learning and perception in the human, most importantly, in the human brain. As models of the brain conventionally took the individual neuron to be its simplest constituent and hence the defining feature of any specific model being the characteristic connections between the neurons, the field became known as Connectionist AI.

Neural Networks were in some sense a success, exhibiting traits that bear the mark of the intelligent: they are often described as learning: in degrading gracefully; in being biologically inspired; and able to generalise. However, algorithmic artificial neural networks are still fundamentally computational, existing outside of time, being Turing equivalent. Hence while avoiding some of the criticisms and pitfalls of symbolic formal systems they share some of the problems, not to mention introducing some problems of their own.

2.2.4 Critiques of Modelling the Brain

There are weaknesses in the approach of Modelling the Brain, although the focus is on the structure of the system which performs the learning. Across all neural architectures the quality of the results is dependent on the composition of the chosen training data. This leads to a large amount of time spent collecting and pre-processing a data set which can be seen as 'sneaking intelligence in through the back door'.

The requirements for the size or composition of the training data may become prohibitive. A study of the comparative performance of speech recognition systems trained on varying amounts of data found that performance improved linearly, in terms of percentage word identification error, as the amount of training data increased exponentially, in terms of hours of natural speech [40]. Significantly, a successfully trained network may not offer

any insight to the user about how the network has solved the problem, and hence offer no insight into how the human brain solves analogous problems.

Furthermore, the neural architectures which are defining the field are arguably a poor representation of the structure and activity of the mind. Neurons as modelled in neural networks are often modelled as static integrators of input. Even in recurrent neural networks, where the structure of the network may include loops, the full real-time dynamics of neurons are rarely modelled. Neuroanatomy work from Freeman [20] described neurons as releasing their neurotransmitters as an eletric potential over time. Not only is this simple behaviour often ignored in artificial neural networks, it is a process which also induces a negative current in the volume surrounding the neuron, not just at the axon, affecting much more than just the 'connected' neuron. Freeman also shows how collections of neurons interact in such as way as to be able to maintain a number of different steady states of firing patterns each of which effectively implement context-specific behavioural units, a major purpose of which is to modify the steady state of other collections of neurons. This real-time stateful complexity is not modelled in even the most highly publicised neuromorphic systems such as FACETS and SpiNNaker systems of the Human Brain Project.

2.2.5 The state of the art

Despite arguments about the ultimate limits of AI projects, a number of successful projects have been released to public use, using novel techniques and powerful hardware to reduce, if not remove, limitations.

Machine Translation, where text in a source language is automatically translated into text in a target language is a classic application of AI. A simple machine translation can be achieved by mapping each word in the source language to word in the target language. The action of such a simple system will be analogous to using a bilingual dictionary, looking up every word in the source text and replacing it with the associated word in the dictionary. Special behaviour would have to be implemented for each case where the word in the source language did not exist in the mapping, or when one word in the source language mapped to more than one word in the target language. These systems are simple to design and implement and will perform adequately with certain combinations of source text, target text, language pair and dictionary. The meaning of individual words, however, is

affected by the context in which they are used. Many examples exist of this contextual dependence, of which the phrases "Time flies like and arrow, fruit flies like a banana" [10] and "Police help dog bite victim" are memorable examples.

Statistical Machine Translation (SMT) is an approach that attempts to address this problem, with considerable success. Firstly, rather than referring to a formal mapping between words, SMT depends on a collection of documents translated by expert humans as training data. Secondly, rather than attempting to create a mapping between two languages that will work in certain contexts, SMT calculates a probability distribution representing the probability of a phrase in the target language being the correct translation of a given source phrase. The calculation is broadly informed by looking up the phrase in the training corpus and seeing how it has been translated previously, taking into account the similarity of the words surrounding the source phrase and similar phrases in the training data.

Google Translate is a product using SMT and a number of publications by Google show various SMT techniques outperforming competing techniques. Google reportedly use United Nations and European Union documents due to the fact that they are routinely translated into six languages, resulting in a huge corpus [63].

A collection of the best techniques, and raw computing power led to the figurehead of modern artificial intelligence called Watson [19]. Watson can do many things often thought beyond the capabilities of AI by its critics, natural language processing as well as the capability to handle ambiguity in its input. Watson competed in and won the American Quiz Show *Jeopardy!*, but made one telling error, answering "Toronto" to a question in the category of "US Cities". The notable error in this response, that Toronto is not located in the US, shows that Watson can be surprisingly incorrect, as well as surprisingly accurate.

Once artificial neural networks had shown themselves to be not only possibly applicable to a large set of problems, but also practically applicable to existent problems, and a training procedure had been developed, research continued and progressed until a number of powerful techniques had been developed. The current state of the art is represented in the techniques: Reservoir Computing [60], and Deep Learning [27].

Reservoir Computing is a technique whereby a network of artificial neurons is used in a randomly initiated and fixed state. In contrast to the Perceptron, where a network is modified until the intended mappings of input patterns to output patterns is achieved, a

reservoir will approximate the intended non-linear function to a varying degree depending on which inputs and outputs are considered to be the ones corresponding to a given problem, the task of training is therefore to find which set of input and output nodes appear to map input patterns to the intended output pattern.

In this way, a single reservoir can be used to solve many problems. Each problem would be solved by reading from different sets of nodes from the same network. Training on the weights of the output layer only is equivalent to simply listening to whichever set of output neurons appears to encode the correct behaviour. This is a much simpler task than modifying an entire, often fully connected and many layered, network. The interest of this approach is that a randomly initiated network, with appropriate dynamics implemented almost by chance, and hugely reusable, is broadly biologically plausible.

Deep Learning uses a sequence of non-linear transformations broadly characterised as expecting each layer to output some higher-level features extracted from the lower-level features output by the previous layer. Individual layers are commonly stand-alone neural networks resembling Perceptrons, but may include any architecture that will map input patterns of activation to output patterns of activation. Deep Learning is often used with a minimum of task specific customisation of the architecture, which suggests a general approach which is not restricted to a single domain, as were Expert Systems. This is only achieved, however, by supplying the network with a set of training data which represents the raw data of the problem to be solved. For a Deep Learning system to be applicable to more than one problem it would have to receive training data containing both types of problems, each of which would greatly increase the effective noise in the other.

2.3 Swarm Intelligence — Mind is social

An investigation separate from the search for intelligent machinery was the endeavour to understand the intelligent behaviour of natural systems such as ant colonies, bee colonies and populations of bacteria, all of which are constituted of a number of entities too simple to understand the whole situation individually. The aim could be described as an attempt to understand how intelligent behaviour of a colony can arise from the actions of a number of relatively simple agents.

In the preface to *Swarm Intelligence* [26] Kennedy and Eberhart make two assertions which indicate the motivation for studying Swarm Intelligence in the context of Intelligent Behaviour. Firstly, *Mind is social*, which demonstrates a belief that an investigation of Swarm Intelligence will provide insight into the behaviour of known intelligent entities such as humans and animals. Secondly, *Particle swarms are a useful computational intelligence (soft computing) methodology*, which demonstrates the belief that practical systems can be developed by using the principles of Swarm Intelligence. These two beliefs, taken together, form a new approach to Artificial Intelligence which is less vulnerable to either of the arguments directed towards the approaches of *Making a Mind*, which was criticised for searching for an architecture for intelligent behaviour that may not exist, and *Modelling the Brain*, which was criticised for potentially offering no insight into the production of intelligent behaviour itself.

Some examples detailed in this section, Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and Dispersive Flies Optimisation (DFO) use a swarm of interacting agents to implement a search algorithm. Search algorithms are among the most fundamental areas of computer science, the task being to retrieve a designated element from a data structure, referred to in this context as a search space. In many cases this can be achieved by simply iterating over every element in the search space and evaluating whether the element is equal to the target element. This naïve approach is often infeasible, for example when the search space is very large, or when it is computationally expensive to perform the evaluation. In cases where certain features of the task definition are known, significant optimisations can be made. For example, in the case of determining whether a specific number is in a list, if the list is known to be sorted then a binary search can be employed bisecting the area of the search space which remains to be searched at each step. In cases

where little or no information is available such as when searching for the maximal output of a chaotic black-box function, there may be no applicable optimisations. In such cases, a random search may be the most suitable approach, as this will evaluate all inputs equally, with no assumptions made as to the best location.

Swarm Intelligence may also provide insights into the behaviour of natural swarms. 'Boids', detailed here is an example of an algorithm which produces natural looking behaviour from simple rules and hence may suggest the method by which the behaviour is performed in nature.

2.3.1 1987 — Boids

Boids is an early computational simulation of the phenomena underlying swarm intelligence, that of complex group behaviour arising from simple agent behaviour. Published by Reynolds in 1987, an apparently natural flocking behaviour can be seen in a group of mobile agents who operate by a few simple rules. The resulting flocking and collision avoidance behaviours can be seen in Figure 2.9, p.59.

Quoting from the original description by Reynolds [48]:

> Stated briefly as rules, and in order of decreasing precedence, the behaviours that lead to simulated flocking are:
>
> - Collision Avoidance: avoid collisions with nearby flockmates
> - Velocity Matching: attempt to match velocity with nearby flockmates
> - Flock Centering: attempt to stay close to nearby flockmates

These rules are simple for an individual with local knowledge to adhere to, but can be seen to produce advanced collective behaviour such as coalescing into a flock, and temporarily separating into two flocks to avoid obstacles or predators. Boids was initially presented at the computer graphics conference SIGGRAPH as a technique for producing realistic animations without explicitly defining the paths of each individual, and is an early example of Swarm Intelligence providing an insight into observed intelligent behaviour.

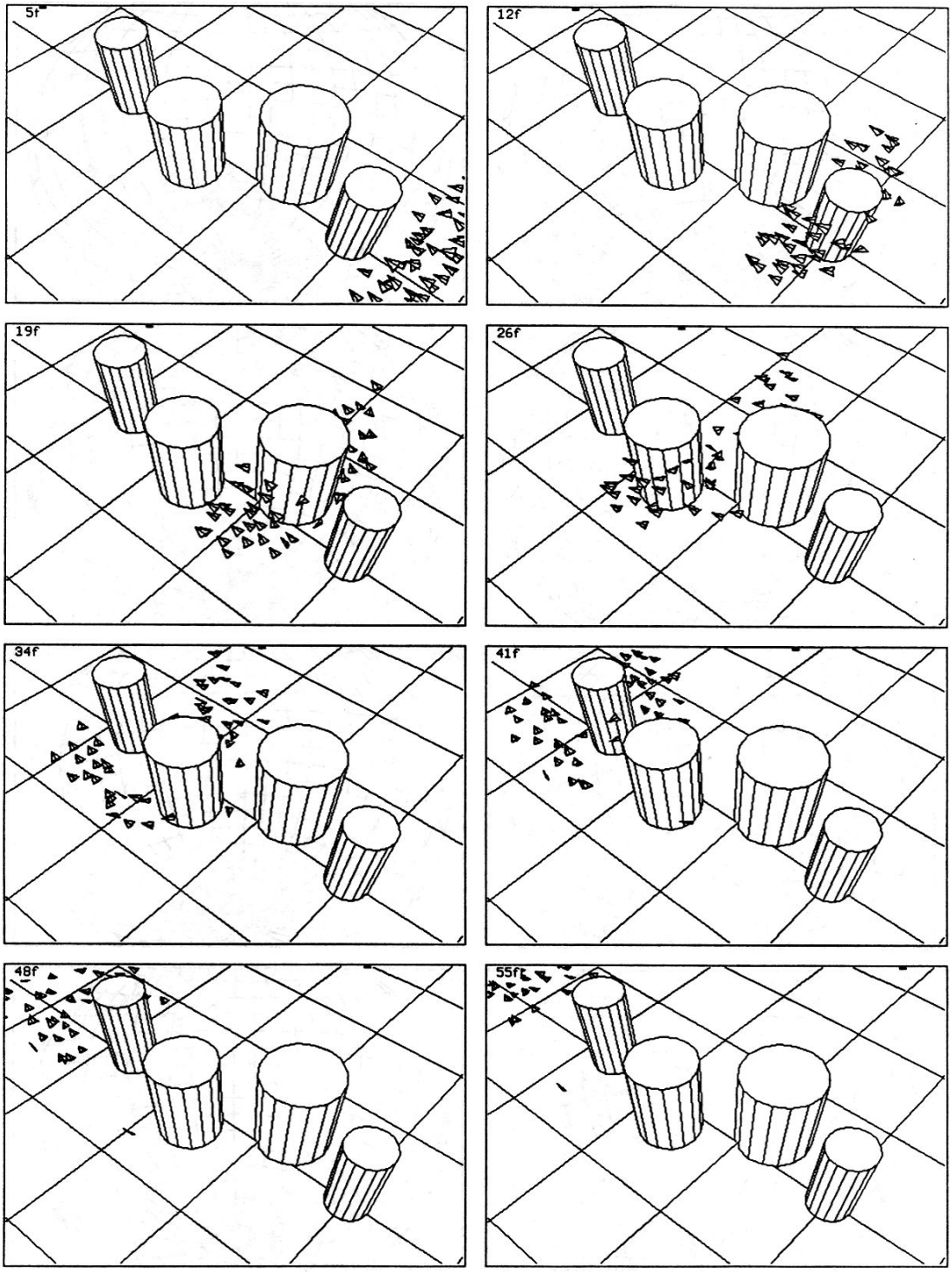

Figure 2.9: A series of frames from an animation depicting A flock of 'Boids' negotiating a field of columns. The image was produced on a Symbolics Lisp Machine in 1986 or 1987. Used with permission of Craig Reynolds, from `https://www.red3d.com/cwr/temp/boids/flock_around_cylinders.jpg`

2.3.2 1992 — Ant Colony Optimisation (ACO)

Ant Colony Optimisation (ACO) (originally called 'the Ant system' [8]) is a swarm intelligence algorithm that is inspired by the foraging behaviour of ants. In ACO, a set of software agents, described as *artificial ants* search for solutions to a given problem, often a combinatorial problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The first application of ACO was to the Travelling Salesman Problem (TSP), where the task is to find the shortest tour, a path which connects all of a set of points distributed in a space. The agents stochastically construct solutions where each element of the solution is influenced by a combination of *a priori* knowledge about the 'desirability' each element and the level of pheromone at that element. In the TSP example each the elements of the solution are the order in which points are visited in the tour, and the *a priori* knowledge is the 'visibility' of each city from each other city calculated as the reciprocal of the distance. A data structure records for each agent the elements which make up its solution, this ensures no agent selects the same element twice in a tour. The exact probability of an agent selecting an edge is the edge's score divided by the score of all edges not yet added to the tour, where the score is calculated by multiplying the desirability associated with that edge raised to the power of a constant parameter β with the amount of pheromone associated with that edge raised to the power of a constant parameter α. Once all agents have generated a solution, the level of pheromone at each edge is calculated by multiplying its current level of pheromone by a constant p in the range of $[0, 1]$ and adding a value which is a function of the score of all solutions which include the edge. In the TSP example, the score of a solution is the reciprocal of the total tour length, weighted by a constant parameter Q. The increase in the amount of pheromone at an edge is analogous to ants leaving a pheromone trail as they perform a tour and leaving a higher concentration of pheromone along shorter tours. The reduction in the amount of pheromone at an edge is analogous to the evaporation of pheromones over time. The result is that the edges which are most likely to be selected are those which combine a low individual cost with a high solution score. This process, shown in algorithm 1 has been shown to produce near optimal solutions to the TSP [13].

A potential disadvantage of ACO is that it requires the selection of six parameters: α, a parameter which weights the importance of the pheromone in the tour construction process; β, a parameter which weights the importance of the 'desirability' in the tour construction

process; p, the rate of evaporation of the pheromone; τ_0, the initial amount of pheromone at each edge; $t_{\max}$, a parameter which determines the number of iterations before halting. Q, a parameter which weights the importance of the score of the solution in the calculation of the amount of pheromone at each edge. While some of these parameters are not critical in determining the performance of ACO, the combination of α and β may determine whether the algorithm exhibits one of three behaviours: i) the algorithm finds good solutions, ii) the algorithm does not find good solutions and all agents maintain the same solution, iii) the algorithm does not find good solutions and all agents do not maintain the same solution [13]. Care must therefore be taken to select appropriate parameters for each task.

Algorithm 1 Ant Colony Optimisation

function ACO(graph, τ_0, p, α, β, Q, $t_{\max}$)
    $t \leftarrow 0$
    for each edge **in** graph **do** ▷ initialise pheromone levels
        amount of pheromone at edge $\leftarrow \tau_0$
    while $t < t_{\max}$ **do**
        for each node **in** graph **do**
            initialise one ant with a path starting at this node
        for each ant **do** ▷ construct a tour
            repeat
                tail $\leftarrow$ last node in path of ant
                unvisited $\leftarrow$ list of nodes not in path of ant
                for each *head* **in** unvisited **do** ▷ calculate scores of unvisited nodes
                    $\tau \leftarrow$ amount of pheromone at edge $(tail \rightarrow head)$
                    $d \leftarrow$ distance between *tail* and *head*
                    $\text{scores}_{head} \leftarrow \tau^{\alpha} \times \left(\frac{1}{d}\right)^{\beta}$
                S $\leftarrow \sum_{i=1}^{n} \text{scores}_i$
                probabilities $\leftarrow \frac{\text{scores}_i}{S}$ for i in scores
                probability of selecting node i is probabilities_i
                $n \leftarrow$ node randomly selected from unvisited
                add n to path of ant
            until path of ant is a complete tour
        for each edge **in** graph **do** ▷ update pheromone levels
            $P_{\text{edge}} \leftarrow$ amount of pheromone at edge
            evaporation $\leftarrow P_{\text{edge}} * p$
            popularity $\leftarrow 0$
            for each ant **do**
                if edge exists in path of ant **then**
                    $L^k \leftarrow$ length of tour of ant
                    popularity $\leftarrow$ popularity + $\frac{Q}{L^k}$
            amount of pheromone at edge $\leftarrow$ evaporation + popularity
        $t \leftarrow t + 1$
        if all ants have the same tour **then** ▷ early halting condition
            break
    return shortest tour

ACO is a stigmergic procedure, where the decisions of agents are guided by modifications made to the search space by other agents, in this case, with pheromone trails. While the exact actions of an agents in ACO are not claimed to be within the capacities of real ants the entire behaviour can be taken as analogous of behaviour which. For example, when constructing a tour an agent in ACO is required to select a edge from a probability distribution constructed by evaluating polynomial expressions, whereas an ant in nature could conceivable achieve a similar behaviour by combining any visible information with pheromonal information of the available options and apply some internal, possible instinctive, bias. ACO therefore demonstrates the possibility of simple agents being able to solve a complex combinatorial problem without ever considering the problem as a whole.

2.3.3 1995 — Particle Swarm Optimisation (PSO)

A subsequently published swarm intelligence algorithm was Particle Swarm Optimisation (PSO) [18]. A swarm of agents, named *particles*, are distributed through a search space and evaluate the solution represented by their current location. The initial locations of the agents are chosen uniformly randomly or with influence from external domain knowledge, but subsequently chosen by moving the agents in a direction which is a combination of its previous movement and information derived from the swarm as a whole. The broad principles of operation can be illustrated by the following metaphor [7].

> Imagine that one dips a finger in a jar of honey and uses that honeyed-finger to trace a line in the air in the presence of a swarm of bees. Fairly soon the bees will begin to buzz around the honeyed finger tip. By using this swarm we can generate a new series of points (defining a new line) simply by noting the centroid of this swarm at successive instants in time.

In this metaphor line defined by the centroid of the swarm at successive instants in time provides an estimate of the position of the honeyed finger over time. For this behaviour to emerge all that needs to happen is for each bee to move as a function of two distances. These distances are: the distance from the bee to the position representing the closest the bee has ever been to the honey, and the distance from the bee to the position representing the closest any bee has ever been to the honey. This is an efficient way of guiding a search through a space without depending on expert domain knowledge being encoded into the

implementation. In contrast with Boids, PSO is a useful search algorithm but is unlikely to provide an insight into the behaviour of natural swarms as all agents need to have access to the swarm's 'global best' fitness.

Algorithm 2 PSO

```
1:  Initialise agents
2:  while not halting condition do
3:      for each particle in swarm do
4:          fitness ← evaluate fitness of particle          ▷ Larger is better
5:          if fitness > particle's personal best then
6:              particle's personal best ← fitness
7:              if particle's personal best > swarm's best then
8:                  swarm's best ← particle's personal best
9:          Update particle velocity                        ▷ Equation 2.2, p.63
10:         Update particle position                        ▷ Equation 2.1, p.63
```

The PSO evolution functions define each agents subsequent position x_{t+1} (Equation 2.1) and subsequent velocity v_{t+1} (Equation 2.2). w is the constant inertia weight, c_1 and c_2 are the cognitive and social learning parameters respectively, r_1 and r_2 are randomly generated numbers between 0 and 1, p_t^i is the particle's personal best position and p_t^g is the swarm's best position.

$$x_{t+1} = x_t + v_{t+1} \tag{2.1}$$

$$v_{t+1} = wv_t + c_1 r_1 (p_t^i - x_t) + c_2 r_2 (p_t^g - x_t) \tag{2.2}$$

2.3.4 2014 — Dispersive Flies Optimisation (DFO)

Inspired by the behaviours of flies hovering over food sources, Dispersive Flies Optimisation (DFO) [49] exhibits a swarming behaviour similar to PSO, but with the extra feature of swarm dispersal, analogous to when natural swarms are dispersed by the presence of a threat, as in nature the swarm will reappear at the original position as the threat moves away, unless a preferred location is found in the interval. To simulate this, in each iteration any component of a fly's position may be reset with a fixed probability, known as the *disturbance threshold*. This guarantees a proportionate disturbance from a solution and hence avoids the swarm becoming trapped in a local minima.

In DFO each individual fly updates its position by updating each dimensional component

in turn, the update is calculated as the distance to the highest scoring of a fly's two neighbouring flies plus a portion of the distance to the highest scoring fly in the whole swarm, the portion being a randomly selected value in the interval $[0, 1]$. As the magnitude of any updates are relative to the distances between flies, the algorithm naturally begins with an emphasis on exploration of the search space. As the swarm begins to converge around a single location, and the distances between the flies decrease, the algorithm begins a more exploitative phase as a promising solution is more precisely optimised.

A feature of DFO is that it is a minimalistic algorithm, in this sense it has been designed to have the smallest number of tunable parameters and the smallest number of components [51]. Other than the population size, DFO has one tunable parameter, the disturbance threshold. Additionally, other than the index of the best fly, each fly consists of merely its position vector along with the index of the best neighbouring fly.

2.4 Summary

The project of building machines with diverse behaviour has made significant progress since ancient times, from fully defined Clepysdra, though configurable mechanisms, and reaching maturity with Turing's model of computation which enabled the study of Artificial Intelligence. AI itself has made significant progress but has been characterised as mostly involving projects which can be described as Making a Mind or Modelling the brain. The approach of making a mind is characterised as the use of symbolic logic to perform explicitely defined tasks. It has been criticised as the symboloc logic required appears to become vastly more complicated as the requirements of the task become larger, and less well defined. Modelling the brain is characterised by Connectionist architectures and neural networks, which have also demonstrated great abilities and a certain level of genralisation or adapability. It was criticised on the grounds that even a successful project may lack any insight into the capacities which made a system adaptive and intelligent.

Swarm intelligence is identified as being able to be described as combining both approaches. Father than fully defining behaviour of a system behaviour with an algorithm of symbolic logic, or fully relying on emergent behaviour from a network of reactive elements, swarm intelligence defines a population of interacting agents, each controlled by an internal algorithm, but an algorithm which determines behaviour as the result of interactions

within the swarm.

This implies that swarm intelligence will be susceptible to the criticisms of both making a mind and modelling the brain, but by being fully defined by neither approach, they may be mitigated somewhat.

We will investigate an example of swarm intelligence which has simple agents performing arbitrary symbolic logic, but which uses the interaction of the simple agents to gain determine the behaviour, we will also see the disadvantages that are present with this approach. We will also see the analogies between its operation and natural behaviour which, if strong, will lend some evidence towards there being value in the method.

Chapter 3

Stochastic Diffusion Search (SDS)

Algorithm 3 SDS as introduced by Bishop (1989)

```
INITIALISE (mappings);
REPEAT
      TEST (mappings);
      DIFFUSE (mappings);
UNTIL TERMINATION;
```

3.1 Introduction to SDS

First described in 1989 by J.M. Bishop, Stochastic Diffusion Search (SDS) was introduced as an evolution of the study of Hinton mapping [5] (Section 2.2.3, p.50). Where Hinton mapping identifies objects by testing for all possible combinations of stimuli, SDS stochastically tests certain hypotheses against partial combinations of stimuli, and subsequently performs a diffusion process where more promising hypotheses are more likely to be tested against further features. At the time, and at least as late as 1998 [41], the term "Stochastic Search Networks" was used, but SDS will be used throughout.

SDS has a number of important features, including; robustness in the presence of noise; a simplicity that lends itself well to mathematical modelling; and stable global state emerging from the stochastic behaviour of individuals. Its robustness make SDS particularly suited

to real world tasks which may be noisy, and may change over time. The relative ease of modelling SDS mathematically has lead to the ability to predict a number of aspects of its behaviour. The stability of SDS means it may also be an important model for methods of guiding intelligent behaviour without a central executive control.

The operation of SDS is most intuitively explained by analogy, a number of which have been used, most commonly *The Restaurant Game* [11], and more recently *The Mining Game* [54]. These are described below, followed by the pseudocode of a single iteration of SDS.

3.1.1 The Restaurant Game

The Restaurant Game, reproduced here from 2003 [11], describes how a group of people may employ SDS to perform a search for the best restaurant in an unfamiliar town.

> A group of delegates attends a long conference in an unfamiliar town. Each night they have to find somewhere to dine. There is a large choice of restaurants, each of which offers a large variety of meals. The problem the group faces is to find the best restaurant, that is the restaurant where the maximum number of delegates would enjoy dining. Even a parallel exhaustive search through the restaurant and meal combinations would take too long to accomplish. To solve the problem delegates decide to employ a Stochastic Diffusion Search.
>
> Each delegate acts as an agent maintaining a hypothesis identifying the best restaurant in town. Each night each delegate tests his hypothesis by dining there and randomly selecting one of the meals on offer. The next morning at breakfast every delegate who did not enjoy his meal the previous night, asks one randomly selected colleague to share his dinner impressions. If the experience was good, he also adopts this restaurant as his choice. Otherwise he simply selects another restaurant at random from those listed in 'Yellow Pages'.
>
> Using this strategy it is found that very rapidly [a] significant number of delegates congregate around the best restaurant in town.

The process of the restaurant game has a number of notable features. Within minimal centralised control the group of delegates acts together to solve a problem that could not be quickly solved by an individual. The delegates will efficiently move to the next best

restaurant if the current one has a significant drop in standards or closes down entirely. The restaurants, the menus, or the individual meals need to be directly comparable, all that is required is for each agent to decide for themself whether their experience was good. Delegates will find themselves enjoying many evenings in a relatively high quality restaurant long before all of the meals in all of the restaurants in town could have been evaluated.

This analogy has been criticised on the grounds that delegates are likely to have differing dining preferences and hence it is possible for a delegate to locate a restaurant in which they enjoy all of the meals on offer, but which is unsatisfying to all other delegates. In the case where only one delegate, or a small proportion of the group, remains permanently at such a restaurant the rest of the group will proceed largely as usual and so the majority will still converge on the best restaurant. Taken to the extreme, however, all agents may find themselves dining alone, even when there exists a single superior restaurant which would satisfy the largest portion of the delegates. This superior restaurant would never be located as all delegates are satisfied with the meals at their current restaurant, and hence never select a new restaurant. This critique led to the development of *The Mining Game*, which depends on the less subjective notion of locating gold rather than dining preferences.

3.1.2 The Mining Game

Originally defined in 2010 [53] the Mining Game uses an analogy of searching a landscape for the best mining prospect. A slightly improved version from 2013 [52] is quoted below.

> A group of friends (miners) learn that there is gold to be found on the hills of a mountain range but have no information regarding its distribution. On their maps the mountain range is divided into a set of discrete hills and each hill contains a discrete set of seams to mine. Over time, on any day the probability of finding gold at a seam is proportional to its net wealth.
>
> To maximise their collective wealth, the miners need to identify the hill with the richest seams of gold so that the maximum number of miners can dig there (this information is not available a-priori). In order to solve this problem, the miners decide to employ a simple Stochastic Diffusion Search.
>
> - At the start of the mining process each miner is randomly allocated a hill

to mine (his hill hypothesis, h).

- Every day each miner is allocated a randomly selected seam on his hill to mine.
- At the end of each day, the probability that a miner is happy[1] is proportional to the amount of gold he has found.
- At the end of the day the miners congregate and over the evening each miner who is unhappy selects another miner at random to talk to. If the chosen miner is happy, he happily tells his colleague the identity of the hill he is mining (that is, he communicates his hill hypothesis, h, which thus both now maintain). Conversely, if the chosen miner is unhappy he says nothing and the original miner is once more reduced to selecting a new hypothesis — identifying the hill he is to mine the next day — at random.

In the context of SDS, agents take the role of miners; active agents being 'happy miners', inactive agents being 'unhappy miners' and the agent's hypothesis being the miner's 'hill-hypothesis'. It can be shown that this process is isomorphic to SDS, and thus that the miners will naturally self-organise and rapidly congregate over hill(s) on the mountain range with a high concentration of gold.

The happiness of the miners can be measured probabilistically, or represented with an absolute boolean value, so long as each miner is either happy or unhappy by the end of each day [50]. Furthermore, if the gold is modelled as a finite resource, reducing over time, then the search is sufficiently adaptive that miners change where they congregate as the location with the most gold changes.

3.1.3 Pseudocode

In this example, agents have access to a function for randomly selecting a hypothesis for the set of all possible hypotheses, and for randomly selecting a microtest from a given set. Each microtest evaluates a specific part of a hypothesis and returns a boolean value indicating whether or not the test passed.

[1] Though 'happy' is a term that is similarly subjective as one's dining preferences, in this case it is used in an objective sense. All miners share an identical process whereby the amount of gold they locate on a single

Algorithm 4 A single iteration of standard SDS

```
1:  for each agent in swarm do                                  ▷ Diffusion phase
2:      if agent is inactive then
3:          polled ← randomly selected agent in swarm
4:          if polled is active then
5:              agent assumes the hypothesis of polled
6:          else
7:              agent randomly selects a new hypothesis

8:  for each agent in swarm do                                  ▷ Test phase
9:      microtest ← randomly selected microtest
10:     agent performs the microtest against their hypothesis   ▷ Partial evaluation
11:     if the microtest passes then
12:         agent becomes active
13:     else
14:         agent becomes inactive
```

To recap definitions. An *agent* is an individual member of a swarm. A *swarm* is a collection of agents. A *search space* is the set of all possible hypotheses. A *hypothesis* is an element of the search space, or an index referring to an element of the search space. A *microtest* is a function which partially evaluates a hypothesis. The *score* of a hypothesis is equal to the probability of an agent with that hypothesis being active after the test phase. A full glossary is available in Chapter 6, p.174.

SDS lends itself well to mathematical analysis. In the next section a simple but powerful model of the convergence behaviour of SDS is described. The features of SDS which have previously been modelled and the insights into its behaviour using this model are then described. Following this a formalism for describing variants of SDS is developed and some of the advantages and disadvantages of all the main variants which have been published are identified.

day defines a probability that the miner will declare themselves 'happy' at the end of the day when miners congregate to potentially share the identity of the hills they are mining.