UNIVERSITY OF TRENTO

SCHOOL OF SOCIAL SCIENCES
PHD PROGRAM IN ECONOMICS AND MANAGEMENT

DOCTORAL THESIS

STUDIES IN CLASSICAL BEHAVIOURAL ECONOMICS

A DISSERTATION SUBMITTED TO THE DOCTORAL PROGRAM
IN ECONOMICS AND MANAGEMENT IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DOCTORAL DEGREE
(PH.D) IN ECONOMICS AND MANAGEMENT

Author:
Ying-Fang Kao

Supervisor:
Prof. K. Vela Velupillai
Co-Supervisor:
Prof. Stefano Zambelli

December, 2013

ACKNOWLEDGEMENT

My doctoral thesis work can not begin and terminate without many people's valuable help, incisive and constructive suggestions, intellectual inspirations and encouragements. I would like to thank all the friends who had contributed in various ways to this thesis. None of the them are responsible for the infelicities in this thesis.

I am immensely grateful and indebted to my supervisors. I have learnt priceless lessons from them, in terms of academics and in life. Prof. Velupillai introduced me to the world of Classical Behavioural Economics among his many other specialities. He guided me with his vast knowledge, and through his extremely disciplined and scholarly commitment towards academic pursuits. Professor Zambelli, himself a Chess master, had contributed many penetrating insights to my aspiration of building a Go program. As mentors, they set an example and inculcated good attitudes towards academic adventures.

I would like to express my special gratitude to my thesis examiners, Prof. Bruna Ingrao and Prof. Cassey Lee and the doctoral committee for their insightful comments and positive feedback.

I would like to thank Prof. Shu-Heng Chen Prof. Shu G. Wang ,and Miss Connie Wang for their untiring and generous support for my pursuit of this PhD program and academic career. They are my academic family back in Taiwan. Without them, I would never have got this opportunity to study in Trento.

V. Ragupathty, my colleague and an important friend, had painstakingly and patiently read through my thesis cover to cover and made it linguistically readable before it was presented to anyone. Besides the editing work, Ragu has provided constructive suggestions and research ideas through his sheer curiosity and illuminating insights.

I would also like express my special gratitude to some of my colleagues: Ivana Catturani and N. Dharmaraj, and Kiryl Haiduk, who helped me as a PhD student and a foreigner in Trento in many ways and did so with generosity. Many good friends and classmates whose eventful company made my time in Italy colourful and adventurous. In particular, I would like to thank Fabio Salvadei, Gabriele Romagnoli, Valentina Viero, Simone Phuderer, Anna Bielova, Lucia Leonardi, Enrico Tundis, Carlo Campangnoli, Kenneth Beyerlein, Ashley Lin, Jin Ran, Kao Wen-Zheng, Silvia Lo Sapio, Veronica Pederzolli and the Wild Voices.

Finally, I owe the most to my family in Taiwan. They are extremely understanding and loving. Without their unconditional support, my journey in Italy could not be so enjoyable. My late father Y-C. Kao, who was widely travelled, was the most excited about my decision to study abroad. He and I enjoyed having conversations about our respective foreign and cultural experiences. My mother Jeanne Lin has been the pillar of my family. Her strength and optimism in facing difficult times is the source of my willpower. My younger brother and a close friend, Kajer Kao, has been a patient listener of my ramblings regarding my research. I dedicate this thesis to my family.

ABSTRACT

This thesis distinguishes Classical Behavioural Economics (CBE) from Modern Behavioural Economics (MBE) and discusses Herbert Simon's pioneering contributions to CBE in detail. CBE emphasises the role of *bounded rationality*, *satisficing* and *heuristics* (procedures) in human decision making in contrast to *optimization*, which is widely used in MBE. In the framework of CBE, heuristics are algorithms, which are embedded in the Information Processing Systems (IPS) in Human Problem Solving (HPS) environments. The argument here is that the premise of CBE is highly suitable for algorithmic modelling of adaptive agents facing complex economic environments. It explores the theoretical foundations of *bounded rationality* and substantiates the difference between satisficing and optimization, and claims that the former is more general than the latter. These investigations are carried out from the perspective of Computability Theory and Computational Complexity Theory, which provide a coherent foundation for Herbert Simon's work on problem solving and for CBE in general. The second part of the thesis explores the game of Go from the perspective of problem solving in CBE. Its generality, suitability and complexity are analysed in detail. A pseudo IPS which is capable of playing Go is also constructed based on the insights from Simon's work on *protocol analysis* and information from Go documentaries. The claim of this thesis is that Go can be a powerful paradigm to better understand human economic behaviour in diverse and complex environments.

Key Words: Classical Behavioural Economics, Bounded Rationality, Satisificing, Heuristics, Computational Complexity, Information Processing System, Human Problem Solving, Game of Go

Contents

Acknowledgement **i**

Abstract **ii**

1 Introduction **1**

2 Behavioural Economics: Classical and Modern **5**
2.1 Overview . . . 5
2.2 Emergence of Behavioural Economics . . . 7
2.2.1 Modern Behavioural Economics . . . 7
2.2.2 Classical Behavioural Economics . . . 13
2.3 Underpinnings of Behavioural Economics . . . 15
2.3.1 Underpinnings of Modern Behavioural Economics . . . 16
2.3.2 Underpinnings of Classical Behavioural Economics . . . 21
2.4 CBE - Notes on the Special Role of Herbert Simon . . . 23
2.4.1 Bounded Rationality . . . 24
2.4.2 Human Problem Solving . . . 26
2.4.3 Classical Behavioural Economics and Computable Economics . . . 28
2.5 Concluding Remarks . . . 30

3 Computable Foundations of Bounded Rationality **33**
3.1 Introduction . . . 33
3.2 Bounded Rationality . . . 35
3.2.1 Simon's bounded rationality . . . 36
3.3 From Turing to Simon: decision making as problem solving . . . 42
3.4 Computable Foundation of Bounded Rationality . . . 44
3.4.1 Solvable and Unsolvable Problems . . . 44
3.4.2 Computational Complexity . . . 49
3.4.3 Computable Economics . . . 53
3.4.4 Human Problem Solving . . . 56
3.5 Concluding Remarks . . . 57

4 Human Problem Solving in Classical Behavioural Economics **59**
4.1 Overview . . . 59
4.2 Human Problem Solving . . . 62
4.2.1 Theory of Human Problem Solving . . . 63
4.2.2 Information Processing Systems . . . 66
4.2.3 Newell, Shaw and Simon (NSS) Chess Program and Onward . . . 69
4.3 Go versus Chess . . . 73
4.3.1 Artificial Intelligence . . . 74
4.3.2 Decision Making in CBE and Board Games . . . 76
4.4 Concluding Remarks . . . 80

5 The Game of Go: An IPS Interpretation **83**
5.1 Overview . . . 83
5.2 Game of Go . . . 84
5.2.1 Go Rules . . . 85
5.2.2 The Philosophical Reflections of Go . . . 88
5.3 Problem Solving of Go . . . 90
5.3.1 Problems and Task Environment . . . 91
5.3.2 Heuristics: evidences from Go literate and documentations . . . 92
5.3.3 The hierarchies in decision making of Go . . . 104
5.4 A Pseudo IPS of playing Go: COMPOSER . . . 106
5.4.1 Highlights of Components and Characteristics of COMPOSER . . . 106
5.4.2 Discussions about Go Terms and Concepts . . . 112
5.4.3 Descriptions of Composer . . . 114
5.5 Concluding Remarks . . . 118

6 Conclusion **119**

A Go terminology **123**

References **125**

Chapter 1

Introduction

"What a person *cannot* do he or she *will* not do, no matter how strong the urge to do it."
Simon (1996b), p. 28

A behavioural approach to economic analysis is slowly becoming an integral part of the discipline. A simple, though not rigorous, way to characterize this approach would be to say that it pays explicit attention to the behavioural aspects - cognitive, psychological, emotional - of the agents, organizations and entities in the economic sphere. This is in sharp contrast to the established wisdom about how agents behave within economic theory -*rational maximizers*. The behavioural revolution in economics is often traced back to Herbert Simon's pioneering article titled "A Behavioural Model of Rational Choice" in 1955. There have been enormous developments in a multitude of directions for the last 60 years or so since then. However, the behavioural approach to economics cannot be considered as a homogeneous entity and the theories classified under this umbrella vary substantially in many respects, such as, their methodology, philosophy, tools and even the points of emphasis.

The issue of how to meaningfully characterize an economic agent is at the heart of the differences between various approaches and an appropriate notion of 'rationality' that underpins his/her decisions has generated enormous debate. Limits or *bounds* to rationality can be considered as a common thread that runs across different strands of behavioural economics. The term 'bounded rationality' itself can be traced back to Herbert Simon. These theories can be broadly classified into two categories based on their methodological approach: Classical and Modern Behavioural Economics - a distinction that was made by Velupillai (2010a). The former is along the lines forged by Herbert Simon, while the modern version can be thought of as being along the lines forged by Ward Edwards, Daniel Kahneman, among others. A more detailed attempt to trace their origins and spell out the differences can be found in the first part of this thesis.

When we observe how human decision makers around us actually behave, optimization as a general formalism can be easily disputed. However, this is not a straightforward and small step for Economics. Behavioural Economics has emerged from various evidences that are used to challenge the axioms of preference ordering in orthodox economic theory. Modern Behavioural Economics has made important contributions in terms of challenging orthodoxy by pointing out anomalies that lie within this theory. However, it has only made peripheral remedies since it tries to fix the problem of anomalies without challenging the existing paradigm of optimization. One the other hand, in Classical Behavioural Economics the economic agents are viewed as *boundedly rational* - more precisely, 'procedurally' rational, information processing systems, with no reliance on optimization. Bounded Rationality is neither sub-optimality nor irrationality and the behaviour of agents is to be investigated through the possible and actual *procedures* involved in decision making.

This thesis deals with some crucial themes in Classical Behavioural Economics and analyses them from different perspectives. At the outset, it needs to be mentioned that this thesis takes a firm stance that economic agents, solving decision making problems, should not be formalized as optimizing agents. Instead, they should be modelled as adaptive *satisficing* agents facing complex environments. However, this distinction cannot be made convincingly without elaborating on the foundations, premises, methodologies and applications. Within the framework of Classical Behavioural Economics outlined in this thesis, crucial notions such as bounded rationality, satisficing and heuristics - ideas that Herbert Simon pioneered and stressed all his life, can be faithfully studied and understood in ways that Simon envisaged.

In Classical Behavioural Economics, there is no predetermined preference ordering over a given choice set of a problem for an economic agent. Instead, the agent is expected to adaptively generate methods to explore the problem space and discover new elements and develop procedures. Besides, the alternatives tend to be compared qualitatively, rather than quantitatively in many cases. Though the agents will gradually know more elements in the set, he(she) has limited attention because of computational constraints on cognitive faculties; therefore the focus is only on a subset of the choices that he(she) explores. That is to say, human reasoning is bounded by his(her) limited cognitive capacity.

For Herbert Simon, bounded rationality, satisficing and heuristics are inherently algorithmic notions. He has developed an approach - Human Problem Solving - to understand the *process* of human decision making. In this framework, the *satisficing* agent is formally characterized as an Information Processing System and the algorithmic aspects of decision making (problem solving) are extensively explored. Classical Behavioural Economics, in turn, inherits Simon's approach on Human Problem Solving and explores and connects it to its algorithmic underpinnings. This is achieved by means of computability theory and computational complexity theory.

This thesis heavily relies on Herbert Simon's widely celebrated, but often misunderstood contributions. He won numerous prizes and honors for his fundamental contributions - Turing Award in 1975 and Nobel Prize in Economic Science in 1978 - to mention a few. Herbert Simon can be considered as a quintessential problem solver, even in his way of doing academic research. He had broad interests in the hard sciences and fascination for applying mathematical tools to the social sciences.

However, he always had penetrating insights in to the problems and the mathematical tools were mere aids to solve these problems and not dictating terms. Being a genuine multi-disciplinarian, Simon can be regarded as a scientist in many disciplines. In my understanding, Simon practiced problem solving throughout his professional and intellectual life. When he was curious about some phenomena in his mind, he searched for solutions and most of the time he did so by moving across different disciplines. Either by corresponding and collaborating with academic friends or by himself, he made profound contributions in each sphere in which he investigated. On the journey of solving one problem, he discovered sub-problems unexpectedly. Eventually, he had been traveling around different science-mazes (Simon's metaphor) to the extent that even he could not anticipate. He ended up as a scholar having both depth and breadth at the same time. For him, problem solving is not merely solving equations or puzzles. Rather it is the process of identifying alternative methods or sub-problems, knowing the possible directions for finding methods, and deciding where attention ought to be employed. It is the core of his view of organizations, economics and his life. It should be noted that for Simon, human problem solving and behavioural economics are not disconnected, distinct entities. His approaches across these two areas have a uniform theme that unites them. This is the notion of 'problem solving', boundedly and procedurally rational, information processing agent who is engaged in satisficing.

Motivated by Simon's special role in economics, and his influential contributions, I aspire to incorporate the notions of Bounded Rationality, satisficing, sequential decision making and heuristics, into modelling problem solvers(economic agents). I study problem situations in the game of Go, and thereby extending Herbert Simon's paradigm of Chess to understand problem solving. The game of Go, which has a finite and large search space, provides a rich paradigm for analysing how human beings solve highly complex problems. This thesis is structured into four chapters.

Chapter 2 formally distinguishes Classical and Modern Economics through their respective origins, theoretical foundations, methodology and applications. The origins and development of two kinds of behavioural economics, beginning with the pioneering works of Simon (1953a) and Edwards (1954), are traced and (critically) discussed. The mathematical foundations of classical behavioural economics are identified, largely, to be in the theory of computation and computational complexity; the mathematical basis for modern behavioural economics is claimed to be a notion of subjective probability. Individually rational economic theories of behaviour, with attempts to broaden - and deepen - the notion of rationality, challenging its orthodox variants, were decisively influenced by these two mathematical underpinnings.

Chapter 3 aims to interpret and formalize Herbert Simon's notion of bounded rationality in the context of computability theory and computational complexity theory. Simon's theory of Human Problem Solving is analysed in the light of Turing's work on Solvable and Unsolvable problems (Turing, 1954). I analyse Simon's models of rational behaviour, emphasizing the characteristics of environments and decision makers, providing rigorous algorithmic interpretations of Simon's concepts - bounded rationality, satisficing, heuristics. Within this framework, it is pointed out that Olympian rationality requires human beings to go beyond Turing computability and computational complexity defines the inner boundary of algorithmic rationality. It is highlighted that bounded rationality results from the fact that the deliberations required for searching complex spaces exceed the actual complexity that human beings can handle. The immediate consequence is that satisficing becomes the general criterion of decision makers and heuristics are the methods for achieving their goals. In such decision problems, it is demonstrated that bounded rationality and satisficing are more general than Olympian rationality (coined in Simon (1983b), p.19) and optimization respectively and not the other way about.

Chapter 4 introduces more detailed content of Information Processing Systems in Human Problem Solving approach. Information Processing Systems vary according to different task environments. However, the essential architectures of IPS , such as the associativity of Long-Term Memory, the limited capacity of Short-Term Memory and Production Systems, are invariant. Classical Behavioural Economics, on one hand, heavily emphasizes the severely limited computational capacity of economic agents, one the other hand, is keen to characterize the general structure of IPS that can be readily extended to complex economic problems. The chapter argues that the game of Go, which is qualitatively consistent with many real-life problems, is a potential domain for CBE to characterise Human Problem Solving approach to complex economic problems.

Chapter 5 initiates an attempt to build an pseudo Information Processing System, which can explain Go players' qualitative behaviour. Go players' skill in solving problems reflects their ability to comprehend, communicate and reason with publicly adopted Go terms. These Go terms differ largely from the idea of chunks, which constitute the domain-specific knowledge for Chess problem solving. From the evidence on how Go players benefit from using and organising the Go terms, a more general IPS that contains a higher-level production system is devised. The insights obtained in this chapter can be suitably extended to economic domains, especially to study organizational decision making. In the conclusion, several future directions and issues are proposed for going beyond the scope of this thesis.

Chapter 2

Behavioural Economics: Classical and Modern[1]

2.1 Overview

> "Let us call [the bounded rationality] model of human choice the *behavioral model*, to contrast it with the Olympian model of SEU[2] theory. Within the behavioral model of human rationality, one *doesn't have to make choices that are infinitely deep in time*, that encompass the whole range of human values, in which each problem is interconnected with all the other problems in the world. ...Rationality of the sort described by the behavioral model *doesn't optimize*, of course. Nor does it even guarantee that our decisions will be consistent."
>
> Simon (1983b), p. 19-23, italics added

Behavioural economics may have, finally, come of age. It is part of the curricula of graduate schools in economics, finance and management, even one of the compulsory courses . More than a decade ago, in a letter to Velupillai (see Velupillai (2010a), p.407-408; italics added), Herbert Simon was optimistic enough to state, after a half-a-century tireless efforts to make behavioural economics a viable alternative to orthodox neoclaasical economics, that:

> "The economists here [at Carnegie Mellon University] remain, for the most part, ...backward ..., but I am encouraged by the great upswell, in the US and especially in Europe, of experimental economics and various forms of bounded rationality. I think the battle has been won, at least the first part, although it will take a couple of academic generations to clear

[1]This chapter unifies and extends several themes and ideas that are present in Kao and Velupillai (2012b); Kao et al. (2012); Kao and Velupillai (2012a, 2013). These papers are a result of a collaborative work with my colleagues in the Algorithmic Social Sciences Research Unit (ASSRU).

[2]Subjective Expected Utility

> the field and get some sensible textbooks written and the next generations trained."

Yet, not much more than one year earlier, at the **84th Dalhem Workshop** on *Bounded Rationality: The Adaptive Toolbox* (Gigerenzer and Selten, 2001, p.ix), two distinguished economists claimed:

> "Bounded rationality, needs to be, but it is not yet, understood."

How, one may legitimately ask, can a 'battle [have] been won', with a crucial concept lying at the foundation of its 'armory', yet to be understood? We believe there is a case for Gigerenzer and Selten to feel that the notion of bounded rationality remains to be clarified. This is because they have been meticulous in having dissected the way the notion has been (ill-) defined by varieties of orthodox theorists , including those we shall shortly identify as some of the pioneers of modern behavioural economics. Moreover, they have also understood, with impeccable perspicacity, that boundedly rational behaviour has nothing to do with either optimization, or irrationality (ibid, p.4).

Where we differ with Gigerenzer and Selten is their anchoring of bounded rationality and satisficing in 'fast and frugal stopping rules for search' without, however, providing this anchor a solid foundation in itself. Bounded rationality and satisficing, in our frame work, is a natural outcome of replacing optimization with *decision problems* (in its *metamathematical* senses), whereby problem solving , in general, and human problem solving in particular, lead to structured search in computationally complex spaces that are classified in terms of *solvability*, *decidability* and *computability*. Optimization becomes a very special case of the solvability of a decision problem, intrinsically coupled to algorithms, which are given measures of complexity that are capable of encapsulating the notions of 'fast and frugal' in precise ways.

In many ways the work of Gigerenzer comes closest to our work on classical behavioural economics, with one important caveat: we identify the notion of heuristics with the formal recursion theoretic concept of algorithms, and hence subject to the Church-Turing Thesis (cf. Velupillai, 2000).

The rest of the chapter is structured as follows. A broad brush discussion of the two kinds of behavioural economics is provided in section 2.2. Next, the analytical foundations of modern and classical behavioural economics are discussed and dissected in section 2.3. Section 2.4 is devoted to a discussion of the special role played by Herbert Simon in forging, *ab initio*, classical behavioural economics and its rich vein of characterizing subfields. The concluding section suggests ways of going forward with a research program in classical behavioural economics - eventually with the hope of exposing the lacunae in the foundations of modern behavioural economics, and its *ad hockeries*.

2.2 Emergence of Behavioural Economics

Behavioural economics, which originated, almost fully developed, during the 1950s, can be classified into at least two streams - *Classical and Modern*. The former was pioneered by Herbert Simon and the latter by Ward Edwards, respectively. The two streams are clearly distinguishable on the basis of their methodological, epistemological and philosophical aspects. Despite sharp contrasts in their approaches to understand (rational) human behaviour, a clear distinction between them was not made until recently (Velupillai, 2010a). Behavioural economics, *in general*, challenges orthodox economics theory and its foundational assumptions regarding human behaviour, its institutional underpinnings (especially in its Classical versions pioneered by Simon), its poor prediction power and its intrinsic non-falsifiability.

The distinctions between *Modern* Behavioural Economics (henceforth MBE) and *Classical* Behavioural Economics (henceforth CBE) can be classified into three aspects. First, MBE assumes economic agents maximizing utility with respect to an underlying preference order - to which 'an increasingly realistic psychological underpinning' is attributed (Camerer et al., 2004, p.3); CBE assumes no underlying preference order and an economic agent's decision making behaviour, at any level and against the backdrop of every kind of institutional setting, is subject to bounded rationality and exhibits satisficing behaviour. Put another way, MBE *remains within the orthodox neoclassical framework of optimization under constraints*; CBE is best understood in terms of decision problems (in the metamathematical sense, cf. Velupillai (2010a)). Second, MBE concerns the behaviour of agents and institutions in or near equilibrium[3]; CBE investigates disequilibrium or non-equilibrium phenomena. Third, MBE accepts mathematical analysis of (uncountably) infinite events or iterations, infinite horizon optimization problems and probabilities defined over σ-algebras and arbitrary measure spaces[4]; CBE only exemplifies cases which contain finitely large search spaces and constrained by finite-time horizons.

2.2.1 Modern Behavioural Economics

Origins

> "The combination of subjective value or utility and objective probability characterizes the expected utility maximization model; Von Neumann and Morgenstern defended this model and, thus, made it important, but in 1954 it was already clear that it too does not fit the facts. Work since then has focussed on the model which asserts that people maximize the product of

[3] The 'near' is defined, in all case we are aware of, by uncomputable approximation processes of uncomputable equilibria.

[4] The most intuitive definition of this essentially measure theoretic concept would be to define it as a non-empty class of sets, closed under the formation of complements and countable unions. However, the kind of subjective probabilities defined by de Finetti and Ramsey avoided, for epistemological and methodological reasons, consideration of events as subsets in a σ-algebra

utility and subjective probability. I have named this the subjective expected utility maximization model (SEU model)."
Edwards (1961), p.474

The origins of Modern Behavioural Economics are often claimed to have emanated from the early works by Richard Thaler, along with Kahneman and Tversky, for example in the following quote:

> "Kahneman and Tversky provided the raw materials for much of behavioral economics - a new line of psychology, called behavioral decision research, that draws explicit contrasts between descriptively realistic accounts of judgement and choice and the assumptions and predictions of economics[5]. Richard Thaler was the first economists to recognize the potential applications of this research to economics. His 1980 article "Toward a theory of consumer choice," published in the first issue of the remarkably open-minded (for its time) Journal of Economic Behavior and Organization, is considered by many to be the first genuine article in modern behavioral economics."
> Camerer et al. (2004), p. xxi-xxii

Contrary to these claims, the real origins of modern behavioural economics can be traced back to Ward Edwards, particularly to Edwards (1954) and Edwards (1961), which provided the methodological framework for modern behavioural economics. Edwards, in turn, draws inspiration from the famous subjective probability theorist and statistician Leonard Savage. The two papers summarize the emergence of core notions that characterize what may, with hindsight, be called a *Neoclassical Theory of Behavioural Economics* and offer detailed philosophical and methodological discussions related to them. More importantly, Edwards posed challenges to orthodox neoclassical notions, focusing on psychological and experimental foundations, constraints and predictions. He also introduces and provides a remarkable and detailed survey of the classic works in the field of behavioural economics till then[6].

The most remarkable aspect of Edwards' paper is the formalization of weighted values and the introduction of *Subjective Expected Utility* (Ramsey, 1926; Savage, 1954). He also sheds light on the early studies on subjective probability before and after Savage's book in 1954. The standard formulation on the objective function faced by a decision maker in an economic model under risk/uncertainty is presented as a *linear combination* of the values of outcomes and probabilities attached to each of these outcomes. The values of outcomes and probabilities, both, can be *objective* or *subjective*.

[5] As if these were not prime motivations for Simon when he launched his programme of research on behavioural economics. It is just that Simon's psychological and cognitive bases for modelling *realistic* economic behaviour were always underpinned by a *model of computation*.

[6] So the claim that behavioural economics was not even a field till 1980 is highly questionable, even from the works by the precursors of Kahneman, Tversky and Thaler.

The formulation of expected utility can be stated as:

$$E(U) = \sum_{i=1}^{n} p_i \cdot U_i \ ,$$

where p_i is the probability of the i_{th} outcome of n possible ones and U_i is the value of the i_{th} outcome. Based on this we can have the following classification: when subjective values are weighted with objective probabilities, it results in Expected Utility. Instead, when subjective values are weighted with subjective probabilities, it becomes *Subjective Expected Utility*. The other two alternatives were considered to be unimportant or proved to be unrealistic in the literature.

The classic Expected Utility formulation was first devised by von Neumann and Morgenstern (vN-M), who explicitly invoked formal, 'objective' probability theory and were even prepared to use the frequency theory of probability[7] - explicitly and forcefully rejected by Savage, whose work was deeply influenced by de Finetti's foundational work on subjective probability theory. Thus, the probability with which they axiomatized expected utility maximization is actually objective. Since then, it became clear that Expected Utility fails to explain and predict individual behaviour under risk - let alone uncertainty (a distinction not carefully maintained by the practitioners of MBE). vN-M attempted to make the qualitative notion of utility and preference measurable just like, say force in physics. The main argument was that, for economics to be a rigorous science, formalised mathematically, preferences should be *measurable*[8]. Furthermore, for preferences to be measurable, they should be numerically definable and mutually comparable. Individuals are supposed to seek and be able to choose the outcome which will give them the highest satisfaction among all the possibilities. But neither the process that underpinned 'seeking', nor the process of 'choosing' were given any *procedural* content, unlike the way Simon, who from the outset sought to emphasise the search processes at the foundations of choice over a complex space of alternatives.

There was a great deal of effort that was dedicated to measuring utilities and probabilities under the framework of subjective (personal) probability around the time of the early work of Ward Edwards. This empirical work went hand-in-hand with the simultaneous formalization by Savage, who built his *foundations of statistics* on the basis of de Finetti's theory of subjective probability. In Savage's scheme, the assumptions of complete preference ordering and the *sure-thing* principle play a crucial role, and the individuals learn and adjust their prior beliefs with the occurrence of events according to Bayes's theorem. These properties for subjective probabilities proposed by Savage, in turn, implies that individuals with different sets of subjective probabilities, over the course of their experience, will end up having subjective probabilities which coincide with each other.

[7]But not in its modern refounding and reformulation as *algorithmic probability*

[8]How to measure variables over the *reals* was never specified except by vague references to varieties of approximations.

The critical point of rapid development of MBE can be attributed to the proposal of *Prospect Theory* by Kahneman and Tversky (in Kahneman and Tversky, 1979), which was considered as a satisfactory replacement of expected utility theory. The theory encapsulates the idea of subjective probability[9] (not directly) and loss aversion. Even today, loss aversion remains one of the most notable behavioural postulates used to interpret and model decision making in different contexts.

A series of "anomalies" - resulting from the violation of transitivity and other axioms, inconsistency with some principles of neoclassical economics - have been systematically collected and investigated by contemporary behavioural economists, notably, Richard Thaler, Colin Camerer, George Lowenstein, Matthew Rabin among many others, since the late 1980s in the *Journal of Economics Perspectives* (for example). The inconsistency in behaviour is mainly observed in experimental environments, and thus the neglect of psychological and social factors are proposed as possible causes for this, according to MBE. The Neoclassical agents are now like physically weakened patients unable to predict even reasonably well, who are being examined with the benchmark idealized case of orthodox theory and its strict rational, constrained optimization, behaviour and the modern behavioural economists are assuming the role of seeking and proposing the remedies for them. The themes and fields challenged from which *anomalies*[10] are found cover Microeconomics, Macroeconomics, Finance Theory, Industrial Organization, Game Theory and even Development Economics. This has led to an encompassing field of behavioural economics, broadly divided into (at least) Behavioural micro, Behavioural macro, Behavioural finance and Behavioural game theory.

Sub-Fields of Modern Behavioural Economics

Behavioural Microeconomics Some anomalies concerning preference and utility in decision making are studied. Preference reversal is believed to be a robust anomaly. This field of research attempts to challenge the commonly agreed notions in neoclassical theory - that the values of goods or outcomes do exist and people know these values directly - by highlighting the presence of *framing effects*, reference based effects, etc (Tversky and Thaler, 1990). It is also suggested that the assumption of a stable preference ordering should be discarded. The preference changes can be due to a variety of factors such as *status quo bias*, *loss aversion*, *ambiguity aversion* and *endowment effects*. Their thesis is that a consideration of these factors can make the analysis of preference

[9] But neither consistently, nor meaningfully. In the whole literature on MBE, all the way from the early works of Kahneman and Tversky, there is a remarkable confusion and conflation of a variety of theories of probability, even within one and the same framework of modelling rational, psychologically underpinned, individual behaviour in economic contexts. See in particular, Tversky (1972)

[10] Velupillai refers to this trait in MBE as *anomaly mongering* in his lectures on Behavioural Economics. His point is that both the Newclassicals, whose analogous notion is 'puzzle' 'equity premium puzzle' being paradigmatic and the Modern Behavioural Economists are consciously invoking Kuhn's terminology and, therefore, suggesting that their programme of research is leading to that much maligned concept of a 'paradigm shift'.

more manageable and *tractable*[11]. The general worry is that importing psychological inspiration into existing economic models may create new complexities and reduce their predictive power (Kahneman et al., 1991). Furthermore, the difficulty and infeasibility of utility maximization was pointed out, and economists sought for possible psychological and social causes as explanation for the "mistakes"[12] in decision making (Kahneman and Thaler, 2006).

In MBE, the majority of research seems to focus on suggesting so-called more 'realistic' utility functions in the context of modelling decisions. A whole taxonomy of varieties of MBEs could be catalogued on the basis of the criterion of 'realistic utility functions', but this will be a detour from our more basic aims. There is also a rich menu of research questioning the fundamental framework of preference maximization and the modelling of satisficing, even within one or another variety of MBE, where, in general, for formalizing the notion satisficing in a pseudo-procedural context, *heuristic searches*[13] are applied. Heuristics serve as guides helping decision makers to find short cuts for relevant information. Together with satisficing, decision makers are supposed to *stop searching* - i.e., an exogenously determined stopping rule for the search process is activated - whenever some (exogenously determined) criteria are reached (e.g. aspiration level). However, they are not necessarily aware of computability or algorithmic undecidability which is inherent in many such procedures. If the heuristic search is programmed as a finite automaton, it will naturally terminate at some point. However, if it is programmed as a Turing Machine, then the decision maker is confronted with the famous result of the *halting problem* for Turing Machine. This means that the agent who is searching is either not able to determine whether the heuristic reached the exogenously determined aspiration level, or - even worse - whether it will ever do so within any reasonable, or even unreasonable, exogenously given time span.

Behavioural Macroeconomics Similar psychological and social reasons are also applied to interpret some Macroeconomic phenomena[14], such as, money illusion, rigidity of (nominal) wages (loss aversion and fairness) and involuntary unemployment (gift-changing equilibrium of reciprocal preference). The most far-reaching challenge might be to address the questionable idea of the traditional notion of *Discounted Utility*. The presence of non-exponential discounting of utility was observed (Loewenstein and Thaler, 1989), and subsequently[15] other ways (hyperbolic, quasi-hyperbolic discounting etc.) of discounting were devised, which are believed to be more realistic and better able

[11]Not formalised in terms of tractability in the formal hierarchy of degrees of computational complexity simply because these models are not underpinned by any formal model of computation.

[12]In other words, 'Anomalies'!

[13]Without, however, any recognition that 'heuristics' are, formally, 'algorithms'.

[14]Akerlof and Shiller (Akerlof and Shiller, 2009) categorise five types of animal spirits: they are confidence, fairness, corruption and antisocial behaviour, money illusion and stories.

[15]Not quite 'subsequently', because the notion of hyperbolic discounting has been 'around' in intertemporal macroeconomic policy models at least since the early 1960s. But it is to the credit of the MBE's practice and insistence that traditional and almost routine recourse to exponential discounting in intertemporal optimisation models is being challenged.

to provide predictive models in the context of intertemporal choices. Other than time discounting, there is also research on behavioural life-cycle theories (e.g. mental accounting Thaler (1990)) on saving and marginal propensity to consume and on *regret theory*, such as using counterfactual, introspective thinking and self control of future misbehavior on consumption and saving. But in no such case have *non-traditional logics* been utilized to derive counterfactual predictions based on introspective thinking.

Behavioural Finance Behavioural finance appears to provide an alternative to the *Efficient Market Hypothesis* and it is probably one of the most developed subfields in modern behavioural economics. In other words, it is commonly believed that the efficient market hypothesis has virtually died out. The well known anomalies in finance include the *equity premium puzzle* (high risk aversion), *calendar effects*, *status quo effect*, *limits to arbitrage*, social preference and other stylized facts (de Bondt and Thaler, 1989; Froot and Thaler, 1990; Lamont and Thaler, 2003; Lee et al., 1990; Siegel and Thaler, 1997; Thaler, 1987b,a). Due to the nature and functioning of financial markets, huge amount of data points, at high frequencies are available. Therefore it is also a rich ground for behavioural and (so called) computational economists to investigate and validate their models.

Behavioural Game Theory Similar to the other fields, behavioural game theory investigates how the results regarding *strategic interaction* deviate from the orthodox game theoretic predictions in the light of some behavioural assumptions regarding decision making in strategic situations. The psychological and social explanations such as *guilt aversion* and *fairness criteria* are incorporated into the traditional models. Behavioural game theory benefits from the fact that most of these models can be tested in laboratory environments by collecting a sufficient number of subjects. Therefore, it coexists with experimental economics and neuroeconomics. A reasonably up to date survey of behavioural game theory can be found in Camerer (2003).

Concluding Remarks

Although neoclassical economic theories have been critically questioned by economists and psychologists for many decades, it is still explicitly specified that *optimization*, *equilibrium* and *efficiency*, and on which Neoclassical economics - and its variants, such as Newclassical and New Keynesian - theories are based, are not completely rejected by behavioural economists (see, for example, the opening, programmatic, pages of Camerer et al. (2004)). The ultimate goal of behavioural economists seems to be to extend, not replace, neoclassical theories in a normative sense.

Modern behavioural economists have, over the years, discovered and categorized different forms of deviations from consistent behaviour. A valid question here would be: why do these anomalies arise and what are the anomalies with respect to? Discoveries such as reference dependence, loss aversion, preference over risky and *uncertain* outcomes and time discounting, came mostly from observations in experimental environments. The anomalies and puzzles that were discovered and discussed are departures with respect to the neoclassical normative benchmark for judging rational behaviour, which is (subjective) expected utility maximization. These evidences or anomalies are in turn used to formulate more realistic utility functions and further, these modified utility functions are incorporated into the existing models. In some sense, Modern behavioural economists modified fractured pieces in the foundations of Neoclassical theories, but still they worked within its basic premises (preferences, utility, equilibrium and maximization).

Thus, MBE preserves the doctrine of utility maximization and does not go beyond it or discard it. Though the behavioural models do consider more realistic psychological or social effects, economic agents are still assumed to be optimizing agents whatever the objective functions may be. In other words, MBE is, still within the ambit of the neoclassical theories or it is in some sense only an extension of traditional theory by replacing and repairing the aspects which proved to be contradictory. These adjustments in turn are expected to enhance the predictive power of the original theories. On the contrary, CBE does not try to endow the economic agent with a preference order which can be represented by utility functions; nor, of course, do equilibria or optimization play any role in the activation of behavioural decision making by CBE agents.

2.2.2 Classical Behavioural Economics

It is interesting to note that even before the advent of behavioural economics, economic debates on decision making were richly based on behavioural and epistemological principles and cognitive psychology. Keynes, of course, is the paradigmatic example of one who explicitly considered psychological factors in his macroeconomic theories. Simon's bounded rationality and his view on decision making can be traced back to his precursors as early as Aristotle's *practical rationality*. Practical rationality is a cognitive state and capacity of identifying the right means or the prescription of actions for achieving desired state of life or goods (Miller, 1984). The emphasis is on the actions that need to be undertaken and the deliberation required for recognizing these actions, rather than the outcomes, let alone the 'best' outcome.

One of the most essential and concrete line separating MBE and CBE is that rational behaviour is adaptive or *procedural* in CBE; this makes rational behaviour naturally algorithmic and the need to underpin it with a model of computation enters right on the ground floor of theory and its empirical counterparts. Given the nature of adaptive behaviour and the complex environment in which it takes place, optimization principles and equilibrium analysis become meaningless and nearly infeasible. The resolving of these difficulties should not be to find approximations of sophisticated mathematical models using numerical techniques, like what we see in some parts of MBE.

As far as *dynamical rational behaviour* is concerned, where *procedure* is central, Simon, Richard Day, Richard Nelson and Sidney Winter are considered as the pillars of CBE (Velupillai, 2010b). In particular, Nelson and Winter's pioneering contributions to evolutionary economics is developed by viewing technological changes or innovations in a non-conventional way. Innovations are consequences of organizational decision making, where maximizing strategies have little role to play. In Nelson and Winter (1977, 2002), uncertainty, complexity and heuristics are highlighted and they play an important role in shaping the theories of innovation or technological changes. This way of viewing decision making is consistent with that of Simon's approach described earlier.

The research line of the thesis was motivated by the questions 'How does the mind work?'; 'What kind of Mechanisms should we postulate for the Mind, based on current knowledge and research in Cognitive Science, to make sense of observed behaviour'; 'What postulates are useful to understand and predict behaviour?'; 'What metaphors are useful to formalize intelligent procedural behaviour?'; 'How do operational institutions emerge and survive'? Research surrounding these questions is intrinsically underpinned by cognitive psychology and the theory of computation. They lead also to what became the natural Simon framework of *Human Problem Solving*, of agents faced with complex and intractable search spaces, constrained by computationally underpinned cognitive processes facing time and resource constraints. A notable precursor for Simon, on these aspects was Polya.

Simon is best known by the felicitous phrase he coined, "bounded rationality", which appeared in Simon (1957) for the very first time (although it had appeared in other forms already from his classic book Simon (1947)). Bounded rationality generally refers to the internal cognitive limitations, and the constraints of the external environment which confront human minds in decision making contexts. This latter is more specifically contextualized by the institutional backdrop for individual behaviour. Therefore, in order to incorporate the notion of bounded rationality into the behavioural model more rigorously, one ought to investigate how human thinking is limited internally and how human beings adapt and interact with the environment, especially as members of an institution.

Simon's insight about modelling adaptive individuals in complex economic environments can be better understood in the fragment:

> "Suppose we are pouring some viscous liquid molasses into a bowl of very irregular shape.... How much would we have to know about the properties of molasses to predict its behavior under the circumstances? If the bowl were held motionless, and if we wanted only to predict behavior in equilibrium, we would have to know little, indeed, about molasses. The single essential assumption would be that the molasses, under the force of gravity, would minimize the height of its center of gravity. With this assumption, which would apply as well to any other liquid, and a complete knowledge of the environment, in this case the shape of the bowl, the equilibrium is completely determined. Just so, the equilibrium behavior of a perfectly adapting organism depends only on its goal and its environment; it is otherwise completely independent of the internal properties of the organism.

> If the bowl into which we are pouring the molasses were jiggled rapidly, or if we wanted to know about the behavior before equilibrium was reached, prediction would require much more information. It would require, in particular, more information about the properties of molasses: its viscosity, the rapidity with which it "adapted" itself to the containing vessel and moved towards its "goal" of lowering its center of gravity. Likewise, to predict the short run behavior of an adaptive organism, or its behavior in a complex and rapidly changing environment, it is not enough to know its goals. We must know also a great deal about its internal structure and particularly its mechanisms of adaptation."
>
> Simon (1959), p. 255

Simon criticized orthodox normative economics for ignoring how human beings actually behave and questioned the result that only rational agents survive the forces of competition - with orthodoxy's Olympian assumptions (Simon, 1983b) on how to formalize rational behaviour, which was - at least as far as Simon was concerned - remote from any cognitive realism. Besides, the study of equilibrium requires little understanding of the characteristics of individuals in *out-of-equilibrium* situations, simply because normative economics has nothing to say about *process* and *procedure*. In the real world Simon saw around him, there exists a great deal of turbulence, not only generated by external shocks, that keeps the system *out of equilibrium* and agents needing to relocate their bearings almost ceaselessly.

Furthermore, Simon stated "decision making under uncertainty" instead of "decision making under risk" in Simon (1959). That is, an economic agent might respond to the changing environment in a personal way rather than knowing the objective probability of what outcomes might occur in the future. This property brings more difficulties on the prediction of rational individual behaviour by using so-called objective characteristics of the environment.

Simon's behavioural economics is almost comprehensively demonstrated by his encapsulation of *Human Problem Solving* and agents and institutions as *Information Processing Systems*. Although the problems which Simon dealt with are well structured problems, such as Chess playing, the combinatorial complexity of the problem is massive enough to prevent human players using *minimax* strategies which are suggested in traditional game theory. In this chapter, only a few of Simon's massive and wide ranging contributions are tackled.

2.3 Underpinnings of Behavioural Economics

In this section, different underpinnings and analytical tools of MBE and CBE will be briefly mentioned. The purpose of this section is not to provide detailed theoretical and technical details, but to try to make clear distinctions on how the two lines differ fundamentally. It is slightly puzzling that this distinction has never been made earlier. As one may realize from the following underpinnings and the sub-branches of MBE introduced in previous sections, MBE can be characterized as a massive magnet which attracts different resources, new tools and ways of explanations. We can almost claim that MBE has already become a new mainstream economics, as a consequence of it playing the role of a revised approach of orthodox economics rather than an alternative approach. On the other hand, CBE is developed on completely different grounds from MBE. From our point of view, MBE is fostered by Orthodox Economic Theory, Game Theory, Mathematical Finance Theory and Recursive[16] Methods, Experimental Economics and Neuroeconomics, Computational Economics[17] and Subjective Probability Theory.

CBE, in our reconstruction of it, on the other hand, is based fundamentally on a model of computation - hence, Computable Economics - computational complexity theory, nonlinear dynamics and algorithmic probability theory.

2.3.1 Underpinnings of Modern Behavioural Economics

Orthodox Economic Theory

It is in human nature to aspire to predict, at least so the sages say, and the traditional wisdom of many cultures concur. Microeconomics, in general, is the study of individual choices and actions. Gradually, economics has developed normative axioms and theories on how the individual entities (including organizations) *should* make choices and how they seem to make choices. There are, classically (but not necessarily exhaustively) the normative and positive approaches to behaviour, respectively. In Neoclassical theory, economic agents are assumed to be fully rational and completely[18] informed. It is not that they know everything, but that they *can know* everything and there are means to learn - epistemology - and they know how to make the best choices for themselves (even if only probabilistically). Second, in order for their choices to be tractable, axioms (completeness, reflexiveness, transitivity, and continuity) of rational preference were devised, within classical mathematical formalisms - which simply means the mathematics of (Zermelo-Fraenkel) set theory plus some variant of the axiom of choice. Individuals are assumed to have underlying preference orderings for all the alternatives which are knowable, although the *means of getting to know them* is never specified. These rational preferences are, often, represented by a utility *function*, which is assumed to be *well-behaved*. Third, the non-satiation assumption promises that the satiation point will never be reached, at least in the economic domain. Thus, the individuals are always in the state of the world where “more is better”.

In passing, it could be mentioned that there have been serious and contentious discussions in the history of the development of economic theory as to whether utility should be cardinal or ordinal, since there might consequently result in differences in the way in which economists try to measure utility. Eventually, ordinal utility seems

[16]Not Recursion Theory
[17]Not Computable Economics
[18]Often this ‘completeness’ is probabilistic of a naïve variety.

to have reached dominance, although not very 'consistently'; subsequently the theory of individual decision making based on preference and choice-based approaches were developed.

In passing, in lieu of discussing game theory and its place in MBE, we would like to make two points: Firstly, we disagree that game theory, even in its strategic form, originates with either von Neumann-Morgenstern or with von Neumann's 1928 paper. Our alternative history is outlined in several of Velupillai's recent papers on computable economics. Secondly, it may be pertinent to add that no game theoretically defined Nash equilibrium is computable and no algorithm which has been claimed to determine it can be implemented without appealing to undecidable disjunctions.

Mathematical Finance Theory and Recursive Methods

A huge amount of mathematical theories and tools have been borrowed to develop finance theories and time series analysis. In these exercises, different stochastic or random processes are imported to represent the data generating process of finance or economic time series, e.g. Brownian motion and Markov chains. The random processes applied here are based on measure theoretic concepts.

Recursive methods in macroeconomic are built on dynamic programming, Markov decision processes and Kalman filtering and again, measure theory, underpinning orthodox theories of stochastic processes and probability, plays a central role - all within one or another form of nonconstructive and non-recursion theoretic real analysis (for example, for dynamic programming the notion of one or another form of contraction mapping in a suitable metric space).

Although the mathematical tools used are much more sophisticated than in non-dynamic methods - but only up to a point, economic entities are still modelled as optimizers (e.g. maximizing present values in intertemporal contexts, *Value functions* and *Euler equations* in the context of dynamic programming and optimal control formulations) where it is little realised that the analysis is around uncomputable equilibria (cf. Ljungqvist and Sargent, 2004; Stokey and Lucas, 1989).

Experimental Economics and Neuroeconomics

Experimental Economics appears as a tool for examining economic theories in computational, numerical and other obviously implementable ways in which idealized subjects are placed in artificial settings that purport to mimic the theoretical environment. Narrowly speaking, it is not categorized as a branch in economics, instead, it is a methodology for researchers to support or refute specific economic theories. While, broadly, it can be considered to be cohabiting with behavioural economics. This is because - or claimed to be because - what people actually do can be observed in experimental environments, and almost all the anomalies are found and induced from laboratory environments or field studies. The methodology of experimental economics is heavily based on so-called *induced value theory* (Smith, 1976). Induced value theory suggests that in the controlled laboratory environment, if subjects are suitably motivated, experimenters can expect to obtain desirable induced values from choices of subjects on certain economic problems they are given to 'solve'[19]. This theory is obtained from non-satiation assumptions, and monetary payment is the most commonly used reward for inducing real values from subjects. However, if economic agents are actually applying *satisficing* principles to the experiments they attend, i.e. they are satisfied by performing *decently* rather than trying their best or thinking hard in order to get the most reward, then the results of experimental economic could be very misleading.

Neuroeconomics is the new extension of experimental economics incorporating neuroscience to obtain the data of brain activity, simultaneously, when the subject is in a laboratory environment. It is also viewed as a young subfield of behavioural economics which is believed will be the main focus in the future. A popular claim is the dual system in our brain supervising our judgemental and intuitive thinking, corresponding to rational and emotional behaviours. It provides the technique to collect data in the brain for examining how and when the behaviour of decision makers could deviate from rational and consistent behaviour. Recent surveys can be found in Camerer (2007), Glimcher et al. (2005), and Rustichini (2005); a critical view of the claims of Neuroeconomics can be found in Rubinstein (2008).

Computational Economics

Computational Economics is also an extension of experimental economics from another perspective, i.e. the subjects are not human subjects but *software* subjects. So far, there are at least two well-developed lines, which are heterogeneous agent models and agent-based modelling, and the survey for these respective lines can be found in Hommes (2006) and LeBaron (2006). A thorough critique of the excessive claims of both these lines - and of other strands of computational economics - is given in Velupillai and Zambelli (2011).

Heterogeneous agent models seem to have been inspired by related results on cellular automata modelling in the physical sciences, resulting in unpredictable and complex phenomena generated by simple interaction rules. The claims in this line of research are as vacuous as those made by agent-based modellers in finance and economics. They both suffer from a serious lack of scholarship and complete unhinging of their foundations in either serious computability theory or even a familiarity with the fruitful and frontier research in the interface between dynamical system theory, numerical analysis and computability. These interactions were the fulcrum around which von Neumann and Ulam, Conway and Wolfram and Turing (1952) pioneered their studies of emergent complex dynamics in interacting systems with simple rules of interaction.

[19] But this is not the search for 'solutions' in any kind of 'problem solving' context, as in CBE.

Subjective Probability Theory

Subjective expected utility theory was proposed by Savage in 1954, between the period in which Edwards wrote his first and second survey papers on behavioural economics (Edwards, 1954, 1961). Savage followed the axiomatizations along the lines proposed by Ramsey (Ramsey, 1926) and de Finetti (de Finetti, 1964), and applied Bayes' rule for updating the prior probabilities over time.

The idea of subjective expected utility 'first' appeared in Modern Behavioural Economics[20] through the work of Kahneman and Tversky (1979), when building a descriptive theory of decision making by individuals under risk. Their theory in turn borrowed heavily from Edwards (1962), who in turn built on Savage (1954). Both Edwards and later Kahneman and Tversky, however do not refer to Bruno de Finetti whose contributions are not mentioned in these two papers. There seems to be some ambiguity while they talk about probabilities in their model and this gets particularly unclear when they refer to decision weights[21].

> "In prospect theory, the value of each outcome is multiplied by a decision weight. Decision weights are inferred from choices between prospects much as subjective probabilities are inferred from preferences in the Ramsey-Savage approach. However, decision weights are not probabilities: they do not obey the probability axioms and they should not be interpreted as measures of degree of belief."
>
> Kahneman and Tversky (1979), p. 280

In this framework, decision weight measures over stated probabilities do not obey the property of *additivity*[22]. In the Savage-de Finetti framework, the sum of the probabilities over exclusive and exhaustive events adds up to unity. In prospect theory, the sum of decision weights is considered less than one in most of the cases. However, while they invoke the Ramsey's approach of inferring these decision weights from choices, it naturally raises the question as to what these decision weights are? Although the propositions of decision weights are derived in the paper, it is unclear how they are different from *degrees of belief* - although different they must be!

In Edwards (1962), two categories of subjective probability models are introduced: *additive* and *non-additive ones*[23]. In Edwards' elaboration, first of all, subjective

[20] Ignoring, for the moment, the much earlier work of Edwards, who was more than a mentor to Kahneman and Tversky.

[21] Decision weights, in Kahneman and Tversky (1979), are the measures associated with each probability, reflecting the impact of probability of the overall value of the prospect.

[22] Additivity of probability is defined as follows: If n numbers of events form a complete set of incompatible evens (meaning exactly one of the events has to be true), then the probability of the logical sum (the logical sum of a group of events is true, if and only if, one of the events is true) is equal to the sum of their respective probabilities. Since n is a finite natural number, the definition above is more precisely *finite additivity*. On the other hand, when n approaches infinity, it becomes *countable additivity*.

[23] The distinction of additive and non-additive probability made in Edwards (1962) is that additive probabilities sum up to specific (real) numbers, non-additive ones are not supposed to do so then, what are they, if they do not do so?

probability is a number raging from zero to one and describing a person's assessment of the likeliness of an event. Further, it is assumed objective probabilities exist, and they are related to subjective probabilities. Edwards argues that its is meaningless to debate whether objective probabilities can be defined, in contrast to de Finetti's and Savage's firm belief that there are no objective probabilities. He goes on to make a distinction between risk and uncertainty. He argues that there are some cases, such as die tosses, which have "conventional" probabilities over their outcomes. Consequently, there events which can be given objective probabilities, are defined as risky; otherwise, they are uncertain. However, both Edwards (1962) and Kahneman and Tversky (1979) considered only risky cases. Tversky and Kahneman (1992) is a revision of prospect theory including (allegedly) uncertain outcomes.

The concept of subjective probability is used ambiguously - to put it mildly - in MBE. One the other hand, MBE introduced the idea of personal probability, defining it and mapping it over the objective probabilities in risky choices. *This is quite different from the kind of subjective probabilities proposed by de Finetti, and does not necessarily follow his axioms of subjective probabilities.*

Subjective probabilities of outcomes, for de Finetti, are the different degrees of belief regarding the occurrences of events that people possess. These degrees of belief, however, need not be the same for all the people. In an attempt to find admissible ways of assigning numbers to different degrees of belief, de Finetti constructed axioms over events and their probabilities, especially, through the logical relations of events. By standardizing a *random quantity* into 1 and 0 representing the truth and falsity of an event and by introducing the *coherence criterion*, de Finetti derives some basic consequences. The most important one amongst them is the concept of *finite additivity*, where the sum of assignments over finite events (logical sums) adds to unity. More specifically, for de Finetti, the qualitative criteria regarding coherence appears first, and then, the individuals are allowed to freely attach numbers to their degrees of belief over a complete set of *incompatible events* (i.e., exhaustive and exclusive events), however, within the coherence constraint. This way, a qualitative idea of coherence is linked to the mathematical expressions of (subjective) probability. The coherence principle demands consistency in assignments, based on the idea that no arbitrary gains should be available for either player by accepting certain books of bets (the *Dutch* Book argument). In order to satisfy the coherence principle, the sum of probabilities of the event has to be unity (the necessary and sufficient condition of coherence). Besides, it should be noted that Bayes' conditional probability formula is derived in turn form coherence, and it is not taken as a definition in de Finetti's theory of probability.

Before de Finetti, Frank Ramsey gave a talk in 1926 and the lecture was published in Ramsey (1926), of which de Finetti was not aware until 1937. Both of them, almost simultaneously but independently, formulated subjective probability as a *degree of belief* held by an individual and devoted their efforts to axiomatize it. In particular, de Finetti assumed and insisted only the use of finite additivity, because *the requirement of coherence implies finite additivity*. On the other hand, Ramsey simply and intuitively addressed this issue saying that is meaningless to discuss infinite events, because he doubted a human being's capability of handling infinite events.

> "[N]othing has been said about degrees of belief when the number of alternative is infinite. About this I have nothing useful to say, except that I doubt if the mind is capable of contemplating more than a finite number of alternatives. It can consider questions to which an infinite number of answers are possible, but in order to consider the answers it must lump them into a finite number of groups."
> Ramsey (1926), p. 183[24]

In contrast to finite additivity, frequentists and measure theorists advocate and justify the use of countable additivity (or denumerable additivity, infinite additivity and σ-additivity) by invoking the strong law of large numbers (Borel) and relative frequency in limits. Howson (2009) discusses these issues in detail and supports de Finetti's idea of finite additivity, but not, in our opinion, in a convincing way.

In particular, we fundamentally disagree with Howson that 'de Finetti himself would have recommended' doing 'probabilistic reasoning ... in an informal metatheory consisting the *usual mathematics of analysis and set theory*' so that:

> "Deductive consistency and probabilistic consistency are ... subspecies of the same fundamental notion of the solvability of equations subject to constraints: those of a classical truth-valuation in the deductive case, and the rules of finitely additive formal probability in the probabilistic case"
> Howson, op. cit, p. 55-6; italic added.

This is a fundamental violation of every tenet of epistemology and methodology advocated by de Finetti. Moreover, Howson does not seem to realize that it is provably hard to devise procedures to validate 'classical truth-valuation'.

Edwards (1962, p.117), inexplicably, considers the infinite case to be more interesting as compared to the finite case . More recently, Bayesian approaches, together with Savage's notion of subjective probability, are challenged by empirical evidences that suggest agents are incapable of applying the Bayesian rule to revise their prior probabilities. Case based theory which is considered as one of the new foundations for behavioural decision theory, bases the probabilities assigned to different events on previous histories regarding similar cases and consequently, adopts a (non-algorithmic) frequentist approach for the probabilities (cf. Thaler, 2005; Camerer et al., 2004).

2.3.2 Underpinnings of Classical Behavioural Economics

> "If we hurry, we can catch up to Turing on the path he pointed out to us so many years ago."
> Simon (1996a), p. 101

[24]It may be apposite to point out that Ramsey's equally distinguished fellow-Kingsman, a few years later, in his classic on computability theory, appealed to the same kind of 'finiteness' for the same kind of reason (cf. Turing, 1936, p.249)

Classical Behavioural Economics was underpinned, always and at any and every level of theoretical and applied analysis, by a *model of computation*. Invariably, although not always explicitly, it was *Turing's model of computation*.

The fundamental focus in classical behavioural economics is on decision problems faced by *human problem solvers*, the latter viewed as *information processing systems*, as we emphasise in this thesis. All of these terms are given computational content, *ab initio*. But given the scope of this chapter we shall not have the possibility of a full characterisation. The ensuing 'birds eye' view must suffice for now.

Firstly, a decision problem asks whether there exist an algorithm to decide whether a mathematical assertion does or does not have a proof; or a formal problem does or does not have an algorithmic solution. Thus, the characterisation makes clear the crucial role of an underpinning model of computation; secondly, the answer is in the form of a yes/no response. Of course, there is the third alternative of 'undecidable', too. It is in this sense of decision problems that we interpret the word 'decisions' here.

As for 'problem solving', we shall assume that this is to be interpreted in the sense in which it is defined and used in the monumental classic by Newell and Simon (1972), which is, in our opinion, an application of the theory underlying Turing (1954).

Finally, the model of computation is the Turing model, subject to the Church-Turing These. To give a rigorous mathematical foundation for bounded rationality and satisficing, as decision problems[25], it is necessary to underpin them in a dynamic model of choice in a computable framework. However, these are not two separate problems. Any formalization underpinned by a model of computation in the sense of computability theory is, dually, intrinsically dynamic. Moreover, *Decidable-Undecidable*, *Solvable-Unsolvable*, *Computable-Uncomputable*, etc., are concepts that are given content *algorithmically*, within a model of computation.

Definition 1. *A Boolean formula consisting of many clauses connected by conjunctions (i.e., $\wedge$) is said to be in Conjunctive Normal Form (CNF).*

Now consider the Boolean formula:

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_1) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

Each subformula within parenthesis is called a clause; The variables and their negations that constitute each clause are called literals; It is 'easy' to 'see' that for the truth value of the above Boolean formula to be $t(x_i) = 1$, all the subformulas within each of the parenthesis will have to be true. It is equally 'easy' to see that no truth

[25]The three most important classes of decision problems that almost characterise the subject of computational complexity theory, underpinned by a model of computation in general, the model of computation in this context is the *Nondeterministic Turing Machine* are the *P*, *NP* and *NP-Complete* classes. Concisely, but not quite precisely, they can be described as follows: 1. P defines the class of computable problems that are solvable in time bounded by *a polynomial function* of the size of the input; 2. NP is the class of computable problems for which a solution can be *verified in polynomial time*; 3. A computable problem lies in the class called *NP-Complete* if every problem that is in NP can be *reduced to it in polynomial time*.

assignments whatsoever can *satisfy* the formula such that its global value is true. This Boolean formula is *unsatisfiable*. This is the kind of 'satisfiability' we ascribe to Simon's notion of 'satisficing'.

Definition 2. ***SAT - The Satisfiability Problem***
Given m clauses, $C_i (i = 1, \ldots, m)$, containing the literals (of) $x_j (j = 1, \ldots, n)$, determine if the formula $C_1 \vee C_2 \vee \ldots \vee C_m$ is satisfiable.

Determine means 'find an (efficient) algorithm'. To date it is not known whether there is an efficient algorithm to solve the satisfiability problem - i.e., to determine the truth value of a Boolean formula. In other words, it is *not known* whether SAT∈P. But:

Theorem 1. *SAT∈NP*

Finally, we have Cook's famous theorem (Cook, 1971):

Theorem 2. ***Cook's theorem***
SAT is NP-Complete

It is in the above kind of context and framework within which we are interpreting Simon's vision of behavioural economics. In this framework, optimization is a very special case of the more general decision problem approach. The real mathematical content of *satisficing*[26] is best interpreted in terms of the satisfiability problem of computational complexity theory, the framework used by Simon consistently and persistently - and a framework to which he himself made pioneering contributions.

We have only scratched a tiny part of the surface of the vast canvass on which Simon sketched his vision of a computably underpinned behavioural economics. Nothing in Simon's behavioural economics - i.e., in Classical Behavioural Economics - was devoid of computable content. There was - is - never any epistemological deficit in any computational sense in classical behavioural economics (unlike in Modern Behavioural Economics, which is copiously endowed with epistemological deficits, from the ground up).

2.4 CBE - Notes on the Special Role of Herbert Simon

A basic tenet of Simon's approach to behavioural economics is that the limitations of cognitive processing should be linked, in some formal way, with the definable limitations of computation, subject to the Church-Turing Thesis (without say space or time constraints). The limits of computational complexity, on the other hand, are naturally

[26]In Simon (1997), p. 295, Simon clarified the semantic sense of the word satisfice:

> "The term 'satisfice', which appears in the Oxford English Dictionary as a Northumbrian synonym for 'satisfy', was borrowed for this new use by H. A. Simon (1956) in 'Rational Choice and Structure of the Environment' "

.

bounded by the time and space. Behavioural models, in which agents are supposed to exercise rational behaviour, whether psychologically more realistically constrained or not, hypothesizing capabilities transcending these theoretical and practical limitations are, for Simon, empirically meaningless. Simon has taken the limits of human cognition into account, transformed into computational complexity measure, for describing agents who make decisions. This is why we are convinced that computable foundations and nonlinear dynamics can be found in *Information Processing Systems*[27], the paradigmatic formalization of agents and institutions in the kind of behavioural economics Simon advocated.

2.4.1 Bounded Rationality

The *idea* of bounded rationality was first proposed by Herbert Simon in the paper titled "*A Behavioural Model of Rational Choice*", which was published in 1953. It was further polished and republished with a same title as the much more famous Simon (1955) and was initially phrased as "limited rationality". Less than a decade later, the idea was more specifically termed as "procedural rationality." It reveals Simon's explicit intention of introducing procedural contents of decision making into the notion of rationality. Though the idea of bounded rationality has been discussed using different names by Simon, there is no inconsistency among them. In Simon (1955), an example was constructed where agents tend to be satisfied by using certain information they have and avoid information they do not really have any means of obtaining in algorithmically meaningful ways. They anticipate something acceptable in the near future without calculating any probabilities or assigning probabilities to prospective future events. Simon further described human behaviour as "intendedly rational" in Simon (1957, p.196).

The book **Models of Man** collected the papers which he published in early to mid 1950s. It is where the phrase *Bounded Rationality* appeared for the first time, in the introduction of Part IV (p.196). The phrase was, then much maligned in its uses and misuses, compared to the original definition and formalizations by Simon. Subsequently, bounded rationality became one of the frequently used terminologies of MBE. On the contrary in Simon's advocacy, human beings can solve their problems relying on heuristics and intuition without a given model in mind[28]. Therefore, there seems to be a mismatch between the contemporary interpretation of bounded rationality and its original definitions. In Simon's point of view, human beings have no capability and willingness to always find procedures to reach the best alternative, even if such a thing

[27] Agents and institutions and all other kinds of decision-making entities, in CBE, are information processing systems which in their ideal form are Turing Machines.

[28] This may well be one way for agents in CBE to transcend the limits of *Turing Computability* subject to the *Church-Turing Thesis*. However, we do not subscribe to the view that Simon assumed that the limits of Turing Computability are violable; we believe Simon could have resorted to oracle computations, when necessary, and also formalize via nondeterministic and alternating Turing Machines to encapsulate procedures - heuristics and other similar algorithms - that give an impression to the uninitiated that there are formal means to transcend Turing Computability.

is meaningfully definable, or make the 'Olympian choice.' Reasoning capabilities, formally defined as algorithmic procedures, are constrained by the limits of computability theory and, at an empirical level, by measures of computational complexity. Simon's definition of bounded (limited, procedural) rationality encapsulates different notions, such as limited attention, limited cognitive capacity of computation, satisficing, and sequential decision making (naturally dynamic) (Simon, 1955, 1956). That is to say, it is not evident and admissible to assume that human beings are able to exhaust all the information and make the 'best' choice out of it. Indeed, the notion of 'best' is given content via the formulation of problem solving by information processing systems in what is known in metamathematics as a decision problem. In such a framework one seeks algorithms to solve problems and classifies them as 'easy' or 'hard' using measures of computational complexity. There is no such thing as 'best' algorithm or a 'best' *heuristic*.

Therefore, the dynamics of *non-maximizing agents* can be described adequately in the following way. The *knowledge* we have, and *interpretation* of the world where we are living in , are associated with our *experience* and *memories*. Gradually, our tastes and understanding are *constructed*. The process of construction is the central pre-analytic, Schumpeterian visionary (Schumpeter (1954), p.51, ff), stage in the decision problem. Therefore, the pursuit for stable gain in taste and knowledge also relies on what has been constructed. This is one part of requiring a program to modify itself. The unhappiness and satisfaction which are associated with our aspirations depends on whether the desires are satisfied in terms of our anticipation. The aspiration level expends with satisfaction and shrinks with disappointment. Nonetheless, the memory that is stored in our mind prevents our aspiration level from becoming null. Thus, we are in the loop of unhappiness and satisfaction, a loop given formal content via the structure of a program for a Turing Machine or a heuristic implemented on one of them Simon (1991c).

This thesis argues that Simon's version of Bounded Rationality is not entirely comprehended and also that it has been misinterpreted and redefined. Many have developed their own versions of bounded rationality, however, none of them capture the essence and the richness of this notion as Simon intended it. The primary feature that distinguishes Simon's version from others is the explicit analogy between rational decision making and models of computation, where procedures are central in the latter. This argument is supported by **?**. Though there is a vast and growing body of modern literature on bounded rationality, we opt to characterize and underpin Simon's bounded rationality through Simon's own diverse contributions and computability theory. Hence, this thesis does not engage in a comparative analysis of all different strands of Bounded rationality found in the literature, though it can be an interesting exercise for the future.

2.4.2 Human Problem Solving

The notion of bounded rationality has been encoded implicitly and explicitly into the information processing system which was proposed in Newell et al. (1958a) and analysed thoroughly with detailed recording and interviews with human subjects in Newell and Simon (1972). IPSs have shown their capability of solving problems, such as cryptarithmetic, logic, and chess games, algorithmically, In their conclusion, it is suggested that task environments of greater complexity and openness ought to be studied. Thus, we can see that they are on the track of pursuing Turing's suggested program of research on *Solvable and Unsolvable Problems* (Turing, 1954).

Simon's notion of bounded rationality, encapsulated within the formalization of an IPS is, in turn, used in simulating (representing) human problem solving. Simulation, even if not precisely theorized in Simon's monumental work on **Human Problem Solving**, nevertheless is defined in analogy with the dynamics intrinsic to partial differential functions or their machine embodiment in the definition of the processing of information by a Turing Machine, or its specialized variants. Problem solving is the implementation, via *heuristics*, themselves algorithms, of *search processes for paths* from initial sates to the target states. The complexity of a problem solving process - the complexity, therefore, of the algorithm that is implemented in the search processes from initial conditions to 'halting' states - defines its hardness on a well-defined computational complexity measure. This also means that there could be problems that will be subject to the famous theorem of the *halting problem for Turing Machines*.

The methods that a problem solver uses are strongly associated with his or her memory and experience. The accumulated knowledge in the memory will form the heuristics - the current state of the program and its structure - to guide the problem solving him(her)self. Intuition is copiously invoked, and defined computationally and cognitively, in seamlessly leading the problem solver to one or another path at a node, when he or she faces a huge number of possible choices, in the *Nondeterministic Turing Machine* formulation of a problem.

Theory of Human Problem Solving

Literally, we need a problem and the problem solver to achieve problem solving, and the problem should be presented, recognized and understood. A problem is faced when one wants to do something about a particular task but does not know what series of actions can be done to implement it immediately. The three main factors that characterise problems are *the huge size of possible solutions*, *the dispersion of actual solutions* and *the high cost of search*. The problem space contains a set of elements which represent knowledge, a set of operators which generate new knowledge from existing knowledge, an initial sate of knowledge, a problem which is specified by a set of desired states, and the total knowledge available to problem solvers. The problem can be further formulated (represented by) *set-predicate* formulations and search formulation.

Representation In the former representation, the set of elements includes symbolic objects which are all possible solutions, not necessarily formally definable. Precisely, the set can be generated by a certain enumerative procedure. Thus, the problem solver will not be given the entire set, rather, is given a process to generate elements out of the set. This is exactly analogous to Brouwerian *constructive spreads*, arising out of *free choice sequences*. In a search representation, solutions as elements of a set, have the format of sequences. For instance, a proof of a theory contains a sequence of steps and chess representations contain continuations for some players.

Task Environment A *Task Environment* describes the attributes that are associated with the problem that problem solvers encounter. It consists of external and internal representations, where the former is the format in which the problem is exactly presented and the latter stands for the subjective representation the player applies. Accordingly, not only the presentation of the current problem, but also the *ability* and *intelligence* of the problem solver should be considered. This is because players with diverse abilities may perceive the problem differently. It should be made very clear that in Simon's framework of *human problem solving*, as well as in Turing's consideration of *Solvable and Unsolvable Problems*, concept like *ability* and *intelligence* are precisely defined, even if *pro tempore*, in terms of computability theory.

Information Processing System The information processing system which is capable of problem solving can be characterised as follows. An IPS is a serial, adaptive (dynamic), and deterministic system which receives input and generates output. It is composed of internal building blocks such as long term memory (LTM), short term memory (STM) and external memory (EM). LTM and STM share identical patterns but are distinguished by their size. LTM can contain all the symbolic objects without limitation, while STM contains only five to seven symbols. The fact of sequential decision making is inherent in IPS; moreover, how a problem solver retrieves objects from LTM to STM relies on heuristic search. This is exactly equivalent to the partial recursive function formalization of computability or a Turing Machine definition of computable process (cf. Davis, 1958), Chapter 1, in particular, and Part 1, in general; indeed, reading and mastering the foundational mathematics of computability theory simultaneously with an approach to problem solving in the Simon or Turing sense is the best way to understand all the equivalences inherent in all these formalizations.).

Heuristics

Heuristic is a *method* of "Rule of Thumb" that serves as a guide in searching. Intuitively, it is an ability and process to refer to one's own memory and experience and lead oneself to focus on appropriate subsets of knowledge. Without external help, one can learn and discover new knowledge by him/herself. Essentially, it is the ability of Machine to reconstruct its internal structure by itself. When an IPS receives information from the task environment, it generates the goals and the methods for the achievement by heuristic search. If heuristics cannot achieve a satisfactory solution, then either the heuristic method will be reprogrammed or the representation, namely, the internal representation in the task environment, will be reformulated. It will be clear that 'satisfactory' here is precisely defined by means of time and space computational complexity measures. In short, IPS and task environment are interdependent, and the process of change is *learning*. This is one way the human problem solver as a learner encounters him/herself as a learning machine.

In addition to bounded rationality and satisficing, Simon uncovered an interesting property, which became a recurring theme in his works, observed in many entities. In the very early 1950s Simon became familiar with Goodwin (1947) and the concept of *Near Decomposability*, culled from Goodwin's notion of unilateral (weak) coupling, began to be used in his papers and he applied it to diverse problems, such as identifying causality (Simon, 1952, 1953b, 1987; Simon and Iwasaki, 1988), counterfactuals (Simon and Rescher, 1966), aggregation (Simon and Ando, 1961), organizational behaviour (Guetzkow and Simon, 1955; Simon, 1993), complex system (Simon, 1973, 1990b), complexity (Simon, 1962, 2001), and human and machine thinking (Simon, 1995). Near decomposability has its rigorous mathematical characterisation, while conceptually the idea can also be connected to heuristics. Especially, in Simon (2002), near decomposability is the basis for causing a greater speed of evolution in organisms with a hierarchical structure. When the hierarchical structure is applied to the problem and problem solver in human problem solving circumstances, then evolution is analogous to learning and discovery.

Near decomposability in human problem solving can be interpreted as decomposing a problem into subproblems when the subproblems are not completely independent. In Polya's little book Polya (1945), "heuristic method" was demonstrated by an educator decomposing and reformulating a problem step by step for a student who is asked to solve the problem. Turing at the same time, also proposed his idea of a child's machine and education process in Turing (1950) (p.456). The influence of Polya, Turing and Goodwin are unambiguously evident in Newell and Simon (1972); and in Newell et al. (1958a) for their postulation of the internal structure of minds and the representation of task environments in human problem solving.

2.4.3 Classical Behavioural Economics and Computable Economics

Satisficing, SAT and Diophantine Problems

Velupillai advocated that the faithful encapsulation of Simon's bounded rationality and satisficing ought to be through models of computation in the context of decision problems. Particularly, he suggests posing problems of rational choices as SAT problems (satisfiability problem) Velupillai (2010b). A SAT problem looks for the truth assignments of the arguments which can make the global statement true. If such assignments can be found, then the SAT problem is satisfiable.

Solving SAT problems can be formulated, equivalently, as linear Diophantine equations, linear systems with nonnegative integer variables, or integer linear programming problems. Theoretically, SAT is NP-Complete (Cook's theorem), that is, a SAT problem is *not solvable* in nondeterministic polynomial time in its inputs, but can be *verified* in polynomial time. However, Velupillai has realized very recently that Simon's notions should be better formalized in terms of space computational complexity. In particular, SAT can be solved with a *linear* space algorithm. An intuitive explanation might be that in real human problem solving, subjects are never given sufficient amount of time to make decisions, rather, they are trained to restructure their short-term memory in order to process a problem in a given period of time. Subsequently, Velupillai has proved, via duality between computability and dynamic systems, that Simon's information processing system is capable of computation universality which is the relevant model of computation for rational choice. Furthermore. orthodox notions of rationality (through optimization) has been shown as a special (easy) case of the more general (difficult) case of SAT problem, in terms of models of computation in a decision problem context.

Chess and Go

Like many other *strategic games*, though the final target is to defeat the opponent in one's own way, Chess players care about many other actions while the game is ongoing. For example, it is important to capture, block and otherwise threaten the opponent. There are the sub-goals that come to players' mind alternatively, simultaneously to playing the game with the global goal, and in the pensive phases between moves. Being aware of the sub-goals, players can reduce their attention to relatively small groups of good moves and play accordingly.

Go and Chess are very fundamentally different. Go has no concrete configuration of terminal conditions, like "Check-mate" in Chess. Instead, a GO game is finished when both players pass, and the side who occupies greater territory wins. This is a most intricate 'stopping rule' for the program to implement the process of playing GO by a Turing Machine. The best moves in the Go games are even more ambiguous than the ones in Chess. Similarly, though it is *difficult* to list out all the terminal positions in Chess, it is very possible to decide whether each configuration belongs to the set of Check-mate. It is only possible for some of the games of GO. Unlike Chess, Go players rarely benefited by playing forcefully or aggressively - assuming these concepts can be given formal definitions in the relevant mathematics - because by doing that they can create unforeseeable 'dangerous' configurations to their own group as well.

A go game can be officially played on a 9 by 9, 13 by 13, and 19 by 19 board. Practically, Go games can be set from 2 by 2, 3 by 3, . . . , 9 by 9, . . . , 13 by 13, . . . , 19 by 19, . . . boards. The *combinatorial complexity* increases exponentially when the board size is enlarged. Thus, the complexity of Go games can be expanded theoretically to countable *infinites*, of a kind. This is the flexibility that the Chess game may lack.

The main task in playing Go is to enclose some areas on the board, so that the stones of the opponent which are in this area have no space to escape and are captured.

On the other hand, when a group of stones are in danger of being captured, the task is to create holes (eyes) to save a region. No matter how big the board size is, the *warfare* will be localized into separate regions on the board. When the game is being played, the attribution of some regions can be determined and it is known for both players that there is no need to fight on those regions any more. That is to say, the players will *decompose* the board into several blocks and try to invade or defend those regions. We conjecture, therefore, that near decomposability will turn out to be a useful way of representing some configurations in a game of Go.

It formalizing the Go games, it is reasonable to start with smaller sizes and apply them to the bigger board with the idea of decomposing into smaller blocks. In spite of the fact that the complexity of a game of Go increases exponentially with the board size, human players can reduce the practical complexity drastically by decomposing the board configurations and attack them separately. However, the Go board can *never be partitioned unambiguously*, this is where a plausible application of near decomposability can be envisaged. Despite all the differences of the two games, there are important similarities, too. Go players need to come up with sub-goals, such as *joseki* (ding shih, in Chinese), creating *atari* (da chi, in Chinese), making eyes and escaping from being captured etc., in order to resolve some situations.

2.5 Concluding Remarks

Science, in most of the cases, is built on asking and answering - often unanswerable - questions. In order to proceed properly, it is critical in most of the cases, that appropriate questions be asked. Decision theory deals with the problems of human choices, and plenty of models have been constructed and examined through the formalizations of orthodox mathematical economics, econometrics or experiments. Nevertheless, behavioural economics emerged based on the failure of orthodox economic frameworks. Anomalies have been collected and discovered with respect to the normative human behaviours which are predicted by orthodox economics. The central doctrines of orthodox economics are optimization subject to constrains and equilibrium analysis. Modern behavioural economics emerged as a field of finding and explaining anomalies in human decision behaviour. The difficulties of solving these problems (optimization and equilibrium) have been noticed; however, their solvability has not yet been questioned and challenged in Modern Behavioural Economics.

Solvability of problems, by problem solvers, requires formal characterisations of both concepts, neither of which has ever been attempted by modern behavioural economists. They are almost defined and characterised in classical behavioural economics and computable economics, as we have argued above.

Herbert Simon introduced the notion of “bounded rationality” and “satisficing” into economic fields along with their psychological and computational underpinnings. Intuitively, computability theory tackles the solvability of a problem and computational complexity theory measures the difficulty of solving a problem. Thus, if a program is designed to mimic human thinking, naturally, the computability of program has the counterpart in reasoning. Simon's ideal models of economic agents can be demonstrated by an *Information Processing System* and its nature of adaptation can be captured in the theory of "human problem solving". Within this framework, "anomalies" are, possibly, those that result in uncomputabilities, undecidabilities and unsolvabilities of problems, forced into solvable modes by inappropriate models, precisely definable as, for example, the use of finite automata where a Turing Machine is required, and so on.

If Simon's postulations are taken into account, then "Olympian" rationality (coined in Simon (1983b), p.19) is merely the special case of bounded rationality, and an optimization problem is, again, the special case of a satisfiability problem (satisficing), within the formal framework of metamathematical decision problems.

Apart from making, hopefully, clear distinctions between Modern and Classical Behavioural Economics, a more faithful encapsulation of Simon's notions - with clear computable underpinnings - was presented in this chapter. In continuing work, we are expanding the scope of Simon's notions of bounded rationality and satisfying, within a formal computable formulation, an exercise already begun in Velupillai (2010b) to the more general and complex cases of combinatorial game theory. Studying, for example, boundedly rational agents, choosing satisfying strategies in a game of Go will, we think, form a meaningful milestone in research along this line.

It is even possible to interpret some strands in Simon's thinking that human beings do try to *solve the formally unsolvable problems*, even while they somehow find 'only' the methods (heuristics) to satisfactorily solve them. This is to say, they try to make good decisions for only the near future, but with long-term targets in mind. No actual agent in his or her right mind (sic!) would even dream of formulating infinite horizon optimization problems in the economic sphere, except of course those endowed with Olympian notions of rationality, solvability, computability and decidability.

Chapter 3

Computable Foundations of Bounded Rationality[1]

3.1 Introduction

Bounded rationality is the central theme of Classical Behavioural Economics[2], championed by Herbert Simon. It is now ubiquitously accepted as a replacement for the otherwise infeasible notion of Olympian rationality, which was strongly disapproved by Herbert Simon. Contrary to the popular understanding, Simon perceived bounded rationality as the more general notion compared to Olympian rationality.

It is worth noting that the classical notions of rationality (also Olympian or substantive rationality) are infeasible both in empirical (human) and theoretical (recursion theoretic) senses. Bounded rationality, on the other hand, is adequately equipped to describe the real life decision processes that are of concern in the decision sciences and can encompass various notions of rationality used in other social sciences and psychology. It is inappropriate to simply characterize or formulate the external environments, and postulate theoretically that the best choices in such cases can always be found by the agents, before a method is provided explicitly, disregarding the procedural aspects of solving a problem. Given the difficulties in finding optimal solution in real life, one of the popularly held views on alternative directions to optimization is that people are in fact approximating these optimal solutions.

Although bounded rationality has often been defined merely as an implication of limited cognitive computational capacity by Simon, models of bounded rationality are far more complex and have deep epistemological implications.

[1]A modified version of this chapter (co-authored with K. Velupillai) has been published as Velupillai and Kao (2013).

[2]It is termed Classical in order to distinguish it from Modern Behavioural Economics. The distinction and pioneers of these two traditions is discussed in chapter 2 and Kao and Velupillai (2012b). The distinction of Classical and Modern Behavioural Economics in terms of their different interpretations of heuristics can be found in Davis (2012). A detailed of historic review of Modern Behavioural Economics with particular attention to Kahneman and Tverskey's work can be found in Heukelom (2012).

If we accept the premise *pro tempore* that human thinking is a process of composing language out of a set of finite symbols, then the limitation of human thinking and humanly attainable procedural knowledge are bounded by computability. That is, there exist some things that human minds can not think, some problems that human minds can not solve, procedurally.

If one takes into account the actual situations of human decision making, then we can soon observe that *knowing* the alternatives in the choice set and their characteristics are seldom straightforward. Once we know them, the concern is then about the difficulty involved in solving the problem of decision making. Computational complexity is one of the measures that describes difficulty associated with solving a problem. The limitations that are associated with economic agents engaging in procedural decision making is evident when viewed from the vintage point of computability theory and computational complexity theory. This has direct implications for what we understand as bounded rationality in the context of decision making where the emphasis is placed on the 'method'. This chapter elaborate the motivations of formulating bounded rationality in terms of computability theory, and strengthening it by invoking insights from computational complexity theory.

The aim of this chapter is twofold: First, to clarify, interpret and reformulate bounded rationality, remaining faithful to the definitions and vision of Herbert Simon. Second, to emphasize that bounded rationality ought to be placed and studied within a well structured algorithmic context, which Simon had been advocating all his life.

In particular, bounded rationality and satisficing are two important notions that are highly relevant for understanding decision making in general. The theory of Human Problem Solving incorporates these two essential themes. Although Simon almost never phrased his theories and concepts in terms of computability and computational complexity theories explicitly, he devoted himself to construct more realistic boundaries of human rationality, however, always implicitly within the framework of rationality as being procedural (algorithmic) and in turn, encapsulated by Turing computability and constrained by theories of computational complexity.

This chapter elaborates the computability theoretic underpinnings of the concept of bounded rationality and discusses the modelling philosophy involved in characterising economic agents. The discussion is proceeded along the lines of Turing computability, computational complexity and heuristics (empirical complexity), which is in the belief that this route was traversed by Simon, although it is not stated explicitly. In other words, computability theory and computational complexity theory are more appropriate as the underpinnings of bounded rationality. While viewing bounded rationality in the context of human problem solving, three aspects of problem solving become relevant: the existence of a method, the construction of a method, and the complexity of a method. This interpretation of bounded rationality can be used to demonstrate the impossibility and meaninglessness of optimization doctrines that largely dominates economic modelling and analysis. Moreover, human rationality should be meaningfully formalized in terms of other, appropriate tools that are faithful to the phenomena. A more important message that this chapter hopes to convey is that the bounds to human rationality depends on the complexity of different problems that the problem solver encounters and the research program on Human Problem Solving initiated by Herbert Simon is precisely along this direction.

In section 3.2, The analysis the definition of bounded rationality and discussions on satisficing, procedural rationality and heuristics can be found. In section 3.3, the meeting ground between Turing's computability and problem solving on the one hand, and Simon's work on Human Problem Solving and Information Processing Systems on the other is explored. Section 3.4 which is the core section of this chapter contains the computable and procedural underpinnings of bounded rationality.

3.2 Bounded Rationality

The term "bounded rationality" was coined by Herbert Simon in his introduction to the fourth part of his collected works - *Models of Man*. He wrote:

> "The alternative approach employed in these papers is based on what I shall call the *principle of bounded rationality*:
>
> *The capacity of the human mind for formulating and solving complex problems is very small compared with the size of the problems whose solution is required for objectively rational behavior in the real world - or even for a reasonable approximation to such objective rationality.*
>
> If the principle is correct, then the goal of classical economic theory - to predict the behavior of rational man without making an empirical investigation of his psychological properties - is unattainable."
>
> Simon (1957), p.198-199, italics in the original

Although the term appeared in 1957, the original idea of bounded rationality can be found in both Simon (1955, 1956) and eventually traced back to Simon (1947) which is the revised form of his PhD thesis.

After Simon proposed his initial models of rational behaviour, successive models of bounded rationality that were developed have been reviewed in March (1978), showing the many different directions in which it was developed by Simon and others, its mild extension and reinterpretations. It is necessary to clarify the difference between modelling human rationality in normative economics, and as it is done in the models that take real human decision making into account. It is evident from March's paper (loc. cit) that the sprouts of distortion and deviation from Simon's notion of bounded rationality has already appeared by then. Often, any inconsistent behaviour with respect to normative rationality is perceived as a mistake from the part of the agent. Consequently, bounded rationality has been explained as the mistake or short coming of human beings that arises due to a variety of factors (largely psychological) for about 50 years. Modern behavioural economics is not the only field that considers bounded rationality as a compromised concept from normative rationality. For example,

> "Alternatively, one can recall all of the deviations from normative specifications as stupidity, errors that should be corrected; and undertake to transform the style of exciting humans into the styles anticipated by the theory. This has, for the most part, been the strategy of operations and management analysis for the past twenty years; and it has had its successes. But it has also had failures."
>
> March (1978), p.597

Although on the surface of it, Simon's descriptions of bounded rationality might seem that the existence of a 'bound' is simply due to the limitations of psychological nature in human decision making, however, one figures out that this is an obvious concept to mathematised economics when one engages on a serious study of his Theory of Human Problem Solving. When we consider both the decision maker and the aspects of the associated environment, bounded rationality emerges naturally within such a setting. Given the characteristics of the environment and the decision maker, "satisficing" (first used in Simon (1956)) is the reasonable action to be pursued in a procedurally rational decision making setting, and heuristics are the means through which satisficing behaviour becomes possible.

Later, through the interpretation of the principle of bounded rationality, with computable foundations, where it be clear that bounded rationality is neither irrationality (Simon, 1957, p.200) nor approximate optimality (Simon, 1972, p.170).

3.2.1 Simon's bounded rationality

The models of rational decision making suggested in Simon (1955, 1956) do not require utility functions to be defined over the alternatives. They provide some important ideas regarding how a boundedly rational entity could be modelled. In Simon (1955), a simplified value function $V(\cdot)$ which takes only two values (1,0) was introduced. The binary values can be associated with "satisfactory and unsatisfactory", "accept and reject", etc. The domain of function V is S, the set of all possible outcomes which is mapped to A, a set of all behavioural alternatives. This is in order to distinguish the means from the ends. The rational decision-process is defined as

1. Search for a set of possible outcomes $S' \subseteq S$ such that the pay-off function is satisfactory ($V(s) = 1$), $\forall s \in S'$
2. Search for behavioural alternatives $a \in A'$, whose possible outcomes are all in S' through the mapping.

This process does not guarantee the existence and uniqueness of a solution, until the sequence in which the alternatives arrive and the dynamics of aspiration levels (a psychological concept) are incorporated into it.

In real life, the alternatives are often examined sequentially and the first satisfactory alternative evaluated is the one that is selected. The difficulty of discovering a satisfactory choice depends on the cost of obtaining better information regarding the mapping of A on S. Thus, if the aspiration level grows when the cost of search is low and declines when the cost of search is high, then this dynamic can lead to near-uniqueness and existence of a solution in the long run[3].

In Simon (1955), the focus is on suggesting a dynamic process for decision makers, without going into the details of the mapping between A and S. However, in Simon (1956), the focus is more on the other important aspect - the *environment*. The problem setting in this paper provides a general platform for constructing more elaborate models. Here, the organism is assumed to have a single and a fixed aspiration level - it needs only food. But, the food heaps are located in such a way that the organism has to walk in a maze where there are branches after each nodes. Each node is a possible location for food. This is combined with the constraint that its vision is limited and therefore it cannot see as far as it would like. However, if it sees a food heap in the range of its vision, it knows the way to reach the food. It has to eat the food before it dies of starvation and there is a maximal number of moves it can make after eating before its energy runs out. These are some of the parameters[4] regarding the environment that the organism faces and the "physical" constrains that the organism has:

- p: $0 < p < 1$, is the percentage of branch points, randomly distributed, at which food is found.
- d: is the average number of paths diverging from each branch point.
- v: is the number of moves ahead the organism can see.
- H: is the maximum number of moves the organism can make between meals without starving.

The first two parameters concern the environment (problem space), on how the targets are distributed and how big the problem space is. The last two parameters are regarding the organism on how far it can search and the capacity it can spend on searching. With these parameters, Simon was able to demonstrate the probability that the organism can not survive.

This setting can be applied to a much broader class of problems. The parameters do not have to be limited only by physical needs and constraints. Especially, the probability is not central in many realistic cases of decision making for Simon. For example,

[3]A similar modelling logic can be found in a simple job search model in McCall (1970). McCall's models contain the idea of the aspiration level which is attached to the variables that are concerned by job searcher when deciding to accept the job offers or not. The model demonstrate the negative relation between the desired level of wage and the cost of information. Besides, it is also suggested in McCall's paper (loc. cit) that non-adaptive behaviour could account for persistent unemployment in the period of recession (p.122). However the processes which exhibits the adaptive policy of decision maker is not the main concern in the current paper.

[4]These parameters are algebraic, rational numbers or other computable numbers.

in chess, a game that was studied intensively by Simon, the goals (some particular patterns) that a player might seek are not randomly distributed in the problem space.

Integrating the models in the two papers mentioned above, we can summarize the situation of rational decision making postulated by Simon as the following: There are always two aspects of decision making - the environment and the mechanism of the decision maker. The two aspects are highly interrelated. The characteristics of the environment or the problem space are the following:

- The alternatives are associated to discrete values
- The alternatives or the offers come in a sequence, while the order is not necessarily known
- The alternatives are of combinatorial nature in some cases

The characteristics of decision makers are

- Satisficing. (They are influenced by aspiration levels)
- Limited computational capacities (such as time and memory)
- Use of heuristics to search
- Some knowledge or clue regarding the stopping rule for searching (starting from any node)
- Adapting aspiration levels
- Knowledge of what to choose and what not to choose

The problem space is viewed as one that is like a tree, and this is 'explored' by the decision maker. This problem step-up of 'searching in a tree' is probably inspired from the means-end schema proposed in chapter IV of Simon (1947). In Simon's view, the description of the environment depends on the needs, drive and goals of the decision maker. This seems to underpin his 'maze' metaphor that many problems in life are like searching in a maze. Therefore, human decision making, which is part of human thinking activity, can be associated in fertile ways to many deep areas, such as, computer science, graph theory, formal logic, etc. The quotation below makes it clear that bounded rationality, which is grounded on the combinatorial nature of problem spaces in the real world, can lead to different frontiers.

> "Informally, a maze is a set of rooms connected by one way corridors. Certain rooms are designated goal rooms and one room is designated the start room. Thus, a maze is a directed graph with certain nodes or rooms distinguished. The maze is threadable if there is a path from the start room to some goal room"
>
> Savitch (1970), p.187

It is important to note that the probabilities that are used to calculate the likelihood of failing to survive or finding a solution in the model discussed earlier are trivial or meaningless in many real life problems:

> "From a still a third standpoint, the chess player's difficulty in behaving rationally has nothing to do with uncertainty - whether of consequences or alternatives - but is a matter of complexity. For there is no risk or uncertainty, in the sense in which those terms are used in economics or statistical decision theory, in the game of chess. As von Neumann and Morgenstern observe, it is a game of perfect information. *No probabilities of future events need enter the calculations, and no contingencies, in a statistical sense, arise*.
>
> From a game-theoretical standpoint, the presence of the opponent does not introduce contingencies. The opponent can always be counted on to do his worst. The point becomes clear if we replace the task of playing chess with the task of proving theorems. In the latter task, there is no opponent. Nor are there contingencies: the true and the derivable theorems reside eternally in Plato's heaven. Rationality in theorem proving is a problem only because the maze of possible proof paths is vast and complex."
>
> Simon (1972), p.170, italics added.

However, it is debatable whether the opponent can be counted on to do the worst in games like Chess. The search space of games like Chess or Go are already certain, but only waiting to be discovered. We can also view the opponent as using heuristics in her own mind, in order to decide what the possible reacting moves of his opponent will be. Facing such uncertainty as complexity, the decision maker has to incorporate some mechanisms for her to terminate the searching process. This leads to satisficing[5].

Many problems have relatively closed and a pre-defined problem space, though the problem space (tree) may be massive. There are many other problems which are far more complex, for example, finding a particular quotation amongst the books in a library. However, most of the time, the material that one is looking for is just in the vicinity, but it is hard to find a good heuristic to reach it.

Satisficing and Optimizing

Satisficing is the other pillar on which Simon's behavioural economics stands on. Here, the decision maker does not look for an optimal choice, where the 'procedure' of search will itself lead him/her to choose a satisfactory outcome as and when one encounters it. This would mean that even though there might be an outcome that could yield a higher level of satisfaction, the choice process stops once a 'good enough' alternative that matches the aspiration level is met. Simon also comments on the relation between satisficing and optimizing and that the latter is a special case of the former.

[5] Minimax is such satisficier.

> "A satisficing decision procedure can often be turned into a procedure for optimizing by introducing a rule for optimal amount of search, or, what amounts to the same thing, a rule for fixing the aspiration level optimally....
>
> Although such a translation is formally possible, to carry it out in practice requires additional information and assumptions beyond those needed for satisficing"
>
> Simon (1972), p.170

For a decision maker, the act of optimization would require a before-hand knowledge of all the available options and the associated outcomes. Moreover, she also requires a method for listing all the options and to compare each of them. When the decision maker is confronted with multiple goals, then association between choice and outcomes gets even more complex. As Simon remarks, this is both unrealistic and excessively demanding (Simon, 1956, p.136). Simon further emphasizes that the optimizing approach facing real-life complexity is indeed approximate optimization. The statisficing approach, on the other hand, is linked with the dynamics of aspiration levels and tackles the problem very differently.

Procedural and Substantive Rationality

Although we do try hard to make a good choice, when in the face of time constraints, it is also likely that a choice just made is bad or disappointing. Furthermore, even though we may know our goals and sub-goals, it does not mean that we know 'how' to attain them. Even if we have an abstract map of a city, and we would like to travel from A to B, it does not actually mean that we can reach B from A without any difficulty. We still have to work through the paths, follow the signs, check the map, etc.

When we begin to claim that "This decision maker is satisficing.", the next question we may ask primarily becomes "What are the procedures that the decision maker uses?", instead of "What does the decision maker choose?". The fundamental distinction between Simon's approach and the other theories that invoke behavioural traits, such as the modern behavioural economics, is the insistence on 'methods' or 'procedures' involved in choosing and their centrality in the theory of decision making. The link between a procedurally rational choice and computation is present from the very outset in Simon's scheme. The insistence here is on the complexity of this decision process, in terms of the effort devoted in doing it.

> "The search for computational efficiency is a search for procedural rationality, and computational mathematics is a normative theory of such rationality. In this normative theory, there is no point in prescribing a particular substantively rational solution in there exists no procedure for finding that solution with an acceptable amount of computing effort."
>
> Simon (1976), p.133

An important way to procedural rationality for it to be reasonable and illuminating, is to observe how problem solvers come up with solutions in reality.

Heuristics

Since it is evident that during the course of decision making, procedures, either in terms thinking or action are involved, understanding decision making would require a knowledge of these procedures. The question then is to ask, what these *procedures* are and how these *procedures* present themselves in the context of decision making, how they are discovered and develop dynamically over time. Simon calls these procedures heuristics[6], which are nothing but *methods* to achieve some goals. They can be discovered by oneself, taught by teachers or forced by regulations. A clever remark on heuristics is that "a method is simply a plan that you use twice" (Newell and Simon (1972), p.835). That is, heuristics represents the methods that human beings actually use to search in a problem space. *They are nothing but algorithms.*

> "Most weak methods require larger or smaller amounts of search before problem solutions are found, but the search need not be blind trial-and-error–in fact, usually cannot be, for the search spaces are generally far too vast to allow unselective trial and error to be effective. Weak methods generally incorporate Polya's idea of "heuristics"–rules of thumb that allow search generators to be highly selective, instead of searching the entire space."
> Simon (1983a), p.4570

It is quite obvious that heuristics are heavily associated with one's experience, knowledge and cognitive capacity. In terms of decision making among finite alternatives, whose mapping between actions and outcomes is combinatorial, heuristics are used in the following three respects.

- What to generate?
- How to evaluate?
- When to stop?

These heuristics are in fact *algorithms* whose commands should be executed step by step and this is evident from the general approach that underpins Newell and Simon's theory of Human Problem Solving. Algorithms are formally connected to symbolic structures which are underpinned by computability theory. Physical symbolic systems such as human beings and digital computers, as Newell and Simon pointed out, can process only a finite number of steps in any given interval of time. However, the finiteness to which the most general model of algorithms - Turing Machine

[6]He called them "rules of thumb" until he got to know the term "heuristics" from Polya (Polya, 1945).

- appeals is in many cases not strong enough to show the severe limitation that human minds have to confront (Newell and Simon, 1976, p. 120). Empirical boundaries correspond to the level of complexity that the human minds can actually handle. A study of heuristics is crucial in order to understand how human beings handle problems whose complexity is beyond the empirical boundary. In other words, heuristics act as procedures that help reduce the problem to a level of complexity which can be handled.

The approach described in Human Problem Solving that encompasses heuristics is underpinned by computational complexity theory, which in turn is based on computability theory. Moreover, the significance of heuristics is not revealed until some algorithmic impossibilities concerned with procedural decision making are formally proved. To further explore this connection, we need to examine the interconnections between the approaches of Turing and Herbert Simon, which is attempted in the next two sections.

3.3 From Turing to Simon: decision making as problem solving

The ground where bounded rationality and computability meet is comprehensively presented in Newell and Simon (1972), where symbolic systems can be adopted to understand human thinking, especially in the activities of information processing. In the theory of Human Problem Solving, the vague idea of an "environment" is precisely formulated into a "problem space" and a problem solver into an Information Processing System. In addition to Simon's approach, the notion of "complex problems" (appearing in the definition of Bounded Rationality by Simon) needs to be given a precise definition, and it is done within the context of Turing's computability theory and computational complexity theory.

The word "computer" in this chapter is used interchangeably to refer to human computer (mind) and digital computer.

> "Moreover, since *Homo sapiens* shares some important psychological invariants with certain nonbiological systems - the computers - I shall want to make frequent reference to them also. One could even say that my account will cover the topic human and computer psychology."
> Simon (1990a), p.3, italics in the original.

The crucial element here is to capture the link between the intuitive notion of thinking that is involved in deciding or problem solving, and the structured machine that can replicate this. For this, there should also be a formal notion that encapsulates this intuitive notion. This was achieved by the seminal works of Alan Turing, through a formal definition of algorithm and a mechanism to encapsulate the intuitive notion of effective computation in the form of Turing machines. This, in turn, forms the intellectual backdrop in which Simon and Newell developed their theory.

Man and Machine

The relationship between man and machine has also been discussed philosophically. One can look back at the discussions along the lineage of Philosophy of Mind back to Aristotle, Descartes and other precursors. More specifically, the investigation on the man-machine metaphor is inspired by the *mechanistic* debates which belong to a branch of Philosophy of Mind (Shapiro, 2003; Feferman, 2009). In order to extend the argument on mind and computation in this chapter more intuitively, one of the intentions that make people built machines can be emphasised. Boring (1946) suggested that a good way to understand ourselves as a creature which has conciousness is to build machines or robots which behave just like us. For achieving this, different questions have to be asked successively and we have to answer these questions by introspection. This intention drives the triangular relationship among artificial intelligence, psychology and computer science.

In many circumstances, thinking is highly dependent on languages, despite that the exact relationship between language and thinking is still controversial and vague. Languages are sets of strings which are composed of symbols from finite sets (Sipser (1997), p.14). Simon was aware of the importance of language and the possibility of linking that to the idea of problem solving in the economic sphere, through a model of computation proposed by Turing (Simon (1991b), p. 192-193).

Once this view of relating thinking to the process of symbolic computation is adopted, then the other pieces of the theory fall in place. The notions of what is achievable, procedurally (algorithmically) solvable, the level of complexity that one can handle becomes clear. In the problem solving environment, modern digital computers have already acquired the abilities that human beings have, such as recognizing symbols, store, copy and compare symbols. Another common feature of computers and human beings is that they are both finite serial (symbolic) processors, and this property shows that they are fundamentally the same systems. Simon stressed the infeasibility of a procedure of optimization by showing that digital computers which overpower human beings in terms of their physical computational capacity find their strength insignificant in front of the complexity of real world problems (Simon (1976), p.135).

Understanding our own limits is one of the lessons we obtain from digital computers. The same logic should be applied to economic decision makers, as Simon suggested in the following:

> "The human mind is programmable: it can acquire an enormous variety of different skills, behaviour patterns, problem solving repertoires, and perceptual habits.....There seems to be no escape. If economics is to deal with uncertainty, it will have to understand how human beings in fact behave in the face of uncertainty, and by what limits of information and computability they are bound."
>
> Simon (1976), p. 144

One of the recurrent questions that sceptics ask is whether machines can also have emotions which play nontrivial roles in the real decision making circumstances.

Simon has acknowledged this point very early on in Simon (1967). Simon claimed that emotions do play a very important role in behaviour and decision making and suggested ways in which emotions can be incorporated into Information Processing Systems. As for exploring the roles of emotion in cognitive modelling, Simon considers that emotion is a gross phenomena or a category of diverse mechanisms among which "interruption mechanism" can be effectively modelled. However, emotion in decision making is not the focus of the thesis, thus this issue will not be discussed in detail in this thesis.

Simon's many approaches (by viewing the problem solver as an information processing system) to develop models of satisficing and bounded rationality are deeply influenced by Turing's invention of the machine (possibly abstract) operating on symbols to understand the behaviour of human thinking. When Turing wrote "computer", he really meant "human computer"; the first digital computer was not even born by then! We can add more theoretical rigour to Simon's distinction of satisficing and optimization by appealing to Turing's computability. We examine three different versions of bounded rationality below - computability theory, computational complexity theory and finally in terms of the theory of Human Problem Solving.

3.4 Computable Foundation of Bounded Rationality

In this section, the focus is on suggesting a computable foundation for bounded rationality. The building blocks can be initiated by relating the setting in the paper to that of another famous 'decision problem' - Hilbert's 10th problem and the idea of general procedure for solving decision problems and the formal definition of the notion of a 'finite process'. The insights can be drawn from the last published paper of Alan Turing dealing with solvable and unsolvable problems. This provides a link between Turing's work on computability and Simon's work on human decision making as problem solving, where computation and procedures (algorithms) take a centre stage. Complexity of decision making is analysed within this framework.

> "Theories that incorporate constraints on the information-processing capacities of the actor may be called *theories of bounded rationality*"
> Simon (1972), p.162

There are three aspects on which the theories of bounded rationality are anchored. They are *solvability* of the problem, *difficulty* of this solvable problem and the *heuristics* that are used to deal with this complexity, i.e., the computational complexity measuring the difficulty of a solvable problem.

3.4.1 Solvable and Unsolvable Problems

Let us begin with Hilbert's 10th problem - a decision problem, which was posed by the mathematician David Hilbert in 1900 as one of the 23 problems that were posed as challenges for 20th century mathematics and mathematicians.

"Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a *process according to which it can be determined by a finite number of operations* whether the equation is solvable in rational integers."

Hilbert,1900, Paris, Second International Congress of Mathematicians, italics added, Devlin (1988), p.141

This problem could not be answered without a formal definition of "a process according to which it can be determined by a finite number of operations" in other words, an algorithm. Until about 1936-1937, when the definitions of such a finite procedure were defined, the mathematicians had only an intuitive notion of an algorithm.

Although Turing does not seem to have made an attempt to solve Diophantine decision problem, he created the Turing Machine for solving another decision problem posed by Hilbert in 1928 (Turing, 1936) and theoretical developments based on Turing machines contributed eventually to the negative solution of Hilbert's 10th problem, 70 years after the question was posed. The *decision problem* that concerned Turing was general: Is there a systematic procedure to decide whether **a** given problem(puzzle) is solvable or not?[7] This decision problem regards all those problems which can be transformed into *substitution puzzles*. The answer to this was proved to be negative by Turing [8]. The negative solution to this decision problem indicates that we need to develop specific procedures in order to decide specific problems. There is no general solution - i.e., algorithmic procedure - to any given problem. This has a direct bearing on the theories of decision making that rely on optimization, without addressing the procedural aspects.

In order to understand how he arrived at the negative solution, let us have a close look at the problem setting:

"*Given any puzzle we can find a corresponding* substitution puzzle *which is equivalent to it in the sense that given a solution of the one we can easily use it to find a solution of the other. If the original puzzle is concerned with rows of pieces of a finite number of different kinds,then the substitution may be applied as an alternative set of rules to the pieces of the original puzzle. A transformation can be carried out by the rules of the original puzzle **if and only if** it can be carried out by the substitutions and leads to a final position from which all marker symbols have disappeared*."

Turing (1954) p.15, italics in the original.

He further wrote "In effect there is no opposition to the view that every puzzle is equivalent to a substitution puzzle."[9]

[7] In computability theory, this could be translated as "Is there a machine by which one can tell whether a set of languages is recognizable?"

[8] The formal definition of substitution puzzle - algorithm - was defined during 1936-7 as mentioned in Turing (1954) by several people - Turing, Church, Post and others - at about the same time.

[9] See Turing (1954), p.13 for the example of a substitution puzzle.

The production rules[10] that are introduced in Turing's example follow type 0 grammar[11], though, the time at which Turing proposed it was before the Chomsky hierarchy was defined (Chomsky, 1956, 1959).

Type 0 grammar is the superset of the hierarchy and includes all recursively enumerable languages. Turing Machines are the most general kind of symbol operators which are capable of recognizing the languages generated from all types of grammar.

In order to prove that there is no systematic procedure to decide, when given any puzzle, whether the puzzle is solvable or not, Turing claimed that there are two equivalences:

- The equivalence between the puzzles and the substitution puzzles.
- The equivalence between the substitution puzzles and the systematic procedures.

A substitution puzzle consists of its set of rules (substitution pairs) and starting position. And a systematic procedure is a puzzle in which there is never more than one possible move in any of the positions which arise and in which some significance is attached to the final result. The puzzle described for systematic procedure is also called "the puzzle with unambiguous moves."

Then this kind of "puzzle with unambiguous moves" is applied in the argument to prove the negative answer to the existence of a systematic procedure for deciding the solvability of a puzzle. Before he started to prove, he transformed the sentences of the set of rules into the same form of the starting position, i.e., a string of symbols. Therefore, there are many different strings that can actually represent the same set of rules of a substitution puzzle. We can represent a puzzle as $P(R, S)$, where R is the row of symbols describing rules and S is the starting position. It is reasonable that the puzzle $P(R, R)$ is also considerable, that is the starting position is the same string of the row of symbols describing rules. Provided that the puzzles being discussed are those with unambiguous moves, then these puzzles will be substituted with their rules, step by step, until no rules can be carried out, and report a certain result. In Turing's example, it is either W or B. That is to say, $P(R, R)$ has as its final result either W or B, it cannot have both possibilities. This kind of puzzles are classified into two classes:

- Class I is to consist of sets R of rules, which represent puzzles with unambiguous moves, and such $P(R, R)$ comes out with the end result W.

- Class II is to include all other cases, i.e.. either $P(R, R)$ does not come out, or comes out with the end result B, or else R does not represent a puzzle with unambiguous moves. We may also , if we wish, include in this class sequences of symbols such as BBBBB which do not represent a set of rules at all[12].

[10] Its formalism can be traced back to Post (1947)

[11] For the formal definition of grammar, see Tourlakis (1984), p.256

[12] Maybe the class should include the situation such as WW, which is a result of the puzzle when no rule can be further applied to it.

It is assumed that there exists a systematic procedure for deciding whether a puzzle is solvable or not. At the same time, this systematic procedure can be transformed into a substitution puzzle whose set of rules is K. Naturally, K has unambiguous moves and it always comes out with final result no matter what R, the puzzle of interest, is. In particular, it will come out with, say B, when R belongs to class I, and W when it belongs to class II. Then, when we look at the puzzle $P(K, K)$ to be investigated, we will find inconsistent results. That is, we should be able to classify that $P(K, K)$ belongs to class I or II. But according to the substitution puzzle K, it has the potential to come out both possibilities, as a result, we could not classify it into either of the two classes. This leads to a contradiction!

This demonstration towards showing that there is no general algorithm for deciding whether a puzzle is solvable or not suggests that we need to seek for separate algorithms in order to decide whether a kind of problem is solvable or not, given the initial puzzle and the desired outcome.

At this point, it should be possible to move on to introduce the formal definition of a substitution puzzle, which sheds light on Hilbert's 10th problem.

Undecidable Decision Problems

Definition 3. *Decision problem: as to the existence of an algorithm for deciding the truth or falsity of a whole class of statements ... A positive solution to a decision problem consists of giving an algorithm for solving it; a negative solution consists of showing that no algorithm for solving the problem exists, or, as we shall say, that the problem is unsolvable.*

It should be noted that when Turing developed his intuitive idea of computation, he always took into account the natural limitations of (thinking) human beings. A Turing machine[13] can be seen as the mathematical formalism of a human computer.

In Turing (1954), he stressed the importance of having formal definitions or representation of "a problem" and a "systematic procedure". Turing had shown mathematically that there exists unsolvable decision problems by using both Cantor's diagonaliztion method and letting the supposedly existing machine encounter itself and leading to a contradiction involving the halting problem. He was able to prove that the halting problem is unsolvable.

[13] A description of a Universal Turing Machine:

> "It is possible to invent a single machine which can be used to compute any computable sequence. If this machine U is supplied with a tape on the beginning of which is written the S.D of some computing machine M, then U will compute the same sequence as M."
> Turing (1936), p.241-2

Here, "S.D." denotes the *standard description* which is a transformation of a sequence of quintuple instructions, e.g. $q_i S_i S_j L q_m; \ldots$ (meaning the current state q_i, the scanned symbol S_i, printed symbol S_j, move to the left, successive state q_m) by encoding the sequence into a sequence of letters. The commonly used quadruple machines were formulated in Post (1947). These alphabets can be further encoded into integers.

Computability Theory

Church defined algorithm (effective calculability) with the λ-calculus, and Turing defined it in terms of Turing Machines. They were proved to be equivalent definitions and the intuitive notion of algorithm captured by these definitions imply the so-called Church-Turing Thesis. If this thesis is true, then the halting problem for Turing machines is unsolvable. Church (1938) mentioned that the intuitive notion of a effective procedure can be formalized into three different ways: Turing Machines (Turing, 1936), λ-definability (Church, 1936) and the general recursive function (Kleene, 1936a). The equivalence of the three notions had been proved in Kleene (1936b) and Turing (1937). The equivalence of recursiveness and computability enables us to apply the definition of recursive function to prove more classes of computable functions. (see Davis, 1958, Ch.3)

> "Concurrently with Turing's work appeared the work of the logicians Emil Post and (independently) Alonzo Church. Starting from independent notions of logistic systems (Post productions and recursive function, respectively), they arrived at analogous results on undecidability and universality - results that were soon shown to imply that all three systems were equivalent"
>
> Newell and Simon (1976), p.117

The consensus on the notion of effective calculability was reached in the late 1930s and this led to the development of computability theory. Both Herbert Simon and Alan Newell were recipients of the Turing Award in 1975 for their contribution to the human problem solving approach, which they initiated in the mid 1950s together with Cliff Shaw. It is evident that their work, where computation plays an important role, was grounded on Turing's contributions.

Hilbert's Tenth Problem is Unsolvable

Definition 4. *A set S of ordered n-tuples of positive integers is called* ***Diophantine*** *if there is a polynomial $P(x_1, \ldots, x_n, y_1, \ldots y_m)$ where $m \geq 0$, with integer coefficients such that a given n-tuple $< x_1, \ldots, x_n >$ belongs to S if and only if there exist positive integers $y_1, \ldots, y_m$ for which*

$$P(x_1, \ldots, x_n, y_1, \ldots, y_m) = 0$$

The "negative solution" to Hilbert's tenth problem was finally proved by Yuri Matiyasevich in 1970. He proved it by applying the Fibonacci sequence to the hypotheses and theorems built along this line by Martin Davis, Hilary Putnam and Julia Robinson[14]. The solvability of Diophantine equations has been applied to many diverse problems. The first application of this in the context of economic theory was by

[14] For more details, see Matiyasevich (1994)

Velupillai. One of the notable applications is in the context of effective playability of Arithmetical Games [15].

For the empirical concerns, unsolvability of a problem does not really stop people from looking for a solution for it. The following quotation shows the pros and cons of proving the unsolvability of a problem:

> "After all, showing that a problem is unsolvable doesn't appear to be any use if you have to solve it. You need to study this phenomenon for two reasons. First, knowing when a problem is algorithmically unsolvable is useful because then you realize that the problem must be simplified or altered before you can find an algorithmic solution. Like any tool, computers have capabilities and limitations that must be appreciated if they are to be used well. The second reason is *cultural*. Even if you deal with problems that clearly are solvable, a glimpse of the unsolvable can stimulate your imagination and help you gain an important perspective on computation."
> Sipser (1997), p.151, italic added.

The cultural reason appeared in the above quotation is relevant for us to understand how bounded rationality is underpinned by computability theory.

Simon was concerned both empirical and theoretical questions of decision making and he grounded himself on the basis of Turing Computability. He later looked for computational complexity in average cases or empirical complexity that is relevant for human problem solving. Now that we have seen that there exist algorithmically unsolvable decision problems, we can appreciate how Turing computability should be the outer limit for human rationality or machine computability. Anything that goes beyond Turing computability is clearly meaningless (especially for procedural decision making) since even the most powerful abstract computing machine cannot solve such a problem, even in principle. But this can only be an outer boundary of how far procedural rationality can go in theory because the notion of pure computability in theory does not take into account time and space limitations, which are essential to solve a problem or compute a function. They become particularly important in the case of human decision making. This only reinforces the conclusions and strengthens the concepts that Simon advocated.

3.4.2 Computational Complexity

Before we start to look for the solution for a problem, it is natural for us to ask whether it is solvable or not, though in most of the cases, we have to try to solve them anyway. It is also intuitive that there are fairly easy problems and difficult ones. Equivalently, we can determine that the problem space of a given problem is more complex than others. Unless we have better heuristics, it is reasonable to say that the more complex problem is going to take more time and effort. Simon offered some insights on how

[15] Velupillai (See 2000, p.108)

to describe complexity which can be associated with computational complexity, which itself is based on computability theory:

> "How complex or simple a structure is depends critically upon the way in which we describe it. Most of the complex structures found in the world are enormously redundant, and we can use this redundancy to simplify their description. But to use it, to achieve the simplification, we must find the right representation."
> Simon (1962), p.481

The above fragment can be interpreted in terms of computational complexity theory, where the complexity of a problem is determined by the time and space requirements of an algorithm that solves the problem. The rigorous definitions on time and space complexity and those of different complexity classes can be found, for example, in chapter 7 & 8 in Sipser (1997).

There are three aspects of problem solving: the inherent solvability of a problem, the procedure to solve a problem and the complexity of the procedure. Provided we have Turing's abstract model of computation, we can use this idea to construct an abstract machine for solving a particular problem. We can then analyse the number of steps or memory that the algorithm would require, approximately, without going about to count the precise time and space required by the problems of the same kind. This helps us to have an idea of the associated difficulty of a problem we are dealing with before we really start to solve it. Computational complexity provides a more solid, inner boundary of bounded rationality with Turing computability as its outer boundary. Although the scale of time steps and space (memory) that computational complexity theory regards is normally pretty large, it is important to have a general idea of tackling a problem by knowing the complexity of the algorithm which solves it. In theory, the reducibility among problems is also used to study the complexity without actually constructing a real algorithm.

As far as problem solving is concerned, according to Turing's interpretation, a decision problem is to decide whether one can change a string of symbols to the desired string of symbols, by only using a set of rules that are given in advance. Knowing that a decision problem is unsolvable leads us to ask different questions and try to solve them, otherwise, it provides no practical help when we try to solve a problem. We need to find a set of rules for a substitution puzzle, that is "algorithm", to solve our problem. However, even if we have an algorithm to solve a certain kind of problem, it does not guarantee that we can solve the problem within the desirable period of time. If the problem involved is complex, it can demand immense amount of computation by the problem solver. Time complexity and space complexity are very useful and standard tools for providing measures of quantitative ideas on how much effort is needed for solving a problem.

When we have an algorithm for solving a problem, we can look at its general behaviour and analyse how many time steps and the space or memory it would require. Time complexity tells the number of steps needed for running an algorithm, and space complexity takes care of the memory needed. Time and space complexity are the functions of size of input, for example, playing 3-disk Tower of Hanoi needs much less time steps than 10-disk Tower of Hanoi. In many cases, it is very difficult to obtain the exact reduced form of time and space complexity of an algorithm. Therefore, in computational complexity, asymptotic notations, such as $O(n)$, are used to present the asymptotic behaviour of an algorithm as the asymptotic approximation of the true function behind it. When large inputs are concerned, exponential time grows drastically faster than polynomial time, and the problem becomes unmanageable very quickly.

It should be remembered that there exist always more than one method to solve a problem, therefore, the complexity of a problem is determined by the method that solves it[16]. The complexity of a problem mentioned in theory are associated with the complexity of the most efficient algorithm ever found. Precise computational complexity in time and space is very hard to attain, but it can be approximated. It can be approximately estimated by constructing an abstract machine for solving a problem and analysing the order of growth of complexity of that machine.

Space Complexity

Space complexity has attracted relatively less attention and effort compared to time complexity, despite the powerful result that $PSPACE = NPSPACE$[17]. By default, when the complexity of a problem is discussed, time complexity is the one that is referred to. Arguably, it is because whether $P = NP$ is one of most popular unsolved problems. One of the claims in this chapter is that in the domain of human problem solving, space complexity is at least as important as time complexity. Although, there is no doubt that the architecture of human brain has the potential to store huge amount of knowledge, the amount of information that minds can process at a given moment is severely limited.

For example, it is very tough to calculate $4593 * 3274$ in the mind for an ordinary person, unless this person has pencil and paper at hand or he/she is an expert of arithmetic calculations. Such calculation requires a certain amount of temporary memory which is a function of input size. In terms of time limitation, minds are constrained by attention span, apart from other externally imposed time constraints, eg. a chess player has to make a move in 5 minutes. How minds are constrained by time and memory varies with different contexts and structure of the problems and among different persons. Furthermore, these two dimensions should not be completely independent, i.e. the memory constraints affects the time which is needed for solving a problem and vice versa. Therefore, it is important to investigate the time complexity of a problem (or an algorithm) together with the space complexity; consequently, we will be able to know what kinds of heuristics are needed based on these two dimensions. Space complexity is even more crucial when the problem concerned requires no aid of external memory.

[16] Sipser (See 1997, p. 229-231), for an example.
[17] This is implied by the first theorem in Savitch (1970).

In spite of the fact that the time and space complexity of an algorithm can be analysed, human beings are constrained very differently from (digital) computers. We normally have only a certain amount of time to make a decision, and we have very limited working memory (no matter expert or layman) to process this task, regardless of the presumably unlimited long-term memory. We are forced to use those algorithms which will be able to halt within certain amount of time, by applying the knowledge and experience we have in the long-term memory. Although, we are often assigned to a task like "find the best person for this job", we are not able to solve it as an optimization problem. At best, we will have the criteria for appropriate candidates and consider only a small group of people. Depending on the time and memory we are supplied with and the procedure we should go through, we have to be selective to different degrees.

Complexity of Combinatorial Games

Turing (1951) suggested that a machine (e.g. Boring's robot metaphor, see Boring (1946), p.177) should be programmed to learn to play the games like Chess, Go and Bridge. Newell and Simon initiated the project on creating a program which learns to play good Chess in 1954. While they pinned down the investigation of heuristics on proving theorems in *Principia Mathematica* to start with. The program they designed was hand-simulated first and interpreted into machine language which gave birth to the *Logic Theorist* which was the first example of human problem solving (Newell et al., 1958a). However, *Logic Theorist*'s success was rejected by Kleene for it contributed nothing new and it also received an unjust criticism from Hao Wang:

> "There is no need to kill a chicken with a butcher's knife. Yet the net impression is that Newell-Shaw-Simon failed even to kill the chicken with their butcher's knife. ... To argue the superiority of 'heuristic' over algorithmic methods by choosing a particularly inefficient algorithm seems hardly just."
>
> Wang (1970) p.227

It was a unfortunate that Wang had misunderstood Simon's attempts (Simon (1990b), p.209-210). Newell and Simon were involved in finding procedures used by human beings and used this information to construct the program *Logic Theorist*. What are hueristics, if not algorithms?

Chess is the recurring example and an important one for Simon, it is also one of the examples of complex combinatorial problems[18] which make brute-force algorithms infeasible. Even though the problem space of chess is closed and certain, the massive size of the game tree prevents human beings or even supercomputers to use brute search algorithm. Let us take the number of the possible continuations of Chess, which is approximated in Shannon (1950) as an example. If we want to know, from the

[18]For a brief definition of combinatorial problems, see Moret and Shapiro (1991), p. 1-2

beginning of the game, whether Black or White has a winning strategy, we have roughly 10^{120} variations to calculate; when each branch reaches its end, we can see whether that branches leads to a win, loss or a draw. Suppose we have a high-speed computer which uses only one microsecond (10^{-6} a second) for one variation, for searching the whole problem space, it will take 10^{100} million years! Obviously, in real life, different actions should be taken, that is why we can always find plenty of Chess tactical guides in the bookshops.

The game of Go can be one of the possible paradigms to go beyond Simon, by programming information processing systems of solving it. In a mathematical sense, Go is more flexible and general than Chess, because its board size can be, in principle, unlimitedly enlarged. The philosophy and heuristics of Go are not necessarily consistent with those of Chess, which makes human problem solving more interesting. Go was also in the choice list of Simon, but they finally chose chess[19].

This game has already been studied by combinatorial game theory. It is shown in Lichtenstein and Sipser (1980) that Go is in PSPACE-Hard, although later on, it is shown that no PSPACE algorithm can exist (Robson, 1983) for Go. Chess, for example, is not yet shown to be in PSPACE-hard. In game theory or strategic interaction models the idea is to look for a winning strategy in a game. The problem asked in the Robson (loc. cit) is "given an arbitrary Go position on an $n \times n$ board, determine the winner".

However, Go was proved to be PSPACE-hard and EXP-time complete, which means Go is in the exponential time class and it is proved that there can be no PSPACE algorithm for solving Go (Robson, 1983). Problems in exponential time are considered to be among the most difficult problems. This result suggests that deciding whether Black or White has winning strategy at an arbitrary position is practically infeasible. Clearly, any expectation for finding an optimal strategy to win in such a complex setting is both meaningless and futile. On the other hand, it also provides a perfect setting for studying 'actual' modes of decision making without being tied to the search for optimal strategies.

3.4.3 Computable Economics

Two important theorems in Velupillai (2000), are relevant in this context and it provides the way for economists to reconsider the philosophy and methodology of 'rationality as a process' in decision making. Computable economics, stated briefly, is an algorithmic approach to economic theory. It is consistent with the research agenda of Simon and we establish the superiority of bounded rationality as a theoretical notion for understanding human problem solving, within this framework. The two theorems from chapter 3 of Velupillai (2000) that are relevant here are the following:

Theorem 1. *"There is no effective procedure to generate preference orderings."*

[19]The reasons for which are stated in (Newell and Simon (1972), p.664-665). Given these reasons, we can only wonder whether they would have chosen the game of Go as their paradigm if they had been from the Orient!

Theorem 2. *Given a class of choice functions that do generate preference orderings (pick out the set of maximal alternatives) for any agent, there is no effective procedure to decide whether or not any arbitrary choice function is a member of the given class.*

Velupillai (2000, 2010a,b) has formalized decision making as an act of choosing a subset from a finite, non-empty, countable set, as opposed to uncountably infinite sets, by using a choice function. Solving optimization of the latter case is equivalent to solving linear integer programming problem. Velupillai transformed the linear integer programming problem in to the optimization problem of a combinatorial system, and then constructed abstract Turing machines to study the characteristics of problem solving.

In order to show that Bounded rationality is more general than Olympian rationality in this framework, Velupillai proceeds as follows: First, the notion of the behaviour of a suitably programmed (universal) Turing machine is shown to be equivalent to the rational behaviour of the "economic agent" and connect the idea of adaptive process with that of a dynamical system, for studying its computation universality. And it is proved that only those dynamical systems which are capable of computation universality are consistent with rational behaviour in economic theory. He further proves that no trajectory of dynamical systems capable of computation universality can be usefully related to optimization in Olympian rationality. On the other hand, it was also demonstrated that a boundedly rational Information Processing System in the decision problem framework is capable of *computation universality*. In the decision-problem framework, optimization is merely a specially case. The overall conclusion is that bounded rationality in the context of decision problem is more general than Olympian rationality.

Satisficing as Satisfiability Problem

Simon showed that Olympian rationality is a special case of bounded rationality by appealing to the act of satisficing. He suggested descriptively that a model of satisficing can be turned into optimizing by setting the aspiration level at a optimal level. This lucid point can be supported mathematically by applying the results in combinatorial complexity.

> "The real mathematical content of of satisficing is best interpreted in terms of satisfiability problem of computational complexity theory, the framework used by Simon consistently and persistently - and a framework to which himself made pioneering contributions."
> Velupillai (2010b), p.9

The satisfiability problem is one of the important problems in modern computer science. In Velupillai (2010b), it is demonstrated that SAT problem is the meeting ground of Diophantine problems and satisficing, in turn this connection leads to the conclusion that bounded rationality is the superset of Olympian rationality, which Simon had been advocating. Velupillai showed that satisficing can be formulated into Satisfiability problems and optimizing is characterised by Linear programming. Therefore, Olympian rationality can be shown as a special case of Satisficing, because Linear Programming is in P (Khachiyan, 1979) and SAT is in NP-Complete (Cook, 1971). P is believed to be a subset of NP. Many problems in computational complexity theory are reducible to a satisfiability problem.

Definition 5. *The problem Satisfiability (SAT) is defined as follows: Given a Boolean formula ϕ, determine whether there is an assignment that satisfies it (i.e., more formally, SAT is the set of all satisfiable Boolean formulas). The Boolean formula has the Conjunctive Normal Form.*

The formal decision problem framework for a boundedly rational information processing system can be constructed in one of the following ways: systems of linear Diophantine inequalities, systems of linear equations in non-negative integer variables, integer programming. Solving the former three problems are equivalent in the sense that the method of solving one problem provides a method to solve the other two as well. The Integer Linear Programming (ILP) problem and SAT can be translated both ways, i.e, one can be transformed into another.

Simon's models where decision making is an act of satisficing can be provided with procedural content. Essentially, a decision maker reviews the alternatives one at a time and stops searching when a satisfactory object is encountered. The procedure can be shown with the following steps:

1. A decision maker applies a kind of mechanism T which draws the objects x_i to be reviewed in a sequential manner, i.e. $x_1, x_2, \dots, x_n, \dots \sim T$. x_i belongs to the set X which is a set of combinatorial objects which are relevant to the problem to be solved.

2. The decision maker has m goals (multiple goals, of course it is possible that $m = 1$). The m value functions $V_i : X \to Z$ (The output of the value function are rational numbers or computable numbers. Without loss of generality, they are assumed to be integers here.) which are characterised by their respective goal take the object x_i and output the values. Therefore, each object x_i is now represented as a vector of its associated values $\boldsymbol{x_i} = [x_{i1}, x_{i2}, \dots, x_{im}]$

3. The decision maker then applies a satisficing procedure S on x_i. The output of of S is either Yes or No, i.e. $y_i = S(\boldsymbol{x_i}),\ y_i \in \{0, 1\},\ j = 1, 2, \dots, n, \dots$. What S does is to compare $\boldsymbol{x_i}$ with the aspiration vector $\boldsymbol{b}$ which contains the aspiration levels associated with the m goals. $\boldsymbol{b} = [b_1, b_2, \dots, b_m]$.

$$y = \begin{cases} 1 & \text{if } \boldsymbol{A}\boldsymbol{x_i} - \boldsymbol{b} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

, where A is an m by m matrix which contains only integers. In this case, A is an identity matrix.

4. The whole decision making procedure will halt when the first solution which makes $y = 1$ is found.

In the above procedure, the aspiration vector should be generalized into a function of time or other factors. Otherwise, the procedure with static aspiration levels alone does not guarantee that the solution can be found in any reasonable time. The mechanism T which generates the sequence of x_i does not exclude the possibility that the same object can be generated again. This mechanisms T and V_i are examples of the algorithms which are called *heuristics* in human problem solving domain. It is not difficult to observe that this satificing process is formulated as integer linear programming (ILP). It has been proved that the methods which solve ILP can also be applied to SAT, i.e. they are equivalent[20]. ILP is in NP complete and it can be *verified* in polynomial time, such as the procedure S in the above example.

3.4.4 Human Problem Solving

Given the above premise, bounded rationality is interpreted in the context of Turing computability. This also provides a bridge from computability to computational complexity, by linking satisficing to decision problems. We are then able to study the complexity of a given problem as a general idea of how difficult a problem is. It is legitimate to ask whether there is an absolute computational complexity beyond which the human mind can not handle any more. Note that the computational complexity of a problem depends on the algorithm for solving it, and brute-force algorithms are normally the benchmark, especially, when an optimization problem is proposed. Satisficing and heuristics now have very important roles to play in reducing the computational complexity of a problem. This is where we would need some input from psychology and cognitive science, in other words, this is the meeting ground of economics, psychology, cognitive science and artificial intelligence . It is evident that human minds rely on very simple heuristics (algorithm) to solve a problem in a satisficing manner and also rely on the support of external memory and knowledge.

The theory of human problem solving, where human beings are viewed as information processing systems[21], is a well constructed paradigm of understanding practical problem solving of an individual or an organization. Simon made this contribution as soon as he proposed his idea on bounded rationality. A brief introduction on theory of human problem solving and information processing system which is the computation model of the human problem solver can be found in (Kao and Velupillai, 2012b, p.63-65). A critical step that Simon took is to ask different questions, i.e. changing the focus from "What is my winning move?" to "What is my next good move?". By doing so, we are already able to abandon brute-force algorithms and ignore its astronomical complexity. Consequently, we might reach the next question - "what is my

[20] Although technically translating a problem of ILP into SAT is not trivial.

[21] The computer programs which represents the information processing system are constructed from investigating the subjects' problem solving protocols painstakingly.

goal for this move such that I can choose a good move?", "what is the more attainable goal of the larger goal?" This approach of dissecting a general problem into its means-end structure has been fully realized in the study of Human Problem Solving (Newell and Simon, 1972).

Therefore, it is argued here that it is meaningful and useful to bring in the measure of time and space complexity into Human problem solving. Subsequently, we can bring in the context of the task environment and look for the possible heuristics for reducing the complexity of a problem. No matter how difficult a problem might be, a physical system has to output an answer with limited resource - time and space - or crash without achieving anything it is supposed to. Heuristics are the methods or algorithms that are used to reduce the complexity of a problem to levels that can be handled. They involve generating a subset of alternatives, evaluating the alternatives with the ability of pattern recognition associated with accumulated knowledge, stopping evaluation. The first two aspects of using heuristics involve pattern recognition, and the last one involves both pattern recognition and aspiration levels. Heuristics are thus central to Human Problem Solving. The three aspects of heuristics - what to generate, how to evaluate and when to stop, are at the core of HPS. The mapping between actions and eventual outcomes - i.e, the evaluation of alternatives - is the most difficult task that the problem solver is presented with.

3.5 Concluding Remarks

The proposal of this chapter is that Simon's Bounded rationality and Human problem solving theory should be formally understood within the framework of computability theory and computational complexity theory. By extracting the procedural content of decision making, heuristics are considered as algorithms. In computational complexity theory, the complexity of a problem is analysed not only through the structure of task environments but also through the heuristics that problem solvers used to solve the problems. In this framework, we are able to show that bounded rationality via satisficing is the general notion of rationality.

Simon's vision and definitions regarding bounded rationality were always intuitive and straightforward. He thus left a large canvas for others to build models based on bounded rationality. Simon's notions concerning bounded rationality can be interpreted more clearly in the light of alternative mathematical formalisms, those which are faithful to the notion of procedural decisions. Also, models should be constructed according to different situations and the actors who handle those situations. In this chapter, it is argued that the two aspects of human problem solving - the task environment (problem space) and problem solver (algorithm) should be distinguished and then studied.

By appealing to computability theory, it is shown that bounded rationality is a superset of Olympian rationality. Subsequently, boundary of rationality is further narrowed down to the inner boundary - the one established by computational complexity.

Finally, it is suggested that Simon's empirical boundaries can be further approached - along the same methodology- by investigating the heuristics which are the algorithms (methods) that are used by human beings in problem solving circumstances.

Chapter 4

Human Problem Solving in Classical Behavioural Economics

4.1 Overview

Consider the following statement: “This problem is complex”. It can carry different meanings to different people unless there is an objective measure to define how complex a given problem is. At the meta level, computational complexity provides such an objective measure for the general solutions (in terms of procedures) of the problems. When someone engages to solve a problem and gradually becomes experienced and sensitive to the same kind of problems, the complexity faced by this person/organism should have been transformed to a very different architecture.

The aim of this chapter is twofold. First, to highlight the fact that Simon has connected his theories of human problem solving (Newell and Simon, 1972) and theories of bounded (procedural) rationality in economic domains (Simon, 1959, 1972, 1978b), both explicitly and implicitly. There is no qualitative difference between these two. The former deals with puzzle-like complex problems, and the latter deals with human decision making in general. These two domains together span what we call as Classical Behavioural Economics[1]. Second, it suggests that the game of Go is a good domain for CBE to further strengthen and enhance the ties between the theories of human problem solving and bounded rationality in economics.

A robust phenomena discussed in this thesis is that human beings suffer from severely limited short-term memory (attention) and computational power, despite their ability to solve massively complex problems. One intuitive interpretation of this phenomenon is that the problem may no longer appear to be so complex to the problem solver. It is from this perspective, we start to investigate heuristics and satisficing that assist the problem solvers to cope with complexity. Likewise, it needs to be reiterated that any humanly impossible or meaningless procedures ought not to be applied to model human decision makers.

[1]This term was coined by Velupillai. For a detailed discussion refer to chapter 2 of this thesis.

In Classical Behavioural Economics (CBE), satisficing is the guiding principle and Information Processing Systems are the models which inherently perform satisficing in the context of problem solving. This chapter attempts at suggesting that the game of Go as an important paradigm for CBE. Despite the very general characteristics of Information Processing Systems, the ones that have been built in the past and realized are task environment specific. Many domain specific architectures have been applied to revise what is known as general and invariant characteristics of Information Processing System in Human Problem Solving. Chess is one such prominent domain. A legitimate question here would be whether Chess, as a domain, has exploited the full potential of the general structure of Information Processing System. This thesis provides a negative answer to this question and suggests that the game of Go is a promising candidate. Formally speaking, Chess is a special case of Go.

CBE seeks for a general enough Information Processing System that can be systematically transformed to many economic problems. Many problems asked by economists may seem straightforward because the inherently multi-dimensional goal in economic decision making is collapsed to an intangible and single-dimensional *utility* in the case of individuals (microeconomics). Such formalisation is very remote from what can be concretely installed into an Information Processing System. This thesis avoids entering the multi-dimensional and ill-structured economic domain, instead focuses on a domain that is general enough and still touches the border of well-structured problems. The game of Go bridges well-structured and ill-structured domains despite the boundary between these two types of problems is ambiguous.

In the previous chapters, Information Processing System has been formally connected to bounded rationality, satisficing and theory of algorithms. The first part of this chapter reveals the finer structures, components and deals with the construction of Information Processing Systems in the context of human problem solving. Having this in mind enables us to have deeper understanding of the problems and the problem solvers. The second part of the chapter is the discussion about Chess and Go within the scope prescribed by the first part of the chapter.

An immediate, perhaps a superficial justification for Go being a good candidate for CBE, in order to extend the human problem solving approach, may be that Go is the new challenge facing the contemporary Artificial Intelligence community. The difficulty of constructing a competent Go playing program shows that the massive number of combinations in Go, which is one of the reasons why it is relevant for CBE. The other reasons[2] why Go should concern CBE are i) its flexibility in representation:

[2]Apart from the formal reasons why Go is chosen, there is an observation regarding personal curiosity. Both Simon and Turing discussed about Go as one of the possibilities. It is interesting to know that

> "[Alan] was able to take advantage of the visit [to New York, in November, 1942, while working on decrypting the Enigma code] to make sundry purchases, including a 'Go' board, and was proposing to attend the 'Go' club meetings in his neighbourhood to discover the American standard of play."
>
> Turing (1959) p.72

unlike in Chess, the players of Go have only one weapon - the stone. The operation is extremely simple: place a stone on a intersection of grids on the board. In each round, the player needs to decide where to place it, despite the fact that the process of deciding it can be very prolonged. A stone has no particular character individually, but they can assume a variety of roles collectively; and ii) its richness in multi-faculties of the players: Go is a protracted game; at different stages of the game, very different strategic and tactical skills are required. The later property reflects and reveals its connections with real world problems, such as politics, management, economic development.

To clarify the relevance of Go as a paradigm for CBE, it is helpful to look at the issues that concern CBE. This will constitute the main discussion in the second part of this chapter. Go is an analogue of many life-related decision makings in the sense that the goal is about how to survive and how to make a better living without being destructive. Instead, in Chess, it is about how get the supreme target through a mortal war. It is not unusual that at the end of a game, both kingdoms are almost ruined. Go's relevance to many other problems is also discussed by one of the pioneers of computer Go:

> "In all of these respects, the goal structure of Go again resembles that of many real world political, economic, and business problems. Success in these domains depends upon making the most effective use of limited resources to achieve a superior dynamic balance among multiple competing goals."
>
> Reitman et al. (1974), p.125

The importance of studying Go and other economic problems through the human problem solving approach has been emphasized by showing the gap that exists between the realistic or realizable procedures of decision making and those in the eyes of theorists of computer science, cognitive psychology and behavioural economics. This gap that describes structural and qualitative differences is often concealed by the increasingly prominent achievements (i.e. competition and tournament) of contemporary AI. CBE is an interdisciplinary approach that encompasses the merits of these three fields and attempts to reveal and reduce the gap.

Cognitive Science and Computer Science are the engines of Artificial Intelligence, Behavioural Science and in turn, for CBE as well. It is gradually noticeable that Go is being considered as a better domain in Cognitive Psychology and Computer Science. Along the same lines, this thesis aims at suggesting Go as a good or better domain to extend Herbert Simon's CBE. In order to construct logical and coherent arguments for supporting this statement, the following questions are considered by this and the successive chapter:

1. Are board games good subjects for CBE in general?

2. What characteristics does an Information Processing System should have in order to play good Go?

3. Are these characteristics constitute a super set of those characteristics which are required for Chess?

This chapter servers as the transition phase between the previous chapter of foundations of CBE and the next chapter on the game of Go. It motivates the idea behind building Information Processing Systems for playing Go and their relevance for CBE, at the same time, familiarizes the reader about the contents of the theory and early development of human problem solving in detail. Section 4.2 presents the highlights of theory and practice of human problem solving. Section 4.3 discusses the qualitative differences between Chess and Go with particular reference to contemporary AI.

4.2 Human Problem Solving

The project of Human Problem Solving can be traced back as early as 1954, when Allan Newell initiated an idea of writing good chess program and proposed to investigate the satisficing aspects and rules of thumb, that were the main concerns for Simon, while playing chess. This intention that induced a series of research questions has led to a broader research scheme unintentionally.

The theory and practice of human problem solving is a well developed and ever growing scheme which came out of a cognitive movement that happened around 1950s[3]. The ideas underpinning this came from a multidisciplinary ground constituted by psychology, linguistics and computer science at the very least. The pioneering contributions of this movement are claimed to be Bruner et al. (1956), Miller (1956), Chomsky (1956), Chomsky (1959), Newell et al. (1958a), and Broadbent (1958).

The theory of human problem solving (henceforth, HPS) is in fact a theory of information processing, which views human beings as *information processing systems* (henceforth, IPS), at least when they are solving a problem. This approach takes *thinking-aloud protocols* as data and constructs computer programs as models or theories. Information processing systems are in charge of the procedural activities or organisations that happen between a given stimulus and the response out of the problem solvers. Protocol analysis which dissects the verbal reports can shed light to these procedures.

The activities that concern the study of HPS can be classified in terms of three attributes: across tasks, across individuals and across time. The time dimension can be further divided into performance, learning, and development. Newell and Simon (1972) analysed three different tasks, and each task is performed by one specific subject respectively. Learning and development were not the main concern of their project in the early stages.

[3]Around the same time, a seminar organised at Dartmouth gave birth to the modern concept of Artificial Intelligence which had much in common with the emerging discipline of Cognitive Science.

Being aware of the non-existence of a general solution to an arbitrary given problem (undecidability) in theory of computation, and not intimidated by the worst-case scenarios proved in theory of computational complexity, Simon was eager to seek for the algorithms that can find a *solution* to a problem using limited amount of computational time and effort. HPS approach was driven by the presence of severe scarcity in terms of human computational capacity, in contrast to those in the neoclassical economics. The general and factual observation is that human beings are able to solve *difficult*[4] problems despite very small processing power as a result of small attention (short-term memory). This very fact has shaped the design and characterization of IPSs.

Chess, among other many tasks that are painstakingly studied by Simon and his colleagues, has shown its fruitfulness in providing access to many issues in HPS. Simon's contribution to Chess is the stepping stone towards the game of Go, which is the subject of this thesis. As a matter of fact, the first mission (see the Overview) of this chapter is not a new idea to Simon. He had intentionally sowed the seeds of HPS in his many celebrated works on (organisational) decision making, economics, social sciences and other scientific fields.

This section summarises the components and characteristics of HPS and presents a concise report on its early development along the line of Chess.

4.2.1 Theory of Human Problem Solving

Theory of HPS unifies the invariant characteristics of human beings solving complex problem across different domains and individuals. Thus, it is richer in qualitative structures than in quantitative assessments. The premises of its scope are i) the existence of the mental level processing regardless the lower level of neurons, ii) human beings suffer server limitations in computational power and attention (working memory) and iii) human begins rely on knowledge-intensive reasoning in some cases. Different premises can be investigated through different task domains to different extents. The first premise is linked to some philosophical and methodological debates on reductionism, which is not the main concern and argument of this chapter. It is thus taken for granted here.

At the early stages, computational models of selective search were created in a stark contrast to infinite inductive power assumed by the mathematical models in game theory. At the later stages, the battles shifted to being between brute-force algorithms and knowledge-based algorithms. This thesis conjectures that the divide is around 1972 when Newell and Simon (1972) was published. I believe that both limitations in computational capacity and domain-specific knowledge are equally important and interrelated for HPS on non-trivial domains. A good task domain can reflect the importance of the two premises simultaneously. Chess was one such good domain

[4]The difficulty in this context can be better understood with computational complexity, however, problems are not characterised only in terms of hard-easy divide.

and several computer programs and cognitive models were developed from studying it. However, it is the twilight of Chess in the context of HPS, and I believe, the dawn of Go.

Discussions and practice concerning different focus points about the theory in varying degrees can be found in Newell et al. (1958a), Newell et al. (1958b), Simon and Newell (1971), Newell and Simon (1972), Simon (1978a), and Simon (1990a). In the following part of this section, the general components of HPS and IPS will be described with the hope of providing a miniature version of this approach, so that the terminologies and concepts that are frequently used in the later parts of the thesis will not appear to be too alien. Various IPSs which tackle different aspects of HPS will be briefly mentioned.

From the outset, the act of HPS is performed by a problem solver, characterized as an IPS interacting with a *task environment*. When a task is given and accepted by the problem solver, he/she represents the domain with which he/she works as a *problem space*. A problem space is an internal and *subjective* representation of the external and objective task environment. Thus, the act of problem solving is the selective search conducted in the problem space.

There are only very few invariant characteristics of the IPS across tasks and individuals. While the general property is unambiguous: the systems are only capable of serial processing, they have small short-term memory and unlimited long-term memory with fast retrieval (tens to hundreds of milliseconds) and slow storage (seconds to tens of seconds). These properties impose strong constraints on the ways by which the systems can seek solutions to problems in larger problem spaces.

A Problem

A problem contains an initial state and a final (goal) state and a problem solver is engaged in finding a solution with which the final state can be reached[5]. Travelling and searching for a path in a maze is an abstract formation of a problem. A final state can be a configuration of symbols or some criteria over the symbol structures. When a problem is well posed, there will be a set of all the possible alternatives, P and a subset S in P represents the desired solutions. Recall that $S \subseteq P$.

When the task is to make a good move in Chess, P is the set of all legal moves, and S is a set of good moves that are defined by some criteria. In the case of theorem proving, P is the set of all possible sequences of expression in a formal language; S will be a set of sequence that proves a specified theorem. In this formulation, the standard procedures applied to find the solution out of a problem space is a) generation of candidate alternatives that are possibly in S and b) verification of the alternatives and deciding whether they are indeed in S.

The domain of problems can be classified according to at least two attributes: i) the definiteness of their structure and ii) the amount of semantic information needed

[5] If a problem solver does not know what is desired to do, then there is no "problem"; in other words, the problem is not properly defined. As a result, at each given step, problem solver has to decide what to do and then decide how to do.

in problem solving. Therefore, some problems are well structured (puzzle-like), and are easier to be carried out in laboratories; the other problems that we encountered in many real-life circumstances come across as ill-structured. The type of well-structured problems has been widely examined and given good explanations by the information processing theories. As for the semantic attribute, it measures the extensive knowledge that is required in problem solving. Chess is a semantically rich domain, for example, and Tower of Hanoi is less rich in semantics, compared to say, chess.

Problem Space and Task Environment

The structure of the task environment[6] determines the possible structures of problem space that the subject indeed works on; the structure of the problem space in turn determines the possible algorithms (heuristics) that can be used to solve the problem. It is important to note that problem solving activities are supposed to take place in the problem space.

A problem space is spanned by knowledge states and operators. The simplest and straightforward problem space consists of nodes generated by all the legal moves. Each node in the problem space is a state of knowledge that is available to the problem solver. The operator applied on a node leads to another node of knowledge stage. Many problem spaces that are represented by all legal sequences (continuations) are massive, and some other problems, such as Tower of Hanoi, have relatively small problem space (constrained by the number of legal moves). Trial-and-error search in the relatively small problem space is already unendurable for human problem solvers, let alone huge problem spaces. This is a property that may suggest that the difficulty of a problem in HPS is not related to the size of the problem space.

Hence, the precise size of a problem space is not the main concern in information-processing theory of problem solving. The genuine skill of the problem solvers lie in their ability to extracting information from the task environment and organizing the problem space so that the highly selective (heuristic) search can be adopted. Apart from the size of a problem space, which is irrelevant in the context of information-processing, the problem spaces also differ in their structures as well. The predictability and redundancy of the structure are the bases for selective search. A difficult problem will acquire more complex heuristics to solve it, regardless the absolute problem size.

Laboratory Techniques

Since HPS is concerned with the procedures and activities that receive input (stimuli) and produce output (respond), mere observations on what the subjects respond to is by definition not sufficient. The standard data collecting techniques are recording verbal reports and eye-movement tracking.

[6] Task environment can be described from the task instructions and the primary characterisation of the subject.

There are several ways to obtain verbal reports, such as retrospective report (posterior report), thinking aloud protocol that is given simultaneously while the task is being conducted and spontaneous report (without intervention by the experimenter). Retrospective report is shown to be too biased and spontaneous verbal report cannot guarantee the necessary amount of report. Only thinking aloud protocols has been proved to be faithful in reflecting the cognitive processes that lead to the subjects' solutions of the problems. Many were sceptical about the thinking aloud assignment to the subject, for it may affect the actual cognitive processes. Ericsson and Simon (1980, 1993) discuss these issues in detail and provide the valid defence over the scepticism expressed.

The thinking aloud protocols can be long and tedious, despite they contain answers to the questions asked about the tasks. The Problem Behavior Graph drawn from thinking aloud protocols is one of the tools for displaying the search in the problem space. The *production system* (defined later) employed by the subjects can be induced from the problem behaviour graph. Theories or models (computer programs) are constructed and verified through the building blocks that are extracted from *thinking aloud* protocol.

4.2.2 Information Processing Systems

The IPS is a model of human problem solver in problem solving environment. It provides sufficient explanations to human thinking[7]. The most elementary unit of IPS is the symbol. Its structure, components, and specific properties are gradually realized by the studies on various task environments. It is like solving a huge jigsaw puzzle[8] of IPS. The specification of human problem solver as IPS has gone beyond the stage of a hypothesis, instead it is a living thesis[9].

Information processing is an activity that happens in the brains, however, it is defined as the level which is above the level of the organization of neurons and, in turn, the chemical and electrical level of neurology. It is the level that processes information as symbols and IPS describes the mechanisms of cognitive processes. Its procedures are conventionally realized by digital computer programs. Computer programs which can carry out symbolic information processing is formally defined as a set of difference equations.

An IPS is a serial and an adaptive system. It consists of an active processor, input (sensory) and output (motor) systems, an internal Long Term Memory (LTM) and Short Term Memory (STM) and an External Memory (EM)[10]. There are several elementary

[7] For example, Logic Theorist, a program shown to be capable of proving theorems in *Principia Mathematica* has its most primitive processes require no more than the ability of reading, writing, storing, erasing and comparing symbols (patterns).

[8] I personally prefer mosaic to jigsaw puzzle in this metaphor. The basic elements of mosaic are meant to be tinnier than normal tiles. The final pattern that shows in mosaic can be both aesthetic and impressionistic, and thus leaves a lot of freedom for perception.

[9] One can imagine its counterpart in the theory of computation being Church-Turing thesis.

[10] EM is associated with what problem solver can visualize immediately during the problem solving

processors[11] whose inputs and outputs are held in the STM. Both STM and LTM share the identical structure but differ in capacity. It possesses a class of symbol structures, the goal structures, that are used to organize problem solving. Its program is structured as a production system, the conditions for activation of a production being the presence of appropriate symbols in the STM augmented by the foveal EM (the line of sight, the central focus of gaze). The parameters of IPS such as the capacity of STM, the time required for storing a new chunk into LTM, the entrance and exit time required in STM, etc. can be estimated through experiments.

The behaviours of IPSs in HPS can be roughly classified into three phases of routine: alternative generators, alternatives evaluators and stopping rules. An IPS should know when to stop evaluating a move and when to stop searching. When different IPSs of different level players are concerned, the IPS need not differ from each other only in terms of a single attribute, i.e., not only the amount of knowledge in the long-term memory that is different, but also the representation and the method of recognition and retrieval of knowledge are different. IPSs were programmed by Information Processing Language (list-processing), which is an assembly and low-level language.

Short-Term Memory

Miller pointed out the magical integer seven that is associated to the limitation of human information processing is repeatedly discovered in many experiments of absolute judgement and immediate memory. In absolute judgement, number seven is associated with the limited number of alternatives that the subjects can distinguish without confusion, and in immediate memory, number seven refers to the number of 'chunks'. A chunk is generally defined as an organized or grouped unit that is familiar and can be recognized by the subject. An relevant example is in English language. Each alphabet can be a familiar chunk, and in turn, the words composed by permutation of alphabets are also chunks. Likewise, familiar phrases or sentences composed of words are chunks as well. By enlarging the organization of a chunk, one can essentially remember more bits of information without altering the limitation on the number of chunks[12].

In IPS, STM holds about five to seven symbols, but only about two can be retained for one task, while another unrelated task is performed. All the symbols in STM

process, e.g. a piece of paper he can write on for solving arithmetic problems, and the chessboard. It has access times of the order of a hundred milliseconds (the saccade, rapid eye movement) and read times to STM of the order of fifty milliseconds. Write times are of the order of a second per symbol for overlearned external symbols.

[11]IPS's elementary processes take times of the order of fifty milliseconds, but the overall rate of processing is fundamentally limited by read rates from LTM and EM.

[12]It is not hard to imagine that a chunk can be as big as a whole article! The technical problem derived from the chunk-hypothesis is that it is then difficult to find the clear cut between chunks. Simon has encouraged the readers that the experiments held for verifying the chunk-hypothesis can benefit from taking the chunk-hypothesis as the premise and finding techniques to 'estimate the parameters' (Simon, 1974)

are available to the processes (i.e., there is no accessing or search of STM). The severe (size) limitation of STM actually directs the search strategies and processes used. An initiative of an arbitrary hypothesis that a particular state of knowledge is true, which leads to further possibilities of knowledge states (nodes) in the problem space, requires the nodes visited by the processors be *backed-up* in the STM. When this kind of search is a burden to STM, it will be avoided.

Long-Term Memory

LTM has an unlimited capacity and is organized associatively, and its contents are symbols and structures of symbols. Any stimulus that becomes a recognizable configuration (chunk) is designated in LTM by a symbol. Writing a new symbol structure that contains K familiar symbols takes about $5K$ to $10K$ seconds of processing time. Accessing and reading a symbol out of LTM takes a few hundred milliseconds.

It is important to note that the infinite capacity of LTM is different from the conventional mathematical understanding; it is emphasized that"Infinite, in this context, need mean only: far more than he could possibly scan during a problem solving session" (in Newell and Simon, 1972, p.819). Besides, one of the characteristics of LTM is that it is consecutively accumulated with more and more knowledge during problem solving session, might even last till the lifetime of the problem solver.

Apart from the capacity of memory and the speed of processing, the organisation of LTM is also an important issue. The symbols in LTM are associative, and the concept of associativity can be traced back as early as Aristotle. A symbol or a group of symbols (chunk) is familiar and recognized not because a chunk is innate in the memory. A chunk as a stimulus or cue can be recognized and evoke other symbols because the internal representation of this chunk is retrieved and evokes its designations. The chunks are accumulated through learning. In particular, the association of symbols in LTM of IPS has been represented as *list-structures*.

Production Systems

The production system is a natural form of an IPS. A production system consists of a collection of *productions* that is specifically ordered. A production has the form like this:

$$\text{If Condition} \ \rightarrow \ \text{do Action}$$

The order of the productions can resolve the conflicts when several conditions are satisfied at the same time. The condition component consists of a set of tests. When the conditions are satisfied, then do the action, otherwise do nothing.

There are at least two levels of production systems. One is at *perceptual level*: the conditions concern the stimuli (visual or aural) and the actions transfer symbols from LTM to STM; the other is at a general level: the conditions testify the symbol in STM and the action operates the symbols in either STM or LTM. A system having these two levels of production systems, for example, is sufficient to produce a move in chess. Tower of Hanoi is one of Simon's favourite examples to build various production systems for solving it. The alternatives are goal-driven, stimulus driven, pattern driven and a neat production system for interaction with the rote memory (Simon, 1975).

In this chapter by taking the game of Go as the paradigm, I argue that an IPS of Go should be equipped with higher level production systems, i.e. semantic level and reasoning level. Undoubtedly, these higher level production systems do not yet transcend the scope of operating symbols, but they enhance the hierarchy in to which the domain-specific knowledge can be organized.

Evaluation

When a problem space is structured, the task of the problem solver (IPS) at any given point of time is to choose the operator to apply to the current node in the problem space. It is certain that some evaluation is involved. Recall that each node in the problem space is a state of knowledge, and some states of knowledge might lead to desired goal state. *Means-ends analysis* uses the distance between the current node and the target node as a measure to choose ideal operators. The criterion for a good operator is thus the one which reduces a good amount of this difference. Such methods are applied by the subjects in the task of theorem-proving.

Different problem spaces will suggest different methods of evaluation. The concrete IPS that is developed by using means-ends analysis is called General Problem Solver (GPS). GPS is a type of program originally designed for logic task, and means-ends analysis is the essential component of it. GPS was developed out of *Logic Theorist*. More recent research has proved that the find-and-reduce-difference heuristic of GPS is a special case of more general processes.

Many specific topics had been marked in the course of uncovering the structure of IPS through many cases studies. Next section chooses Chess as the lens and demonstrates the issues that concerned Simon at different points of time.

4.2.3 Newell, Shaw and Simon (NSS) Chess Program and Onward

In this section, we take Chess, the 'drosophila' of AI and cognitive psychology, as the example to see how the IPSs of a specific domain are built from tackling different faces of the domain over decades. By wondering about the cognitive mechanisms that Chess players employ, Simon, Newell and Shaw had made the very first step on logic as a domain and later tackled many different aspects of HPS. Gradually, they have created different systems which take into account different hypotheses about HPS.

Selectivity

NSS is a Chess program which was developed by Newell, Shaw and Simon in 1958 after few attempts by others. It demonstrates the very first attempt to prune the search tree, both width-wise and depth-wise. It is a relatively sophisticated program in comparison with its precursors. It is sophisticated in the sense that it included the merits of earlier programs and considered more realistic characteristics - selectivity - of human decision making. In a nutshell, it imported the concept of "dead position" from Turing's primitive program (see Turing, 1953, for detailed definition) and "move generators" from Bernstein (1958). The two concepts are highly related to "selection" and "evaluation". Dead position[13] generally indicates those positions which are not worth exploring any further because it will not yield any immediate advantage or exchange. It is useful for pruning the search tree. Move generators are linked with the chess features respectively, and Bernstein limited the maximum number of moves for further evaluation to seven[14]. As a result, the same amount of resources can be used to evaluate each continuation more carefully. Besides, NSS program incorporates *satisficing* in making a move and the first move reaching the acceptance or aspiration level is executed.

Later, the behaviour of NSS program is compared to a subject's protocol. The verbal report is divided into episodes and shows several tendency i) the subject gives a general description of the board first, ii) The subject considers only a handful of possible moves, and iii) in most of the cases, the subject explores the game-tree straight down till some point and goes back to the initial position (base-move or base node) to start all over again. This kind of search method is called progressive deepening, which will be defined shortly.

Perception

Newell and Simon, motivated by the comparison and the qualitative distances between programs and the human subject, went on extending the analyses of problem solving of Chess to various directions. This line of research was initiated by de Groot (de Groot, 1965). De Groot's analysis of Chess players is prone to thrive the knowledge-oriented modelling for expertise and computer programs. De Groot's analyses imply that perceptual processes are the radical repertoire of skilled Chess player, especially the results of the tasks of reconstructing meaningful positions and arbitrary position support this claim. Namely, the most noticed and frequently mentioned result of de Groot is that experts outperform nearly perfectly the novices or beginners at reconstructing his-

[13]The famous alpha-beta procedure used in NSS applies the concept of dead position and *min-maxing* to obtain the static evaluations of the nodes which are the possible outcomes after applying the moves suggested by move generators. Notice that:

> "The alpha-beta procedure is not a heuristic. . . . Here, however, the problem is to know when a better value cannot be obtained by additional exploration, and this is indeed a heuristic matter"
>
> Newell and Simon (1972), p. 684

[14]Bernstein's program was compared with Los Alamos whose search is much less selective. Newell and Simon (1972) comments that "selectivity is a very powerful device and speed a very weak device for improving the performance of complex programs."(p. 678).

torical meaningful position after a glance for only 5 seconds. However, when it comes to arbitrary positions, the experts perform as equally badly as the beginners.

Despite the limitation of protocol analysis, Newell and Simon have confirmed that the subject's behaviour can be characterised as search in a problem space which is defined by the rules of Chess and a few abstract moves. It is also evident that the subject's search strategies match the *progressive deepening*. Progressive deepening introduced by de Groot and based on observation of subjects' protocol consists of four phases: the first phase of orientation, the phase of exploration, the phase of investigation and the phase of proof. It indicates that at different phases of thinking, various degrees of depth and breadth are required. At the first phase, the subjects identify few possibilities without going deeper. The search conducted in the later phases results in the traverse in the tree looking not bushy but reaching certain depth and restarting all over again from the same initial point or new variation. Depending on different needs and goals in the phases, one branch can be sought more than once with various depth.

Although NSS program and the subject's behaviour are consistent in their basic structures, none of the Chess programs by then encompassed all the characteristics of the subject. Newell and Simon had dedicated their attention to search mechanisms instead of perceptual and noticing mechanisms. However, they believed that the perceptions of Chess board are inherent in the sub-processes of search algorithms, such as MATER[15], whose sub-processes take the relations on the board into account.

Perceptions in Chess are highly visual activities. PERCEIVER (Simon and Barenfeld, 1969) is another specific program which simulates the local information gathered by using fixations observed from eye-movement. That is, when the eyes are fixated on a single piece or a square, its relation between neighbouring pieces and squares are obtained. However, PERCEIVER had not been designed to explain the Chess players' ability in retention that the players need to hold up the information observed from the position in the memory and reconstruct the whole position.

Considering the excellent performance of de Groot's grandmasters in reconstructing a position most perfectly after five-second exposure to the position, the reasonable explanation is that they hold up the information in STM and a position is decomposed into no more than 9 chunks. The premises of HPS is the existence of LTM and STM, STM's capacity is 7 ± 2 chunks, and it takes at least five seconds to write a chunk from STM to LTM. Therefore, a glance which lasts for only 5 seconds is not enough for the problem solver to utilize the storage in LTM.

Based on the associativity of LTM, the discrimination net of EPAM (see below) is believed to have facilitated the grandmasters to remember the whole position by a number of chunks in the STM. Any familiar chunk is recognized through the discrim-

[15]MATER (Baylor and Simon, 1966) is an end-game program that looks for a sequence of checking moves that lead to a mate and it has incorporated the scan-search strategy - a strategy that humans use in general. Although the scan-search strategy imposes limitation on the depth of search, it requires a much bigger capacity for storing the nodes visited in the memory. Provided that the capacity of STM is small, this strategy has to rely on LTM and write the information into LTM, which will take considerable amount of time.

ination process and is given a symbol that designates it; this symbol will be retrieved to STM. In the reconstruction phase, the master can reconstruct the chunks back to the board by using those symbols that designate those familiar patterns in the LTM. A grandmaster is able to recognize more pieces as one chunk than, say, a master does. That may explain why grandmasters perform better than masters or Chess club players in reconstruction provided that they have the same capacity in STM.

Elementary Perceiver and Memorizer (EPAM) EPAM is an IPS that was firstly designed for rote verbal learning and modelling the learning behaviour through association and performing behaviour according to what is learnt at the terminals of discrimination net. The construction of the net is supervised by macroprocesses (Feigenbaum and Simon, 1962) and microprocesses (Feigenbaum, 1959). Macroprocesses take the processing time limit into according and coordinate the order and use of time by microproccesses. Microprocesses are programs that deal with symbolic operations. Discrimination net is represented as tree-like structure whose intermediate nodes are tests for checking certain attributes of the stimuli and the terminal nodes are symbol structures that may contain the stimuli, responses, cues and templates. EPAM's performance in terms of recognising stimuli is essentially a sort of production system, however, the condition-action relation has been internalized into the discrimination net. More discussions about EPAM-like structures can be found in Feigenbaum and Simon (1984).

Accepting that the masters do perceive a position as few familiar chunks allowed Newell and Simon to connect the move generations from the features of a position to means-ends analysis of GPS. Chess indeed plays the role of a non-trivial and complex domain that brings together various programs that are designed for different tasks.

The studies on Chess perception progressed further over the years. The proposal for combining PERCEIVER and EPAM was implemented in Simon and Gilmartin (1973). Chase and Simon (1973b) conducts a new experimental techniques using time intervals to isolate the chunks that are recognized by the subjects from a position. Later the *chunking theory* for the skilled Chess players' cognitive process was developed in Chase and Simon (1973a). A series of extensions of memory tasks gave birth to the *template theory*[16] (Gobet and Simon, 1996) which revises chunking theory and addresses a template in the LTM without altering the assumption of capacity of STM. A post-Simon program, Chunk Hierarchy and REtrieval STructures (CHREST), developed by Fernand Gobet since 1992 is derived from EPAM (de Groot and Gobet, 1996). Model of Syntax Acquisition In Children (MOSAIC) is in turn an instance of CHREST.

The discrimination net of EPAM has been the core structure of knowledge representation of IPS since then. Its evolution relies heavily on the learning processes and the environment that the problem solver is confronted with. The actual architecture and content of a discrimination net vary according to the special cases depending on the time and circumstances. Each discrimination net is a database that reflects the prob-

[16]It turns out that the template theory is a relevant architecture for problem solving of Go which is discussed in next chapter.

lem solver's own experience and knowledge. Hayek (1945) argues that the knowledge of special cases owned by each individual cannot be easily accessible by an authorized central planer who refers to a body of knowledge that is the aggregate "scientific knowledge." Thus, he argues pure central planning can not be efficient. Hayek's argument suggests the importance of treating individual as having spacial-case knowledge, which is consistent to the nature of discrimination net, but there is also the difficulty in aggregating the individual bodies of knowledge to the whole body of existing knowledge. This issue is indeed deep and demands much more deliberation, which is beyond the scope of this thesis.

Pattern recognition and perception: A comment The magic behind the Chess grandmasters' ability is often pointed to them having at least 50000 familiar chunks in the memory. A relevant metaphor is the amount of vocabularies one should know for mastering one natural language. Newell and Simon had similar opinion:

> "The quantities demanded by this explanation are of the right order of magnitude. In the years required to attain mastership in chess, a player might be expected to acquire a "vocabulary" of familiar subpatterns comparable to the visual word-recognition vocabularies of persons able to read English, or Kanji (or Kanji-pair) recognition vocabularies of persons reading Chinese or Japanese. These vocabularies are of the order of $10^4 - 10^5$ symbols. Hence, sequences of seven such symbols could be used to encode $10^{28} - 10^{35}$ [i.e., $(10^4)^7 - (10^5)^7$] different total board positions"
> Newell and Simon (1972), p.781-782

The above quotation suggests that remembering patterns for chess player is like remembering vocabularies in English or Chinese. Experienced players know and remember more patterns or chunks than the beginners do. However, it is also obvious that placing vocabularies together will not necessarily make meaningful sentences. The amount of familiar chunks in the memory alone is not sufficient enough to explain expertise in general. Chess as a domain has the tendency to veil this aspect of necessity. Because these so called familiar chunks are themselves the conditions or actions which form the production systems. To our delight, the organization of the domain knowledge of Go experts reveals this property. This is the main argument of this chapter and detailed discussion can be found in later chapters.

4.3 Go versus Chess

Chess is both a recurrent example of Simon's argument and his main focus of HPS. The game of Go, which is more complicated than Chess in terms of complexity, however, is very different from Chess from many points of view. It is still reasonable to extend Simon's research on Chess to the game of Go as a small step, because Go still has an objectively well-defined problem space compared to other problems which we encounter in reality. This section is dedicated to bridge the domain of Chess to Go through an information-processing path among many other possibilities. We place importance on selective heuristics and those methods which do not engage in selectivity are irrelevant for HPS and therefore, ignored.

One of the reasons why Go is worth studying despite the success of Chess is related to the directions taken by contemporary Artificial Intelligence. I believe that AI has unfortunately been understood as merely concerned with "machine's intelligence." Any algorithm that disregards the human limitation of computational capacity will be considered as outside the scope of Herbert Simon's definition for AI, which is synonymous to information processing for Simon.

4.3.1 Artificial Intelligence

If there is an AI spectrum where "Brute-force" is at one end, and "Knowledge" at the other, then HPS should be located at a place very close to the end of Knowledge. It shows once again that our computational power is very severely restricted but we are very good at reasoning and accumulating knowledge. Any form of algorithm that tends to work towards the side of "Brute-force" is a deviation from human characteristics and thus contributes almost no insight to "intelligence", instead only to the "artificial". From this point of view, Go serves as a good example due its massive branching factor while searching a game tree. Brute-force algorithms were never under consideration in the field of Computer Go at the beginning since its vastness was beyond the reach of brute-force AI.

Taking advantage of the improvements in the speed of digital computers today, many competent algorithms rely on brute force search rather than strategic or selective search. Relying on the speed and power of digital computers is entirely justifiable if we program computers to assist our daily work, but an Artificial Intelligence that is divorced from Cognitive Psychology seems hardly just. Furthermore, if one day the computer Go in AI reaches a level of beating a human grandmaster, then that result should be a *consequence*, not the *aim* of information-processing AI.

Before modern supercomputers were born, it was clear to the Chess programmers that increasing the computational power of a physical device cannot reduce the complexity of the problem as effectively as selectivity. However, the charm of the speed of computer took over in 1990's. It is about the time when Kasparov was defeated by *Deep Blue*.The victory of *Deep Blue* over Chess champion Kasparov in 1997 is one of the indicators of triumph of computer Chess achieving enough expertise. Many believe that *Deep Blue* benefited from its specifically designed hardware, which facilitates the program to evaluate two million positions in one second. The same success of brute force has not been seen in the world of Computer Go till then. Two widely accepted reasons which might explain this phenomenon are (i) Go playing skill is knowledge-intensive and (ii) there is no appropriate evaluation function.

The ability and quality of HPS and machine problem solving are usually examined through competitions. Inevitably, people pay more attention to those programs and human experts that win the title or beat all other competitors. This diverts attention from the key characteristics which contribute to these triumphs and to the qualitative distance between human methods and machine methods, despite being the "strongest" program. It is both commendable and sad at the same time that Chess playing programs already have the ability to beat human champions. It is commendable because it is a big achievement for artificial intelligence. It is sad because people might be mislead to overlook the existing gap between humans and machines and think that Chess as a complex problem has been "solved." I want to emphasise that the game of Go is a good paradigm because it reminds us that such a gap does exist.

It should be clear by now that the research on problem solving is, in many cases, fulfilled by machines (algorithm), which consist of procedures that the researchers derive from their observations and deduction. However, these procedures of the machines may be completely different from those applied by the human problem solvers, whose methods of solving a problem can sometimes be imprecisely described or observed. Therefore, all the procedures discovered in artificial intelligence are in the form of approximations to different extent and the gap between the procedures of machines and those of human beings exist in principle.

Go is an important and fascinating paradigm for our future research not only because it is far more complex than Chess or any other problem, but because it is too complex and ill-structured from the point of view of mechanical brute-force search, which still plays an major role in AI. If we interview an expert of Go, we might find out that many problems that are computationally complex are not complex at all for him or her. In other words, the expert may be able to sense a good solution in a short period of time when a computer program needs to take a massive amount of resource to achieve the same result or worse. An intuitive answer to the question "why is there a huge gap between the performance of Go programs and human Go experts?" may be that the game is no longer so objectively complex to the experts. The huge computational complexity that a novice (or a preliminary programmer) suffers from has vanished or been transformed. Computer programs can not improve qualitatively if they do not find the right internal representations.

In Simon and Schaeffer (1992), a summary of the status of computer Chess associated with artificial intelligence is provided. The development of computer chess was divided into three eras - pioneering era (pre-1975), technology era, and algorithm era (post-1985). The similar remark on the three eras is made by David Erbach[17] in a report written after his participation of the *workshop on New directions in game-tree search*, in which many remarkable pioneers of computer chess participated. Among the conference papers, they were only two contributions on Go. Erbach showed his concern and regret on the rapidly increasing number of Chess programs:

> "Here is my summary of some of the points they made. Go does not yet suffer from most of these. In fact, by a sort of curious inversion, they show

[17]He was the editor of a quarterly journal (magazine) called *Computer Go* which functioned during 1986 winter and 1991 spring.

why computer go research is interesting. But the winds may blow the same way:

"Research in computer chess has progressed over the past three decades to the point where programs will soon be grandmaster strength. Superficially this accomplishment seems impressive ... However, three decades of progress have relegated the program of building chess programs to only a peripheral relationship with AI. Why has chess fallen from grace to the mainstream research community."

Computer chess may be said to have passed through three periods: the first *pioneering* ear . . . The second *technical* era was marked by the exploitation of the brute-force $\alpha - \beta$ search. . . . Now, a new *algorithm* era has begun with emphasis on new search methods, such as "selective deepening."

But "why haven't we tackle the real problems of knowledge representation, acquisition, and usage?". . .

There is no doubt these tournaments have been a tremendous boosts to computer chess. But how does one search for solution to difficult problems, if success may take many years, when one has an obligation to have an improved program each year?. . .

". . . There is lots of theory on search algorithm... However, there is no theory of chess knowledge and its interactions with the search. Chess programmers alter their program and have no means for understanding the potential consequences of change. Instead, they usually conduct experiments to see if the change is beneficial." "

Erbach (1989), p.9, italics in the original.

Chess as a domain has contributed to the success of game-tree search techniques, such as those approaches that benefit from $\alpha - \beta$ procedure. However, being a much younger domain, Go has resulted in more 'advanced' search approaches up to now. The competitive computer Go programs up to year 2000 all discard full-width brute force search and apply selective and heuristic search with the aids of goal-oriented pattern recognition, life-and-death analysis, etc.. They have reached the advanced kyu level. However, the contemporary computer go rarely takes the direction of HPS, instead they look for statistical sampling search - such as Monte Carlo. Optimization technique is also quite ubiquitous in Computer Go literature. "Crazystone" is the strongest program up to date. It has beaten Yoshio Ishida (9 dan)[18] on March, 20, 2013.

4.3.2 Decision Making in CBE and Board Games

We need to understand why board games are important paradigms in CBE to understand decision making. Then, we can appreciate why and what characteristics of Go make it a better candidate than chess. CBE views that decision making entities (no

[18]He is a Japanese professional Go player who has won many important titles

matter individuals or institutions) adapt to the external environment by manipulating or re-configuring their internal structures. Objective optimality may not make much sense as a criterion, because the systems in such decision making entities might have their functional criteria derived from different dimensions. A solution is considered as good, if it solves the current problem to a degree that is satisfying to all the faculties. For example, if we consider the human body as a system that contains many subsystems, then there is no 'best state' of the system. Because each system is in charge of different aspects of the functions of life, it is more likely that they have conflicts in their needs. However, a healthy body is the kind of system that keeps a good balance of the operation of each subsystems. The same structure reflects on individual decision making. A decision, or a problem always involves more than one dimension (attribute) of our lives. A good solution is a satisfying solution that adaptively takes into account all aspects of the needs.

Herbert Simon, with one foot in the theory and applications of computation and the other foot in theories and practice of HPS (where Chess played a crucial role) came to suggest a few economic domains to which procedural rationality can apply. They are "normative microeconomics", "theory of business cycle", and "Schumpeterian domain of long-term dynamics" (Simon, 1978b). He was confident enough to address that these three topics may not be the only and the most promising economic areas to be benefited by procedural rationality. We suggest that taking the game of Go as a fine extension from HPS is very likely to unveil more economic areas to the collection of this application and the like. Given the above discussion, we can summarize the aspects that concern Classical Behavioural Economics in the context of HPS.

1. Core premises: Satisficing, Heuristics, Bounded Rationality, IPS, HPS, Computational Complexity, Computability Theory. Finding 'satisficing' choices is the ultimate goal. There may be several sub-goals with in a problem.

2. Any issue that addresses severe computational limitations of the real computing entities, who face objectively complex problems and their abilities to solve the problem satisfactorily well are of concern for CBE.

3. CBE is interested in understanding the growth and evolution of organizations (organism) in complex environments.

4. It strives to comprehend the 'procedurally' rational processes that are involved in (organisational) decision making, both at individual[19] and institutional levels.

5. It seeks for a general theory of human decision making along the lines of an information-processing approach.

This thesis has taken an initiative on the last point above and suggests that Go is a promising domain for characterising a more general theory of HPS. First a all,

[19] It is possible to view an individual as an organization in terms of knowledge during problem solving.

using "games" as domains of inquiry is not unfamiliar to behavioural economists. The popular examples in behavioural economics are the notion of a lottery and some social games, e.g. dictator game, ultimatum game, trust games, to name a few.

Board games are shown to be comprehensive and representative subjects for investigating many different aspects of human decision making (Gobet et al., 2004). Computer Science, which provides the techniques to transform the 'processes' of decision making into executable algorithms serve as a platform to test and validate different cognitive models. A field in Cognitive Science believes that board games provide a solid and a rich platform to study human behaviour in the face of uncertainty. Chess is well studied and while Go is a new challenge even within cognitive science.

An important message of this chapter is that knowledge plays a central role in complex problem solving. Chess expertise, for example, involves the perception of meaningful chunks that suggest good candidate moves. Therefore, domain-specific knowledge has gained a lot of attention for constructing expert systems. It seems to suggest that the expert system aimed for specific task environments should be build independently based on the domain-specific knowledge. Nonetheless, human beings as general-purpose thinking and reasoning organisms can potentially be all kinds of experts. A human being can be educated with medical knowledge for ten years and become a practitioner, likewise, for artists, mathematicians, and lawyers, etc.[20]. Although, there is no doubt that there exist individual differences in talent, human beings' potential strongly induces queries such as "what are the general characteristics or architectures underpinning knowledge acquisition", "what is the general structure of knowledge?" "how does new knowledge emerge from existing knowledge", "to what extent our knowledge can be represented?". "If the concepts are not innate, can we really begin with nothing?"

The game of Go as an increasingly important example in artificial intelligence, which has as least two advantages over other problems; first, the representation of the game in the program is fairly simple, because the game is played by identical stones with black and white colors. Second, the structure of the game is general enough so that its problem solving architecture can be applied to more problems in the future (Reitman et al., 1974).

A domain like Chess has helped us unveil many aspects of human thinking and problem solving activities. It has somewhat reached the limitation that is inherent in the nature of the game. The game of Go, on one hand, still belongs to the category of board games where we can confidently claim that the subject is fully working within the problem space confined by the its task environment when he/she is solving the Go problem. On the other hand, it is a qualitatively different from Chess and provides more insights to HPS as we will see below.

The distance between Chess and Go is evident, in particular, Burmeister and Wiles (1995) has provided an illuminating summary and listed out 12 features by which Chess and Go can be differentiated. Among the distinctions, the most remarkable ones

[20]Personally I think that philosophers do not fit for this law, on the contrary, too much training might kill a philosopher!

are that i) in computer Go there is still no good enough evaluation function like the ones in Chess. This property is the main obstacle in computer Go when defeating human grandmasters or other computer Go programs. ii) Go games do not terminate with absolutely defined configurations, instead it requires certain levels of mutual understanding between the players. There are several other interesting and important distinctions can be added to their list, Some of them are peripheral observations and some of them lead to deeper discussions in the next chapter:

1. In Chess, the player are provided with a team of army and a small battlefield; In Go, the players are provided with a big piece of land and identical raw materials.

2. The operator in Go is extremely simple, the complexity should result from the meta-level organisation.

3. Lookahead might be more straightforward in Go then in Chess. Because Go requires the players simply to add stones to the board, but in Chess, the players have to move the icon, not only what to move, but also where to move.

4. History is much more important in Go then it is in Chess. Thus, static analysis of a given board configuration cannot always contribute to selecting a good move.

5. Game tree may not be a good representation of Go any more.

6. "Life and death" problem is not so crucial in Chess.

7. In Go, players also need to calculate and care about the number of points they can potentially get to a very precise level, although this scheme happens mostly in endgame. One or two points here and there can flip the result of the game eventually.

8. Board coordinate is less important in Go than it is in Chess. Because the Go board is highly symmetrical.

9. The counterparts of 'chunks' in Go are called 'Go terms', both of which however are qualitatively very different. Chunks consists of definite icons and relations among them. In contrast, patterns recognized by Go players are categorized with the aid of definitions labelled by a given 'Go term'. A Go-term may not refer to a single or definite configuration (like a chunk in chess). Instead, it refers to a membership criterion in which many qualitatively similar patterns are grouped using single definition (a Go-term)

The research questions asked in this chapter result from the fact that there is a delay of at least 2 decades of achievement in computer Go in contrast to the one of Chess. There were attempts at programming Go with the intention of dissecting the structure of the game itself, such as the programs of Zobrist (1969) and Reitman et al. (1974). The common drawback of these programs is that they all ignore the physical limitations of human beings. It is very tempting to utilize the speed and space that digital computer is now equipped with, but then it is in some sense harmful if we aim to understand how human beings can be selective in search and perform very well as an expert.

There are at least two ways to tackle and understand the distance between Chess and Go. One is to simply accept that Go is an extremely complex problem and develop more efficient and powerful "artificial" programs to solve it. Another one is to take a few steps back and wonder the mystery underlying human cognition while playing the game of Go and obtain insights for potential algorithmic approaches to a build a strong Go playing program. The second direction matches with the spirit of HPS. Many attempts have been made on the second direction, such as Saito and Yoshikawa (1995), Saito and Yoshikawa (1996), Yoshikawa et al. (1999), Saito and Yoshikawa (2000), Burmeister (2000), Burmeister et al. (2000), and Gobet et al. (2004). These research attempts guide the postulates of the next chapter.

I conjecture that Go is a more general paradigm than Chess, in other words a superset[21] of Chess. This specifically relates to knowledge representation structures, where the chunks in Chess are merely special cases when compared to Go terms. The problem space is far more complex in Go. The reasoning and hence the production systems in Go are way more intricate than Chess. The terminal configurations in Go are not as well defined as in Chess and therefore, providing a more general setting to investigate strategies, decision procedures, evolution of goals, stopping rules, tacit knowledge etc. In particular, the generality of Go in comparison with Chess is demonstrated by a more general architecture of IPS of playing Go in thesis (see section 5.4).

4.4 Concluding Remarks

In orthodox economics and modern behavioural economics, an individual decision maker is often modelled as a mathematician solving optimization problems; in classical behavioural economics, where algorithms play a central role, the decision maker is modelled as an IPS. Unfortunately, the latter is often misperceived and dismissed as a bloodless machine. We reiterate that IPS is actually a general representation of the mental architecture of a human problem solver.

CBE adopts theory of HPS, which has been proved to be one of the fruitful approaches to understand the human heuristics and procedures in decision making. The gap between the *realistic* procedures and the *procedures in theory*, in terms of problem solving, stem from the discrepancy between *true internal representations* of a problem and those internal representations that are *assumed* by the researchers.

In CBE, decision making is multidimensional in general. It is tied to different goals and needs of decision making entities. A preference ordering over a set of alternatives can never be 'effectively' generated in general. A decision is an output of

[21] The idea of a superset is still vague so far, because it can be understood through different media, such as mathematical properties, complexity and educational motivations, etc..

structured procedures that take into account all the faculties involved. Satisficing is not sub-optimality, but merely an action which ensures a good balance among the aspirations from different faculties.

In Go the power of brute force computing is very insignificant, therefore, it provides the genuine test bed for AI. This chapter attempted to show that the game of Go is a rich domain for understanding decision making in CBE, cognitive psychology and computer science, the three fields which are intrinsically connected by HPS. It provides a different point of view about this game and concludes that Go serves a important domain for CBE in terms of organisational decision making (it is a quintessential case that matches the manifesto of Simon (1944, 1947, 1979, 1991a)), for cognitive psychology in terms of memory structure and knowledge acquisition, and for computer science in terms of new criteria for knowledge representation. One of the main messages of this chapter that will be relevant for the next chapter is: "Chunking theories were developed out of studies of Chess; studies of Go can contribute to different kind of knowledge representation."

Chapter 5

The Game of Go: An IPS Interpretation

5.1 Overview

In this chapter, the game of Go and the abilities and behaviour of Go players are investigated. In addition to the selectivity that is universal and a recurring theme in human problem solving, Go players rely heavily on the use of Go terminology. With the aid of evidences obtained from various Go documentations[1], a pseudo Information Processing System (COMPOSER) is constructed to explain the problem solving aspect of Go. In particular, COMPOSER highlights and encompasses the verbal processing aspect of problem solving in Go.

In the previous chapter, we have seen the development of Chess playing programs where the chunks in Chess knowledge play a critical role in explaining Chess expertise. Throughout this chapter, the readers' attention it to be drawn to the observation that almost all labelled chunks in Chess have definite number of items, members, and relations among the members. That is to say, a program can reproduce exactly the same pattern when it is given a name (symbol) of that chunk. In other words, there is an one-to-one correspondence between the label of a chunk and the pattern of a chunk.

However, many Go terminologies refer to definitions on membership or *concept*[2] rather than a single and specific configuration. That is, many patterns can be referred to the same term, and the program may not be able to display all the cases that belong to it. An immediate example is the Check Mate in Chess that it works on definition. It is a "many-to-one" relation.

The main argument of this chapter extends from the above observation and some investigation into the problem: *The problem spaces of Go are semantically richer than the ones of Chess*[3]. This statement follows from the research on Go based on problem solv-

[1]Among the various documentaries discussed in this chapter, the verbal protocol that comes from Simon's tradition of protocol analysis plays the most important role.

[2]By concept, our understanding of it coincides with its definition Simon and Kotovsky (in 1963, p.534), "a *concept* is taken to mean a subclass of some class of objects, or, alternatively, a procedure for identifying a particular object as belonging to, or not belonging to, such a subclass."

[3]Recall that the problem space should be distinguished from task environment, see previous chapter

ing tasks and protocol analyses and closer observation of Go terminologies. Go players' reasoning operates not only with specific patterns but also the meanings of a configuration, thus some of the intermediate responses produced in the course of problem solving are the output of the higher-level production systems. If we are allowed to expand this interpretation to other task environments, then our conjecture is that more ill-structured a task environment is, more semantic the problem space might be.

The fundamental difference between Go and Chess emphasized by this study is that the explicit rules of the game of Go is extremely simplistic and the most elementary units of Go are the characterless and identical stones. However, massive number of implicit rules have emerged from the accumulated Go knowledge that is encapsulated by the Go terms. Without the understanding these implicit rules and their interrelationships that are known to the human players, it is impossible to conduct effective evaluation procedures like those successful cases in Chess.

The general properties of the game of Go that are summarized in this chapter match those of many real-life problem domains that concern CBE. A decision making process involves a number of faculties, which are in turn driven by different regulations and even interactions with other faculties. If we take these implicit rules into account, the regularity of Go is much more complex than other domains than it may appear at first. A better player can understand and manipulate higher level rules (concepts) that are not comprehended thoroughly by the intermediate players. Therefore, this chapter suggests that the beauty of Go lies not in the search algorithms alone, but in the organization of Go terms and inferences that can derive from them.

Section 5.2 introduces the game of Go, its mythical origin, rules, and its relevance to other problems. Section 5.3 provides a more detailed analysis of the problem spaces of Go in the perspective of Go learners and players. Section 5.4 brings together the evidences of problem solving of Go and suggests a theoretically feasible pseudo IPS that explains the behaviour of Go problem solvers.

5.2 Game of Go

The game of Go is believed to be roughly 4,000 years old, making it the most ancient board game in the world and it seems to have originated in China. There is no consensus on its precise origin. It was a game played by aristocrats and royals. The contemporary version of the game is played worldwide on a 19 by 19 board. There was an evolution of board sizes and from archaeological evidences, board size with 15 by 15, 17 by 17 and 19 by 19 were found at different sites.

This game appeared in classical Chinese literature and folk stories, and some of them are mythical and exaggerated. For example, some ancient Go veterans cared about their game and dedicated themselves so much so that when they were about to lose a game they vomited blood and died. Although Go originated in China, it later developed and was preserved in Japan. Even in Japan, there were several aristocratic 'Go families', in which the title of the Go master goes only to their sons. Yasunari

Kawabata, a winner of a Nobel prize in literature, has written a story which records the historical match that ended the 300-year old of aristocratic tradition (Honinmbo). This match lasted for almost 6 months.

A game of Go, played on a relatively big board with very elementary rules and identical stones, can last for days or months if there is no time limit specified. A complete game if often roughly divided into three phases: Opening, mid-game and end games. The target in the opening stage is to get a good span on the board, because the arrangement on the board can be used to develop future solid territory. At this stage, fierce short distance battles are rare. At the mid-game, there are many local life-and-death and killing battles and the boundaries between territories are still vague. In the endgame, the players are usually competing for one point because the rest of the territories are all settled. At this stage, players usually have to look for the "optimal play". The most important feature which distinguishes Go from Chess is that the stones are added to the board without being removed (except for being captured) one by one. Thus, killing (capture) such as the exchange in Chess is never the main goal of the game. When a game is finished, there are plenty of stones remaining on the board; on the other hand, a Chess game may end with only few pieces on the board. The ultimate skill of playing Go is the ability to balance the whole board and detect the weakness of the opponent.

5.2.1 Go Rules

One of the attractions of Go is that it is possible for a novice to learn the rules and to recognize those important rule-concerning patterns in a few minutes, however, facing a rather empty board, a novice might not have a clue on what to do.

A Go board has 19 vertical lines and 19 horizontal lines which intersect 361 points. There are 9 small black dots on the board, and they are called the star points (See figure 5.1). The game is played by two people, who sit face to face with the board in the middle. One player possesses 181 black stones, and the other holds 180 white stones. The total number of stones is exactly the number of grids on the board. The stones are initially stored in the containers. Conventionally, the weaker player takes Black and the stronger players takes White. The game terminates after both two players pass in succession or by resignation. The player who gets a higher score by surrounding more territories wins the game.

The Black places the very first stone on the board, and every point of the grid is possible. White places the second stone. They continue to place one stone at a time alternatively on an unoccupied intersection. Every vacant point is a legal move except for two cases: *suicide* and *Ko*[4]. A player commits *suicide* in the game when the placement of a stone causes the exhaustion of liberties of his/her own chain of stones. *Suicide* is not allowed, because a chain of stones which has no liberty will be captured

[4] In this chapter, all the Go terms will be italicized and their meanings are either explained in the text or collected in the Appendix A in the alphabetical order.

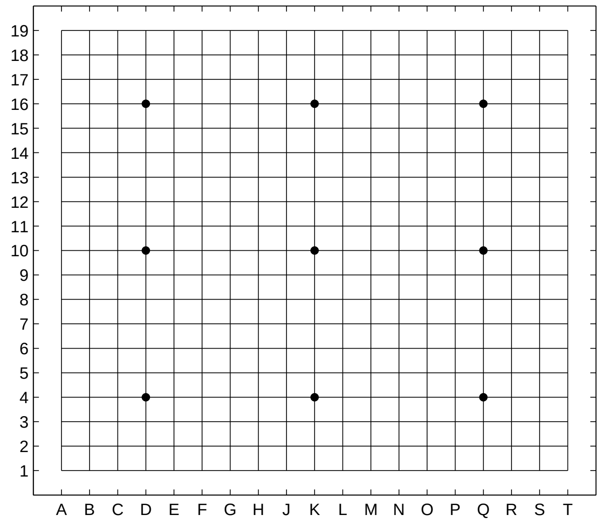

Figure 5.1: An Empty Go Board

and become the prisoners of the opponent. The idea of liberty and chain need to be defined later. Ko is a situation of (infinitely) alternating capture of opponent's single stone that causes repeating loops of the board configuration. The possibility of infinite loops resulting from this kind of capturing is excluded by Go rules. Ko situations will be demonstrated later.

Chains, Liberties and Go

A chain is a unit of stones that are either vertically or horizontally adjacent to each other. By definition, a single stone is a chain of one stone. The Go rules can be more illustratively explained by the aid of the small diagrams within figure 5.2. Diagram *a* shows that a chain of two black stones which are horizontally connected, on the other hand, Diagram *b* shows the same case for vertical connection. Diagram *c* is an example of a chain of 3 stones. Diagram *d* is a counterexample; this is not a chain of two stones, but two unconnected stones. Diagram *e* is an example of a chain of 8 white stones.

Liberty After knowing the definition of a chain, it is then straightforward to observe the liberties of a chain. Diagram f shows the liberties of a single stone. The liberties of the stone are marked with the star signs. By definition, the liberties of a stone are the unoccupied intersections immediately adjacent to the stones through the ways (the lines). They are called liberties in English, because these are the ways through which the a chain of stones can be extended. As it is shown in the Diagram f, a single stone which is located at the interior part of a board has at most 4 liberties. Diagram g shows a chain of 3 stones and its 8 liberties.

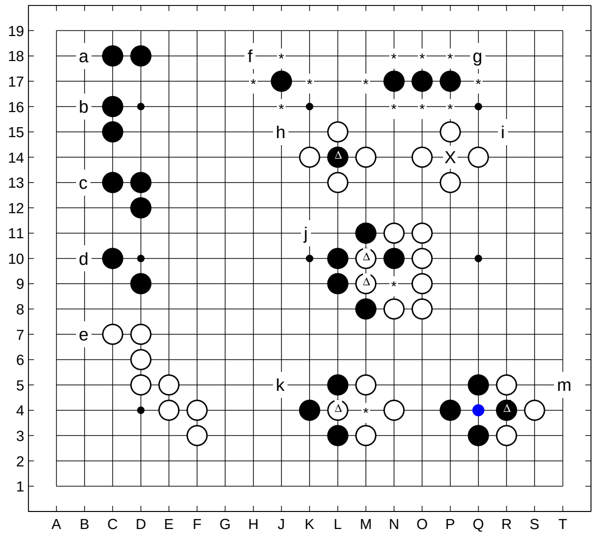

Figure 5.2: Demonstration of the Go Rules

Capture In Diagram *h*, 4 liberties of the single black stones are occupied by 4 white stones, therefore, the black stone has run out of liberties and is about to be captured by White. Diagram *i* shows the situation after the black stone is captured. Black can NOT place a stone to the $\times$ position any more. By doing so, the black stone will be captured as soon as it is place on that spot; this is suicide and is prohibited. However, there is an exception when suicide is allowed, that is the placement exhausts the liberties of opponent's chain even though that position is also the last liberty of own stones. Diagram *j* shows an example of this case. It seems that Black cannot place a stone at the $\star$ point, because it is a suicide. However, by doing so, the two white stones with the Δ mark have also run out of their liberties, therefore are captured. In this case, Black is allowed to place a stone at the $\star$ point to capture 2 white stones. If it is White's turn, then White can also place a stone at the $\star$ point to capture the single black stone.

Ko Diagram *k* is a standard shape of Ko. Black can place a stone at the $\star$ point and capture the single white stone, and the result is in Diagram *m*. In Diagram *m*, if While places a stone immediately at the blue dot, the configuration goes back to Diagram *k*. By doing so alternatively, the board enters an infinite loop of Diagram *k* and *m*. To prevent this from happening, the rule about Ko is that after the board changes from Diagram *k* to *m* for the first time, the White is prohibited to recapture the black stone immediately, he/she has to place the white stone somewhere else, and then it will be the Black's turn again. After Black plays at the blue dot to win/terminate the Ko or elsewhere, White is now allowed to recapture the black stone and continue the Ko fight if the blue dot is not filled.

Other regulations

The above discussion has explained the Go rules. Note that, the stones, once placed on the board, will not be removed, except for being captured. After the game is finished (either by passes from both sides in succession or resignation), the scores will be calculated to decide the winner.

Score Count Score counting is not very straightforward, and the beginners are not expected to perform it flawlessly. It demands the player to identify the territories which are composed by a hierarchy of stones from the rather complex configurations and the neutral points (*dame*). The hierarchies of the Go stones on the board can be classified into: stone, string (chain), group, territory, and the whole board. Up to strings, they are clearly defined, and groups that are alive are a potential territory. However, the definition of the groups is rather vague and sometimes they are overlapping.

There are two systems of calculating scores: Chinese counting and Japanese counting. In Chinese counting, both the own stones on the board and the vacant area surrounded are considered as a final territory and an imprisoned stone is worth two points. In Japanese counting, only the vacant areas on the board are final territory and an imprisoned stone is worth one point. Based on the definitions of these two methods, there are respective counting heuristic procedures applied by the players. The essential result from calculating is the difference in scoring of the two sides. Therefore, despite the fact that these two methods are slightly different, the consequences (the winner) are the same.

Komi and Handicap Black plays first in Go, so it is believed that Black receives some privilege. In order to compensate White, 5-8 points (commonly it is 5.5 or 6.5 to prevent the draw) will be given to White at the end of the game. This rule is rather modern, it was formally introduced in the first half of 20th century. *Komi* is the compensation for White being the second hand. There is another compensation called *handicap* for the differences in players' strength. The number of handicap stones can vary from one to nine, and they are placed on the star points traditionally. Normally the number of stones indicates the difference in the player's rank.

Ranking System The ranking in Go begins with 30 *kyu*; and as the player improves, it decreases one *kyu* at a time to 1 *kyu*. After *kyu* stage, it comes amateur *dan* which ranges from 1 to 7 dan (with abbreviation *d*). The successive stage is professional *dan* which ranges from 1 to 9 dan (with abbreviation p).

5.2.2 The Philosophical Reflections of Go

> "One of the most extraordinary aspects of the game of go is that it has been proven that in order to win, you must live, but you must also allow the other player to live. Players who are too greedy will lose: it is a subtle

game of equilibrium, where you have to get ahead without crushing the other player. In the end, life and death are only the consequences of how well or how poorly you have made your construction."
- Go proverb, author unknown.

The game of Go is essentially a game of territory, not a game of killing. Of course, some local attacking and battles are inevitable in a game. The ultimate target is to survive in ways better than the opponent. The kind of subtlety in Go can also be seen as being analogous to many real-life problems

Its Counterparts

The nature of the game of Go has been metaphorically linked with many different kind of conflicts, dilemma and problems in real situations. These situations include warfare, politics and international relations; between two players or business competitions between two rivals. Due to the nature and ultimate goal of the game - occupying larger territory - it provides more freedom for high-level reasoning than the pure killing tactics do.

The battle between black and white by using their seemingly powerless and identical stones has given rise to different metaphors and semantics of the Go board. These metaphors, in turn, play important roles in the analysis of the games. Moriarty (1996) suggests that there are three semiosis of the Go board: warfare[5], light and dark, and life. Let me bring together some other metaphors to this collection: drops of oil and water, electronic charge (Zobrist, 1969). Different ways of perception will lead to different internal representations and in turn affect the way of reasoning[6].

Understanding Go via Go Seigen

One can get many interesting and deep perspectives about this game in real life through reading the autobiography of Go Seigen (Wu Qingyuan)[7], who is considered as the best Go player in the 20th century. The autobiography is basically organized in chronological order and with plenty of his mentors, friends, competitors, etc., as sections.

Go Seigen, who was born in China and whose talent for Go was discovered by a Japanese, was one of the three pupils of Kensaku Segoe. Lee Chang-ho (a South Korean player), 60 years younger than Wu, was considered as the second best Go player. However, they play with very different styles, that is, Lee is a conservative and Wu was innovative. One of the reasons for this difference might be that at Go Seigen's era,

[5] Boorman (1969) interpreted Mao's revolution by using the game of Go.

[6] My hunch is that at different stages of the game, different metaphors are applied. If we are interested in building a machine which is capable of learning to play good Go, then we cannot ignore how a good player is trained.

[7] Go Seigen was also the person who invented the 9 by 9 board with the intention to broadly promote the game of Go to Western societies. He realized that women's opinion in western families is quite dominating, therefore, it is better to attract females' attention first.

there was no regulation of *Komi*; therefore White who has the disadvantage of playing after Black has to fight more aggressively than maintaining the balance on the board.

Go Seigen and Mironu Kitani were the best rivals and friends; the first game played by Wu and Kitani was a mimic Go[8] which upset Kitani a lot, but according to Wu, that was the only way to beat Kitani. Wu started the game by placing the stone on the *tengen* (the very center of the board) and began the mimic Go, i.e. playing on the symmetric point where White has just played. Wu lost by two points. After this match, they became friends and invented new *fuseki* that focuses the central territory together.

From Go Seigen's own lifelong study of Go, he concludes that the game should be viewed as a whole, rather than as a combination of different openings or styles. A good move for him is the move that balances all parts on the board, just like the balance of Ying and Yang[9].

5.3 Problem Solving of Go

Herbert Simon never aimed to create an ambitious model, which can comprehensively explain every characteristics of human thinking. Here, my aim is to provide an initiative of problem solving for Go, along the direction that Simon has set up. Despite abundant studies in the literature of computer Go, the application of *satisficing* as the principle to mimic human selective search is rather rare. However, it is always important to start somewhere. My target at the stage of this thesis is to construct a general architecture capable of playing Go, which allows individual differences to be its detailed content and learning across time (adaptive behaviour).

The study of problem solving of Go is not new, especially it is a popular domain for computer science, combinatorial game theory and cognitive psychology. There are competitive commercial programs of Go and studies of Go players' behaviour, but so far there are no concrete algorithms of Go which are developed out of the premises of IPS. This section focuses on understanding the possible problem spaces and the acquisition of expertise of Go, with occasional reference to Chess which is one of its precursor in HPS.

In the real Go games, the only external memory that the players have is the Go board itself. Unlike Chess, Go players can spend as much time as they need[10] for a move, therefore, sometimes a play can last for days. I'm motivated to uncover the mental processes which go on in the head of the players when they spend a lot of time staring at the board.

As for a tentative remark, individual players differ in the ways in which they construct the problem spaces given the same task environment. As we will see soon in

[8] This involves the player making exactly symmetrical moves of the opponent.

[9] From this point of view it seems to be formally tied to Satisfiability problem, in which the objective is to find the solution which satisfies all the sub-goals of the game.

[10] At least in the old Japanese tradition.

this chapter, the problem spaces that the subjects construct differ in the quantity and the quality of the semantics.

5.3.1 Problems and Task Environment

There are at least two kinds of task environments in standard games of Go. One is to decide to make a single move by looking at a given configuration on the board[11]. The other is to start the game from an empty board and play alternatively one stone at a time with the opponent.

When a subject faces the first kind of task, there are pros and corns in contrast to a real game. On one hand, the subject has no definite idea about the sequence of game which results in the configuration presented; on the other hand, the subject does not need to face the consequence of his/her solution[12]. When a player is given a static configuration, then the player should spend some time and effort to have an overview about the configuration and decide the sub-goal of this move[13] (the "search widely scene" introduced on p.737 or "initial orienting behavior" on p.776 in Newell and Simon (1972)). The amount of computation spent on overall search varies according to tasks and also the strength of the players.

The second type of task environment is exactly the same as how a standard game is conducted. A normally executed game lasts for more than 200 moves, as a result, the players have to plan more carefully and activate all the faculties that take care of different aspects of a game. The player is facing the consequence of each move thus he/she has to inspect the flow of his own thought and the opponent's purpose and make successive moves. Although the players might interpret the board configurations differently over time, these changes often result from the new input, i.e, the latest move by the opponent. The players may focus their attention more precisely on particular goals or regions without much orientation and in turn generate a few candidate moves. In this kind of task, a procedure for reasoning and playing 200+ moves is certainly not equivalent to an exact sum of each single move. This is different from the first type of task environment in which it is possible to carry out static evaluations and generate a move from taking current board configuration as the input. Here, a holistic perspective and history of the previous moves matter. It is unlikely that a player can generate a good move, simply by looking at the current configuration of the board, without a knowledge of the different battles and aims that characterized the game thus far. This directly relates to the inability of devising robust evaluation functions for a static configuration. In this sense, the game of Go is more dynamic. It can be more or less in

[11]It is exactly like the task faced by S2 in Chapter 12 of Newell and Simon (1972) and de Groot's subjects.

[12]One exception to this statement is found in those problems set designed for educating Go learners. Such problems may appear as soon as a new concept is taught. However, most of the problem sets are hinted with the sub-goals and are provided with possible answers. Therefore, the trainee can know whether his/her solution is right or wrong as soon as he/she answers.

[13]Because, anyway the subject cannot win the game with only one move!

different cases. Sometimes there is only one candidate for a good move, and sometimes a move is forced.

I believe that a good player has the ability to solve both two kinds of task effectively, because this is how they are trained and practice. Likewise, a reasonable Go program should differentiate these two kinds of task environments; that is, whether it is in a course of a game or whether it is given a configuration as input and is expected to generate a good move as output[14].

5.3.2 Heuristics: evidences from Go literate and documentations

In the history of Go, there have been many different themes of play which were popular in their respective eras. A theme is related to a certain pattern or strategy that should be used at a certain stage of the game. These themes are claimed to have been invented by famous players and largely discussed by media press and veterans alone in the initial stages. Gradually the themes will appear in the analyses of commentaries and later be included into pedagogical textbooks.

Due to the time and space limitations, the research summarized in this section is highly selective. In particular, the discussion here is in favour of knowledge representation, therefore, the most competent approaches in computer Go society - Monte Carlo sampling - is intentionally omitted due to its irrelevance to human cognition and procedures. Only those programs that conduct selective search are considered and likely to be relevant.

Instead of dissecting the problem structure and taking perspectives from a outsider or an observer's point of view, I am more interested in how real Go players perceive this game. This section presents evidence from protocol analysis of Go players pioneered in 90s and the Go documents which assist Go learners' and veterans' development and understanding of this game.

Early Go Programs

The earliest attempts to investigate Go players' perceptions, in the context of artificial intelligence, are Zobrist (1969); Reitman et al. (1974); Reitman and Wilcox (1975); Reitman (1976). From the early state of development of Computer Go, the importance of perception, knowledge and coordination of a good play were amply recognized. The definition for knowledge in Go was largely influenced by the notion of *chunks* in Chess.

Zobrist (1969) emphasizes the importance of the ability to recognize meaningful patterns of good Go players. Human players visualize a static board and organize the whole board configuration into their own internal representations. Zobrist's program has the following features:

[14]This problem is less critical in Chess, because there are good static evaluation processes. Therefore, a Chess playing program can be thought of as a composite of repeated processes of input-output or stimuli-response procedures.

- Grouping of stones in terms of local and global perspectives
- Viewing mental picture as internal representations
- When the program reads a board, it generates 7 internal representations as 7 matrices. Each internal representation concerns different attribute of the game, such as segmentation of a board, the domains of influence of the stones and the armies of stones.
- There are many built-in templates (chunks) which are n-tuple stones, which contain the relative location of the stones in the template and the respective references. The references are characterized as the combination of numerical conditions based on those internal representations.
- The program runs through all the templates which are associated with the internal representations and simply selects the position which has the maximum weight. However, how templates assign the weights to the particular positions when the templates are recognized on the board is rather unclear.
- This program is not able to calculate score when the game ends.

This program may look complicated enough, but its characteristics are remote from the human ones. To mention a few, human players are unlikely to evaluate a given configuration on a board by a linear combination of different attributes and criteria.

The programs came after Zobrist's have a different organization. That is, there are faculties from different hierarchy in charge of different tasks of the game. This kind of improvements did contribute to better strength of the programs. The next question is whether human experts also possess such hierarchical structures in their mind for playing Go.

Reitman (1976) replicated de Groot and Simon and Chase's method for finding the structural difference between Go expert and Go novice. The results of different treatments suggest that the chunks stored in the memory of experts are not well nested and very often overlap with each other. This paper is often cited by later papers as a disputation of chunking theory proposed by Chase and Simon (1973a,b). This result revealed the very sign of qualitative difference between Chess and Go in the context of human problem solving.

Cognitive and Protocol Analysis in Go

> "The results of the experimental studies indicated that inference impacts to some extent on memory performance, that memory for sequences of moves is related to Go skill, and that Go is a better domain than chess for investigating memory for sequences"
>
> Burmeister (2000), p. iv

The content of this section is a summary of the work by Saito and Yoshikawa (1995), Saito and Yoshikawa (1996), Yoshikawa et al. (1999), Saito and Yoshikawa (2000), Burmeister (2000), and Burmeister et al. (2000). They are among the very few whose work digs into the Go players' cognitive mechanisms and performance. In general, according to the empirical evidence of Go players' thinking processes, sufficient amount of inferences are involved. This discovery is a counterexample to de Groot's assessment that (Chess) masters' ability is merely an outcome of perception and pattern recognition.

Yoshikawa et al. (1999), Burmeister and Wiles (1995) and Burmeister (2000) recognized the dominating role of language that is used to assist sequential memory of Go players. It is surprising to know that even a static configuration can trigger dynamic perception, note that "human players recalled the board situation in the sequence order even when they only observed a static pattern."(p.297, Yoshikawa et al. (1999)). Language or terms used by the Go players connects a sequence of play into meaningful stories, which reduce the size of the search space dramatically. This research scheme can also be traced back to de Groot's analysis and Chase and Simon's chunking theory. Burmeister has concluded that inference in problem solving is applied more evidently by Go players than Chess players.

Iceberg model, which is the source of inspiration for this chapter is proposed by Yoshikawa et al. (1999) to interpret the advanced players' ability to communicate with each other by using highly abstract Go terms. The authors claim that Iceberg model is related to the template theory[15] proposed in Gobet and Simon (1996), but there is a difference.

The key phrase of the 'iceberg model' is that "Each Go term is an iceberg." (op. cit. p.295). Metaphorically speaking, the huge part of the iceberg which is under the surface of sea that is unspoken and unrevealed has the capacity to contain many slots of emergent meanings of the Go term. Yoshikawa, Kojima, and Saito think that the slots in the iceberg, which tie to each Go term, are defined dynamically and vaguely depending to the situation on the board. However, it is said that the slots in template theory are predefined. The template theory is proposed to extend Chase and Simon's chunking theory with new evidence collected from memory tasks of Chess players. The template hypotheses are proposed for keeping the capacity of Short Term Memory assumed by Chase and Simon intact.

To complement the chapter's emphasis on the mysteries behind Go terms, it is useful to have some examples of Go terms. The most simplistic Go terms among a huge glossary are positional terms, which describe the relations between two stones from the same color that are in a small neighbourhood. Of course, this bunch of locations can be mirrored to four or eight different directions keeping the reference stone at the center. Figure 5.3 indicates these spatial or positional Go terms. These terms which appear frequently in the Go textbooks and commentaries are associated with different aspects of Go tactics and higher-level concepts. They are like the most basic elements which

[15] Templates in the cognitive models have their correspondences in AI, such as 'frames' in Minsky (1975), which is later developed into Frame Representation Language by Roberts and Goldstein (1977).

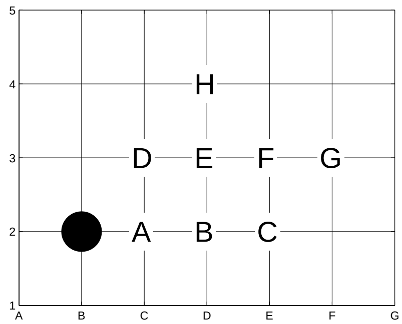

Figure 5.3: Local Spatial Go Terms that describe the relation between two stones of the same color. A=*nobi*(stretch); B=*ikken-tobi*(one-point jump); C=*nikken-tobi*(two-point jump); D=*kosumi*(diagonal); E=*keima*(knight's move); F=*ogeima*(large knight's move); G=*daidai-geima*(very large knight's move); H=*hazama-tobi*(diagonal jump)

go into different levels of Go concepts. Each Go posture has its own dual meanings in defence and attack, however, those higher-level meanings can only be realized with a series of follow up and elaboration.

It should be illuminating at this point to systematically bind together the facts and evidences collected from cognitive observation of Go players by Saito, Yoshikawa and Burmeister. I summarize the evidence of subjects' behaviour on Go related tasks based on novice/expert comparison and general tendencies. It should be noted that although the distinctions made in table 5.1 is binary and simplified, the evidences collected suggest a smoother spectrum on the variable of individual strength. The Go players' abilities and characteristics are in many ways consistent with de Groot's subjects. However, the use of language and Go terms in players' thinking processes is too profound to be ignored. The rationale behind the Go terms should be that naming a definition or a situation can reduce the time of describing it all over again. Similar to Chess players, Go players' cognitive processes also travel through several states (episodes), such as candidate move generation, lookahead[16], evaluation. However, Go protocols of advanced players are richer in Go terms and show the ability of memorizing the sequences of the games. The second part of the summary after table 5.1 is mainly based on the behaviour and reports of experienced players.

[16]According to my intuition, looking ahead by imagining stones being played on the board in succession seems to be easier than the cases in Chess. Because in Go, the stones are placed and will not be removed or exchanged, except for being captured. Therefore the imagined sequence of play should be properly achieved with the aid of external memory, the board.

Attributes	Experts	Novices
Board Description	Using Go terms	Using their own words
Size of Protocol	Big	Small
Knowledge of Go terms	More	Less
Positional Judgement in Protocol	Yes	No
Global Plan	Yes	No
Strategic Terms	Yes	No
Evaluation in Protocol	Go terms	Sentences

Table 5.1: A General Comparison on Protocol between Experts and Novices

The ability to use Go terms In Go player's protocols, lots of Go specific terms and idioms appear. Most Go players are equipped with good amount of Go vocabularies, but the individual differences occur in their ability to explain these Go terms in words. Even weak players can recognize the patterns using Go terms, however, they are weaker in explaining the terms by placing the stones on the board (retrieval)[17]. In the course of making a move, their utterance contains 'naming', 'purposes', 'candidate moves' most of the time. Even form and positional terms can describe the purpose. Opponent's move, candidate moves and board situations are usually named and referred to by the name. An expert player can use only positional terms to imply judgement and planning. Abstract Go terms (adjectives) used by the experts have more than one correct answer. When a subject looks at the board, he/she is conscious of the purposes and future images are expressed through the Go terms. Language level inferences can be found.

Sometimes a given local configuration on the board may be understood by and refer to different Go terms by different players and in turn, the deviations in naming might cause completely different interpretations of the game in retrospect. One of the examples of pair-Go protocol in Saito and Yoshikawa (1995) shows the situation when one pattern is understood with different terms by the competing teams. After the game had finished. Black team said the reason why they lost the game is because they underestimated the effect of the two white *Kake* in the center, white 54 and 56, of figure 5.4. However, the white team simply called white 54 and 56 as "Keima".

"Keima", the knight's move which is introduced in figure 5.3, is simply a positional term. "Kake" (means *press* in English, is a term that means to prevent the opponent's stone from coming out toward the center by blocking them from above) is a term that conveys the intention and action. *Press* as a definition can have many various configuration on the board. In this case, White's intention of making a knight's move in the center confused Black.

[17]Learning a foreign language has the same phenomena. A word can always be perceived by its visual, aural or spelling patterns; and a word which contains meaning can be used in forms of conversation and writing. Only when the users nearly fully comprehended the many aspects of a word, can he/she retrieve it and use it effectively and correctly.

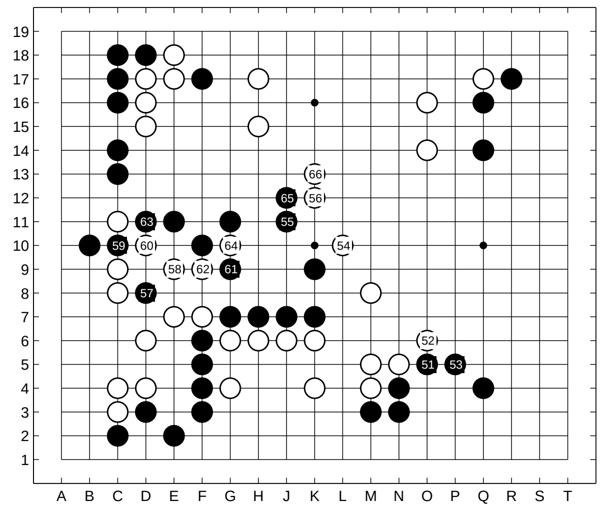

Figure 5.4: A reproduction of an intermediate game record from a pair-Go match demonstrated in Saito and Yoshikawa (1995)

Chunking and Board Representation The evidence of chunking of a static Go board is weak, however, it seems that there exists sequential chunking from the analysis of episodic memory and inferential memory. The players do not pay attention on the whole board all the time, but there is a flow of thought about how the game is led. Two players may also pay attention to different regions on the board. A board configuration has a hierarchical internal representation in the players' mind. The hierarchy is a result of different degrees of abstraction, which is inversely proportional to its relatedness to the current battlefield. This relatedness is not always associated with spatial proximity of the stones or closeness. This kind of hierarchy alters according to the status quo of the game.

Sequential Memory The master players have the ability to remember a sequence of moves by connecting each move into a story. Especially when *joseki* are present, they are easy to be remembered. Unusual moves are remembered as well, because they surprise the subjects. The rest of the moves are remembered because they make sense and are meaningful. The story line that the subjects create facilitates them to predict the successive placement. A static board configuration is read and understood sequentially. A good player will have his own perception on how a board is developed from the beginning.

Decision Making Procedures Go players' decision making procedures contain both spontaneous and elaborate thinking. A forcing move or a recognized *joseki* sequence can trigger fast and inevitable responses. Evaluations of a few candidate moves and selection can be instantaneous. When an elaborate lookahead and evaluation are conducted, the players often become silent. It is natural for a player to think from opponent's point of view. More than 80% of the time, the players understand their opponent's purpose correctly. The losing point of the game happens often when the player fails to understand his opponent's purpose.

The generation of candidate moves Advanced players think about their own purpose and their opponents' purpose at first, only then do they generate candidate moves and lookahead. A few candidate moves are generated from quick and pattern-directed identification process. The number of candidates is one or two and it is very rare that it is more than five.

The look-ahead behaviour Lookahead does not happen all the time, but it occupies about 20-30% of their verbal report. Quick look ahead is spontaneously verbalized and progressive deepening is observed. The average depth is about four, and the most frequent lookahead is just two moves ahead. Branching-over lookahead occurs early, and it is more often for binary-branch than tri-branch and very rarely four-branch. Lookahead in the branches is shorter[18].

Evaluation Evaluation of a move varies according to different phases of the game. In the opening, personal preference and style/feeling are the most frequent reasons. In the middle game, when looking ahead, players visualize the board situation in their head (internal representation) and evaluate that situation according to their current purpose, general good shape or good *moyo* (large framework of potential territory). When there is no lookahead, evaluation is quick and instantaneous. A move is rejected, if it would in turn present a good move (good response) for the opponent, and a move is often selected because it achieves several goals simultaneously.

Go textbooks

Looking into Go textbooks is a good way to investigate how Go knowledge can be accumulated and what kinds of tactical skills are in concern. We can find some constructive procedures of learning Go from the way pedagogical Go tutorials are structured. One realizes that the important issues which concern the Go players are only handful, and the same topics appear repeatedly in textbooks which are claimed to be for different levels. However, the depth in which a topic is touched upon varies drastically. The most direct measure of the depth of a topic is the number of moves required to complete an action.

Especially in pedagogical texts, the evaluation of a certain kind of strategy or configuration is always demonstrated with a small continuation of stones. That is to say,

[18] This is in sharp contrast to orthodox game theory, where the player is capable of even infinitely long lookahead.

the evaluation is mostly dynamic and rarely static. Another big difference is that the evaluation in textbooks is qualitative, and never numerical. Beginners start learning by looking into problems which address local tactics, such as 'to live' and 'to kill'. They are trained in considering a sequence following a good move. That is, a solution is not merely a base-move, but a sequence of moves following this base-move.

I summarize the important issues that Go players learn and study throughout their Go life in the following list. More detailed discussion can be seen after the list.

Elementary Level:Basic and Essential

1. Connection and Cutting: A competition for the cutting point
2. Capture and Escape: The concept of liberty
3. Connection + Capture:
   - The use of edge
   - *Atari* and *Shicho* (ladder)
   - *Geta* (net)
   - To know how to sacrifice for bigger benefit
   - To know how to escape more rapidly
4. Life and Death (two eyes is alive):
   - How to make two eyes from different number of vacancies and different shapes
   - How to identify false eyes
5. *Semeai* (Capturing Race): it begins when both sides have no chance to make two eyes
   - In principle, the side which has more liberties wins, the caveat is that the sequence of play is important
   - The race between 'one eye' and 'no eye'
   - *seki* (mutually alive)
6. Ko: the importance of Ko threat
7. Endgame: Calculating Territory
   - To care about living with bigger number of points
   - To understand that some capturing race is unnecessary

- To be sensitive to the different results in terms of the number of points corresponding to different actions (evaluation)

Advanced Level: Go theory

1. *Tesuji* (strategically important position)

2. *Joseki*

   - To support one's own stones at the corner or side in order to construct a good shape or to destroy the opponent's framework
   - To obtain *sente* in order to occupy the critical position on the board
   - To better execute the plan for opening

3. *Fuseki*

   - Corners, Sides, and Center: With the same number of stones one can surround biggest territory at the corners, and then on the sides, and the least in the center.
   - Center oriented

Duality in Go Concepts One can learn the first two topics in the elementary level within minutes and soon realize that in many cases the position that attributes a connection (strengthening one's own groups) and the one which blocks the potential connection is on the same point. That is to say, if one has learnt how to connect, one already knows how to cut. The same analogy happens in capture and escape as well. This kind of duality in Go results in the competition in inferring and reasoning for Go players. The third task is a kind of skill that combines the concept of connection and capture which attributes to many births of Go terms and variations of capturing. All techniques above are encompassed by the bigger concept, *Life and Death*. However, killing is not the prime purpose of the game, but the competition for territory is.

Problem Sets It is conventional to add few exercises right after each section of the textbooks. When a beginner is provided with a concrete problem to solve, such as "What can White do now to capture Black?", it is not so difficult for him/her to come up with a good solution, as long as he/she understands the concept of liberty. However, it is quite a different case when a real game is played. When one does not have a hint with some guidance, the beginner might not know what to do and to foresee the possible consequence of the move. Their reasoning is very constrained by what they learn about the Go rules and are not able to understand the bigger picture.

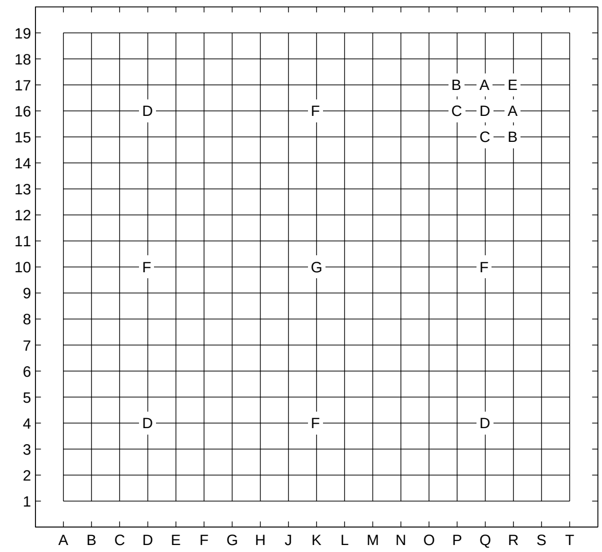

Figure 5.5: Named Position on the Board: A=*komoku* (3-4 point); B=*mokuhazushi* (3-5 point); C=*takamoku* (4-5) point; D=*hoshi* (star point); E=*san-san* (3-3 point); F=*side hoshi* (side star point); G=*tengen* (10-10 point, origin of heaven)

Fuseki The discussion of *fuseki* cannot be separated from displaying few famous locations on the Go board. Some intersections on the board are given absolute names, and these names show up very frequently in the opening (*fuseki*). See figure 5.5. A Go board is highly symmetric. The points *A*-*E* in figure 5.5 are also mapped to their coordinates at other three corners. These points are named based on their distance from the very corner point of the board. Each of these points have their advantage and disadvantage to corner territories and the speed of development, however the theories on each point are based on the possible successive sequence that follow each the placement of these points. These names appear also frequently in *joseki* dictionary.

Joseki Dictionary The collection of joseki (the sequences of moves that are proved to be able to balance the local areas) is the consequence of the institutionalization of the game of Go. That is to say, it is the natural result from the severely limited individual rationality. They are normally named after the positions in figure 5.5, the name indicates that the location where the joseki sequence initiates. However joseki dictionary is a dynamic collection, where new joseki are constantly invented and some old joseki are modified or disputed from time to time. It is believe that memorizing joseki by rote memory is not a sustainable way to become a very good player, instead, one should understand the meaning of each component in a joseki.

The collection of joseki is the result of the archive of Go wisdom uncovered from numerous games and research. The more joseki one remembers, the more flexibility one is endowed with for proceeding with a good opening. However, I doubt that joseki sequences are organized into LTM merely in the form of rote memory. A good player should have understood the motive and reason of each move in the joseki sequence and use them in clever ways. The ability of understanding the reason and motive of each move cannot be built without the elementary concepts and a good fixation of those concepts in the LTM by reinforcing them through practices.

The same argument can be applied to the development of fuseki as well. The birth of new fuseki theory, which emphasizes the importance of central influence, was very prosperous in 1930's. Prior to that era, if a player places the stone at a position which is considered to be an unusual response, he might appear to be unprofessional if he does not know how to support this decision ex-post. However, no Go rule excludes the possibility of "peculiar" moves. On the other hand, if the player knows the "style" of his opponent, he might invent some new strategies to fight with certain new fuseki. As a result, a new fuseki is born. These sequences will not make sense without the meanings attached to them. They are more than just spatial patterns.

From the Go textbooks, the players are guided to learn the important topics by looking at massive number of examples that show only a local segment of the board. When they are at the real match, they are confronted with a big piece of land. There is no aid of problem set which can remind them which side of the problem they should focus on. At least, we can conclude that the players' ability at the tournament is a a result of their cumulative skills for solving specific problems.

Why Go Terms Help? A player can certainly become stronger by getting to know and use more and more Go terms. Even before the players start to know the official Go terms, they have a rough understanding of the spatial relations and some consequences of their actions. As long as those patterns that are familiar as a whole or partially to the beginner are given the names, then these ideas can be encapsulated as packages and stored in LTM. Henceforth, those ideas can be retrieved faster as terms without going back to inference according to the Go board configuration.

Commentaries

I believe that the Go commentary is also a rich source from which we can gain the insights into the experts. It is habitual for a commentator to evaluate alternative moves along with the chosen move, to show those alternatives are not chosen because they are inferior. The way they evaluate is to display an imagined sequence until certain weakness has appeared to the side that is going to play. At most one or two alternatives are considered. It confirms to what de Groot observed and defined as progressive deepening. It indicates that a good player has the ability to know, at least pretty certain in his own mind, how the opponent is going to respond to his each move. In other words, an expert does expect that his opponent will conform to his flow of thought. I believe that Go Seigen is one of the best players is because his moves are often unexpected and innovative.

Go commentary is a retrospective report on a Go match. The entire *kifu* will be displayed with the order of the placements of stones. Some evaluation and comment will be addressed to the moves. Go commentaries is also an illuminating source of Go protocol and reasoning, especially for amateur players. The amateur players have little access to play with professional players; as a result, they can get new stimuli and inspirations by reading the commentaries of professional matches. A complete game can contain more than 200 moves, therefore it is unusual that the commentator explains and evaluates every move in detail. Conventionally, the opening catches more attention. If a players does not know enough Go terms and Go theory, a commentary can be somehow not be comprehensible and may appear too abstract.

Let us summarise the general patterns observed from the commentaries and they reveal how standard games are played as well. A move is either positively an initiative or continuum of a plan, or passively a prevention from an undesired opponent's plan.

1. In the opening, two players will occupy 2 different corners each in turn.

2. Black invades one of White's corners or strengthens his own corners.

3. Several placements on the corner or sides, in turn as the preliminary settlements of future fights.

4. The direction of the game depends on the fuseki used, and depends on the leading party of the game.

5. Tactical fight is at one of the four corners, so we can see a cluster of moves at one of the 4 corners or one of the 4 sides. This should be marked as the beginning of mid-game.

6. Normally, the fight moves from corners to sides and towards the center.

7. If one of the players puts a stone at a region with some intention, these actions may already break the temporary balance of that region, and the opponent might have to find a move to maintain the balance of power or make it a *quiet position*[19].

8. Several episodes are divided by *tenuki*.

9. One of the players will place the stone somewhere else, seemingly unrelated to the earlier battle. It might be a *tenuki* or a *Ko threat*.

[19]Quiet position is similar to Turing's dead position, however I think 'quiet' will be more illustrative and less confusing than 'dead' in the context of Go. It is a local concept that in a region none of the players have the immediate incentives to place more stone there, unless one of the players initiates a capturing race or resolve the life and death problem. It could also be the case that both players have agreed through their mutual (implicit, but in silence) understanding that there is no more need to fight at that area, because there is no other way for one side to turn over the losing battle. Of course, this kind of peace can still be temporary, that is due to the fact that the Go board is not completely decomposable until the endgame phase.

10. If the opposite side answers the *tenuki*, the game might be led to another corner or side. If the *tenuki* is not answered, then there might be some minor battle at the previous area.

11. In the endgame, the players will fill some vacancies in order to obtain the territory in a definite manner.

Pedagogical Commentaries These commentaries are written for educational purposes. Naturally, almost every move will be commented to a different extent of detail. Conventionally, when a real move is formally introduced, the authors mention one to two possible alternatives and display the lookahead (ranging from 4 to 12 stones), rarely with branching, from that position and terminate at some positional judgement (*keisei handan*) with the conclusions similar to "the framework (*moyo*) or thickness (*atsui*) will not be good."

Professional Commentaries Go World Magazine[20] had been frequently publishing commentaries on professional tournaments. They are extremely concise, and only the important, critical and flaw moves will be mentioned. The words used to describe these moves are very abstract, such as "forceful", "unusual", "winning", "unreasonable", "clever". No phrase related to the spatial patterns would be mentioned. A sentence like "White 58 is better at point *a*" appears very often and without further explanation.

Other applications of Go

Berlekamp and Wolfe (1994) applied surreal numbers that is the foundation of Combinatorial Game Theory to solve for the endgame of Go optimally. It requires strong assumptions, such as complete decomposition of the board. The ideology of this method is quite consistent with the faculty of endgame of human players in the sense that it is an important ability to be able to count the territory precisely. However, this ability is gradually acquired and mastered in the later stage of Go learning. Go has been applied to understand cooperation among team mates for achieving organisational goal (Hasan and Warne (2008)). The latest cognitive modelling of Go is the neuro-cognitive model (Bossomaier et al., 2012).

5.3.3 The hierarchies in decision making of Go

> "Note first that Go a is resource-limited problem. Not only the two players compete with one another, but each side's multiple goals compete for resource among themselves.... Here too, the goal structure of Go differs

[20] This is a quarterly magazine published first by The Ishi Press and then Kiseido Publishing Company since the first issue in 1977 until the last issue in February 2013.

> from that of most current problem solving tasks. Any problem solving program faces process restrictions deriving from the computational and memory limitations of the computer. In Go, however, restrictions upon available material resources also are an intrinsic part of the problem."
> Reitman et al. (1974), p. 124

In my limited knowledge of Go, I suppose that the goals or subgoals can be represented as concepts or particular shapes (framework) of a group that is the potential territory, despite the ultimate goal is to enclose more territory than the opponent. In order to clarify the goal, a good player or a good computer program should at least acquire the ability to calculate the territory and potential territory at any given point of time in the game. However, this task should be achieved differently at the different stages of the game.

Decomposability also plays multiple roles in the game of Go. They are conceptual (hierarchical), sequential and spacial and these roles also correspond to each other. In the beginning of a game, the players place the stones carefully in order to create a good *moyo* on the board. At that moment, the board has not been divided completely and clearly. In the endgame, the players may fight over a local area while the other regions on the board are settled. At that moment the board will be almost fully decomposable for score calculation. The most interesting and difficult phase is the middle game period. The boundaries that divide territories are still movable, and the players will focus on one battlefield until a certain situation has been reached. This scenario can be support by the order with which the game is played. In the middle game of a 200+ stones game, there will be always a cluster of stones played in a local area in successions, and then suddenly there will be a *tenuki*, meaning one of the players decides to place the stone somewhere else rather than answering his/her opponent at the previous area.

The spatial relations on the Go board are not always the precise ways to confine the players' attention. Instead, there is a hierarchy of the players attention to the whole board.

> "The degree of abstraction is inversely proportional to its relatedness to the current battlefield. Even if a certain area is far away from the current battlefield, if that area contains some stones which are intrinsically related to the current battle, those stones are precisely represented in detail."
> Saito and Yoshikawa (2000), p.254

I believe that the use of specific Go terms plays an important role to decide the players' perception and decision. As a player enters a different phase of a game, he/she needs to move back and forth from different levels of attention and concepts. The Go terms play the role of orienting the players.

As a summary of what have been observed in the Go documentaries of problem solving, it is evident that despite the easy and unambiguous rules of the game of Go, there is a set of ambiguous, ill-defined, unwritten rules that is known and respected among most of the players. This is more related to what Simon would call, culture. There are some styles or methods of play that would be considered as unusual, unwise, or even rude, despite the fact that they are not excluded by the rules. Although there are absolute definite ways to generate all legal moves, but a) the criteria for judging whether the goals (intermediate more than final) of the problem have been achieved is too complex and b) the scope of information needed for solving this problem is not definite. For these reasons, I would like to place the game of Go on the grey boundary between well-structured and ill-structured problems. Since there is no clear-cut division between these two types of problems, I claim that the game of Go is more ill-structured or general than Chess. Go players have to cope with the complexity by using heuristics that are encapsulated by the rich collection of Go terms.

5.4 A Pseudo IPS of playing Go: COMPOSER

5.4.1 Highlights of Components and Characteristics of COMPOSER

Before we see the mechanism of COMPOSER, deeper discussions into the relevant topics and aspects of problem solving of Go may be needed.

Sequential Memory

An Example of Sequential Memory with Go Terms It is evident that the memory of Go players are sequential. That is, it takes a certain order to create some patterns. In figure 5.6, the stones Black1, Black3 and Black5 all together construct an usual pattern in the opening. The Go board in figure 5.6 is displayed with European coordinates, i.e., the black stones' locations are, R17, R4 and Q15 respectively. Instead of recognizing this pattern of three stones with their board coordinates, it is more likely that they were first understood as a sequence of meaningful placements. Black1 on R17 is 3-3 point, and Black3 on R4 is named *Komoku* (4-3 point). The first two stones together is a typical way to occupy the corners at the right hand side of Black. Black3 is too close to the very corner despite that it effectively secures the up-right corner, so Black5, a Knight's move from Black3, is placed to support Black 3 at the corner. In a nutshell, it is the relation between the successive stones and the current stone that provides the meaning for the last placed stone. Imagine that Black1's position is a term called X, and Black3 is called Y and then the knight's move from Black3 is called Z; from time to time XYZ might have become one of the familiar vocabularies for Go veterans. XYZ is a vocabulary composed of 3 symbols. An expert might be able to memorize and utilize the "vocabularies" and "phrases" which contain more symbols.

An expert who can memorize a whole game of more than 200 stones and comment on each move is like an advanced reader who thoroughly understood the meaning of each move of the game. A good player is metaphorically like a good composer. None of these symbols (positional or spacial relations) and vocabularies (sequences of moves) are defined in Go rules, but recognized and verbalized in to terms used

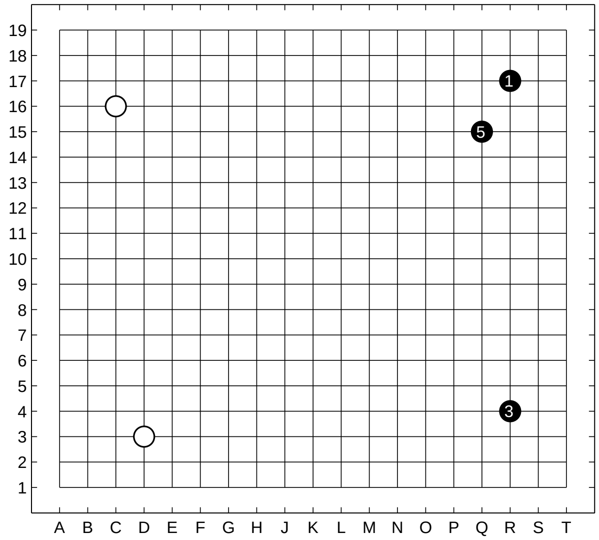

Figure 5.6: An Example of Sequential Chunk in the Opening

in Go textbooks and among players. Go terms consists of different hierarchical aspects. The abstract Go terms used by advanced players are constructed and defined by sub-level terms, and sub-level terms are constructed by sub-sub-level terms, etc.. Naturally, a player who has studied Go textbooks and has experience in playing complete Go games should have already acquired a good amount of terms.

EPAM Net: Stimulus and Responses

Iceberg Model and Template Theory This version of COMPOSER is inspired by the proposal of Iceberg model that is similar to template theory, but the slots in the templates should be dynamically assigned according to the situation on the board and they vary according to the level of the player.

A discrimination net in EPAM functions as a general but dynamic catalogue of memory structure of a particular domain. The terminal nodes are images and are essentially list structures which are internal codes of external patterns. If we adopt the similar discrimination net for organising Go knowledge, then the terminal nodes must also contain programs which are the procedures of some specific actions in Go, such as, capturing race, Ko threat, ladder, life and death, connecting and cutting. The terminal of the discrimination net can also be a familiar sequential chunks as the outcome of frequently used production or chain of productions. This architecture suggests that an external stimulus can lead to a response that activates a program which checks the possibility of certain action.

The discrimination net is developed by encountering more and more stimuli. The processor installs a new test to discriminate an item whenever it is not recognized, that is , it does not belong to any terminal of the current net. The most elementary level of Go terms are learned first, so that they will become the base for higher-level Go terms. Learning is a bottom up cumulation and playing is top down performance. They are both tied to hierarchical components in the memory. For simplicity, the effect of forgetting should not be considered in the current model, but whatever is learned will be strengthened by more knowledge and practised effectively.

Production Systems

COMPOSER includes production systems which are the core of IPS to explain the sufficient amount of inference that appears in players' protocols. Inference is the reasoning that requires more than perception and pattern recognition. Identifying the Go board configuration with the Go terms or Go concepts can be accomplished by entering the EPAM-like net (discrimination net). However, the meaning of the Go terms that the players infer from the Go board require *production* like relations: "Term A & Condition 1 $\rightarrow$ Meaning or Purpose". The production system is a suitable way to specify such reasoning activity. Moreover, a production system that is associated or attached to a Go term or concept is stored in the template of the Go term, which is in turn at a terminal of the discrimination net.

Some inference happens at merely semantic level, such as "if the opponent is trying to escape then I should block it"; such inference does not necessary involve any specific Go terms. By doing so, the player can generate only few candidate moves. My conjecture about the number of candidate moves is that the higher the inference level, i.e. more specific in semantic meaning, the the fewer the candidate moves are. Go Seigen's wisdom says that one should look at the Go board as a whole. I suppose that one needs penetrating observation and determined dedication to reach this level. What actually involves in acquiring this ability is the very high-level inferences instead of intensive lookahead.

One of the advantages of the production system is that the production can be added independently one by one after the previous productions without affecting the system which already exists. The analogue of addition of productions into an existing production system is that as the player learns about more patterns and implicit rules of the game, his strategies will be enriched by adding more conditions and actions. For example, after the player learn about Ko in Go, he might develop a situation when the Ko thread can be applied.

Knowledge Acquisition and Representation

COMPOSER's structure is not based on any particular subject's behaviour, but refers to a widely accepted and observed tendency of Go players. As a result, there is no legitimate reason why one should discriminate export systems from the architectures of intermediate players and beginners. Because, after all, the current mechanism of the program simply represents a state of development of a problem solver. Programming human expertise is a very justifiable approach to achieve strong AI. However, one cannot ignore the transition from being a weaker player to a stronger player. Because a grandmaster is not a born expert, instead expertise is a consequence of accumulating knowledge and massive practices of memory retentions and retrievals. One cannot view expertise as a steady state and model it.

Apart from the explicit rules of Go, such as the definition of a chain, the legitimacy of a capture, there are many implicit patterns and names which are known to Go players. I distinguish those patterns that are absolutely specified (see figure 5.3) and those patterns are only defined and thus refer to a membership, such as *atari*. Those absolutely specified patterns can be associated in a discrimination net. However, the Go terms that work on definitions should be organized as cues and programs that check whether the definition is satisfied.

Regarding the evidences collected from cognitive and protocol analyses, it is reasonable to assume that even a Go beginner's knowledge contains many definable Go terms. What really differentiates the strength of players is their different organization of these terms. These differences are embedded in the templates that are attached to each Go terms in the discrimination net. COMPOSER should be a system that combines EPAM-like discrimination processes and GPS-like inference processes.

Equipped with fundamental pattern-related Go terms and the basic list of productions, an ideal Go playing architecture should be allowed to learn from its own experience (at the spot or retrospectively) or from external knowledge. I propose that the model can learn from a detailed commented *Kifu* (game record). A annotated *Kifu* will be stored in a separate storage in LTM. When I say commented *Kifu*, I mean each move is marked to a Go term (Go verb or noun) and if possible, with some adjective or emotional comment, such as "too heavy", "*tesuji*", and "losing move". The program has very limited access to the data in this storage while it is solving a problem. However, when the program is not assigned to play a game, it has free access to it even before it can fully understand the meaning of each move that is played and associate each component to its existing discrimination net.

First of all, the annotated *Kifu* should be encoded into the language that the program can read, so that it can classify the sequence of the whole game into opening, midgame, and endgame. The information about different phases of the *Kifu* will be associated to different faculties. When the program is "reviewing" the game, its attention follows the flow of the game exactly like it is playing the game. When a move is unfamiliar to the program, the comment that is attached to the move will be learned; when a move is familiar and its comment is what already known by the program, this piece of knowledge is either not altered or reinforced. When a familiar move has an unusual comment on it, the program learns to discriminate it further or temporarily ignores it and marks it as something still unclear. When there are more and more such comments appear in the database, it will decide to digest it.

General Discussion

I list out a number of properties that the IPS as human problem solvers of Go should equip itself in the light of the above evidences. These properties are further categorized into general properties and domain-specific properties. Later on, some of the properties will be applied to build pseudo IPS models. No attempt in this thesis is made to encompass all the properties in one go.

- General Properties

  1. Heuristics and satisficing: The database of Go knowledge in the LTM assists the human heuristic behaviour. The IPS's intensity in reasoning and computation is constrained by the capacity of STM and time. The macro-processors will not embark the kind of search which is too time-consuming when a good amount of time is already wasted in the course of the problem solving.
  2. The ideal IPS of Go should permit the flexibility for learning and interpreting the qualitative difference between masters and a intermediate players, and the difference between intermediate players and beginners. The learning activities include the implicit learning from the problem solving tasks and the explicit learning from education.
  3. Opponent modelling is a reflection of the player's own knowledge, unless there is more information about this opponent, i.e. his style of play. In the later cases, the prediction of opponent's behaviour can be pinned down to a subset of the player's own knowledge.
  4. There exists a preliminary processor which takes care of the Go rules. It will be able to identify all the legal moves. There is a evaluation processor which takes care of positional judgement. It works closely with the macro-processors and micro-processors that are stored in the templates. Because at different phases of the game, different kind of judgements are needed. Positional judgements answer various decision problems and report either "good" or "bad."

- Domain Specific Properties

  1. *Kifu* should also be the source of the learning activity of the IPS. The examples of wrong moves or sequence should also be collected in the database.
  2. Productions: There should be interconnections between some productions in the production system and memorized sequence of play. The very frequently adopted chain of productions may gradually become a fixated sequence of moves in the memory. A simple example shows that "If A then B, if B then C" may become "If A then B then C". When C is not a desired position, then A is denied. This organization of the database should allow

the forming of standardized sequential chunks from familiarized chained productions. Likewise, joseki sequences should not be remembered by rote memory, instead they should be associated with some production systems.

3. The program should distinguish between playing from the beginning on an empty board and choosing a move from a given configuration. These two kinds of task environments should trigger different procedures.

4. Sequential Chunks: Go players should have the ability to start from remembering, say, a sequence a 2-3 stones, to a sequence of 10-15 stones and to the whole game[21].

5. Handicap and Komi. Because the rule of Black's komi, Black sometimes would like to speed up the opening; for example, the 4-4 is a better start.

6. Go veterans are sensitive to the change of balance and thickness (positional judgement). I believe the stronger a player is, the bigger fragment of a board that he/she can read and judge. Recall that Go Seigen focused on "balance" in his play. The professional players look for a position which maintains the balance of two players. I believe this idea of balance is hierarchical and dynamic[22].

7. Purpose of own moves and opponent's move are central in players' reasoning. When a stone is placed, no matter it is a own stone or opponent's stone, it is a seed of a purpose that might be realized only few steps later. Following this seed, there is a sequence of imagined moves in the player's head [23].

The purpose of constructing such an pseudo IPS is to demonstrate a possible and potential mechanism that explains the Go players abilities of thinking, communicating and decision making with Go terms. One of the unsolved problems of playing Go is that Saito and Yoshikawa were not able to answer how a good Go players can come up with only few candidate moves in short time. This section provides a tentative but possible mechanism with which the this question can be answered. My hunch is that higher is their level of reasoning in Go concepts, the fewer and more precise they can

[21] I believe when they remember the whole game, they memorize the sequence by dividing them into episodes, and the dividers are *tenuki*. A *Tenuki* is like a punctuation. Within episodes, each move is connected with each other by their meanings or Go verbs. Of course some of the segments of these sequence have become *Joseki*. We can imagine the whole game sequence is a long article, and the episodes are the paragraphs. *Joseki* becomes phrasal verbs or idioms.

[22] From the opening of the game, they will keep the balance for the whole board, and then they search for regional balance, and then local. When the battle becomes local, few lookahead are necessary. While, when they look ahead, the continuation seldom branches out. Besides, they are able to give each move a meaning, that is, the sequence in looking ahead is connected with a story. Therefore, when they expect that a sequence is leading to an unbalanced or disadvantageous situation, they will exclude that position from the search space.

[23] In COMPOSER, purposes are tied to (programs) algorithms or production systems. Each program decides whether a certain action is possible in a finite numbers of moves.

be in generating good candidate moves. In particular, I utilized the players ability to use Go terms in this model.

As mentioned in previous chapter, CHREST is a remarkable program that provides a more comprehensive structure, which is dedicated to Chess thinking. However, verbal processing is overlooked by CHREST, instead as it suggests, playing Chess is a highly visual activity. On the contrary, as it is analysed above, Go players' reliance on all levels of Go terms is significant. The meanings and purposes attached with the Go terms play a radical role in the decision making in Go. Certainly, pattern recognition is a very important faculty in Go playing, but it is not enough. Borrowing the terminology from Chess, when a chunk or chunks are recognised as their names (Go terms), some inference based on the Go terms are conducted in order to generate the purpose of the player, and then candidate moves are generated and some lookahead is implemented.

The sophisticated nature of Go results in complex and hierarchical knowledge structure required to be a Go master. Their skills should at least be categorized into spacial, tactical, strategic and psychological. Language or Go terms play a central role in Go thinking. The qualitative structure of COMPOSER is demonstrated to show how and why labelling Go patterns and situations can help reduce the search-space drastically. We attempt to build a reasonable knowledge structure where Go terms and Go concepts are properly involved.

5.4.2 Discussions about Go Terms and Concepts

The game of Go is played and studied by four main ethnic communities: Chinese, Japanese, Koreans and the Westerners. This game is called "Weichi" in Chinese, "Igo" in Japanese, "Baduk" in Korean and "Go" in English. English being the necessary language of communication in international occasions for Go shows its preference in favour of adopting Japanese terms into English terminologies. Many Romanized Japanese Go terms are adopted into the glossaries of English Go terms. The reason might be that romanized Japanese is more pronunciation friendly. However, such adoptions do create communication problems to Korean and Chinese players who participate in international congresses and tournaments. One thing in common is that every community utilizes huge amount of Go terminologies in their respective languages and fortunately, many but not all concepts that can be named have their counterparts in other three languages. In some cases, a term has variations that are used by different groups of associations, publishers and researchers. There is no unifying version of Go term dictionary in the international communities. Professor Chihyung Nam from the Department of Baduk Studies at Myongji University, South Korea has made the first attempt to systematically collect the logical definitions of English Go terms with their counterparts in other three languages (Nam, 2004).

I believe that the emergence of Go terms are the consequences of the accumulated knowledge of the game of Go over thousands of years. An immediate analogue of this phenomenon is the evolution of natural languages for us. It is debatable whether a concept or the form of an object came first or the name of that concept came first in our daily reasoning and communication. However, it is quite clear that naming in the game of Go can either encapsulate the already existing concepts or define a new concept for the players. Two examples of these two cases are given as the following. First, a concept such as a chain of stones has only one liberty left and is about to be captured is familiar even to a beginner. Because it is derived from the Go rules. The concept exists in the beginner's memory before he/she gets to know that it is called *atari*. After the term *atari* is learned, the player can associate this concept with other concepts faster. Second, the concept like "large knight's move" (*Ogeima*) was not innate in the players' understanding of the game until the term is defined. The term is connected to strategic and tactical concepts and the development of joseki sequences.

Apart form the various categories and hierarchies of Go terms, these terms are not entirely independent of each other. Some Go terms are based on a combination of more than one Go term. From the hindsight of Go textbooks, we have already seen at least two associations of Go concepts: i) the association of connecting and capture, and ii) the association of capturing race and one eye.

In Go societies, intermediate players and the above can recognize almost as many named Go terms as experts, while they can not retrieve, demonstrate and explain some terms as well as experts. However, the same term or concept, when named by different languages with their slight difference in literal meaning, might evoke different interpretations and cultures of play by different communities. More importantly, a massive number of patterns can refer to the same Go term. For this reason, it is difficult to estimate the number of chunks stored in an expert's LTM based on the definition of chunking theory developed from studies of Chess.

There is a hierarchical organisation among Go terms. The elementary level is pattern and position related, such as *atari* and the positions introduced in figure 5.5. The second level can refer to the Go verbs; some of them describe the binary relations of two stones, the latter placed stone is the transitive verb having the first stone as the object. The examples of this category are in figure 5.3. Each such term has its own tactical meaning (tied to other actions or higher-level Go terms) that reflects on the later development; of course, how a player reads the potential meaning of these terms depends on his strength and experience. The terms introduced in figure 5.3 are restricted to the relations on two stones of the same color. There is another set of Go terms referring to the same binary relations on stones of opposite colors. Because the tactical meanings they carry are very different. Higher level of Go terms are strategic concepts which require more experience and commitment in order to understand them, the examples of these terms can be found in Nagahara (1972).

There are always more than one relation a person can read when a stone is added to a board. My hunch is that experts are able to eliminate the objectively multiple relations to only one or two and they are called by the Go terms. According to the Go terms, the expert can work on the opponent's purpose and then derive their own strategies and then generate good candidate moves. On the other hand, the multiple relations on the Go board might be too overwhelming for the beginners.

From the diverse characteristics of Go terms that are used heavily in the problem solving of Go, I conclude that the game of Go is semantically richer than Chess. It is evident from its multi-level Go terms library. Thus, when there are number of levels of Go terms/concepts, using these terms in the course of a game to make good moves involves different levels of semantic reasoning[24].

5.4.3 Descriptions of Composer

The construction of COMPOSER takes into account how Go knowledge is acquired by human players, and how Go memory is organised. If one can get this part of the whole structure fairly right, then the performance of a problem solving case is simplified to an execution of the existing system. This idea is inspired by the way in which EPAM utilizes the discrimination net.

Having many sophisticated and competent computer programs of Go, the motivation of building COMPOSER is not to participate in the brutal race. It is more fruitful, at this moment, to back off and inspect what latest Computer Go approaches have overlooked. Knowledge-based algorithms play a role against brute-force algorithms, however, the game of Go might contribute new insights to what has been known about domain-specific knowledge and expertise.

Composer is an IPS for playing Go. It has some elementary macro-processors and microprocessors. The most elementary processors take care of the basic Go rules, so that the illegal moves will not be considered later in other processes. There is a short-term memory which holds up the input, output and intermediate variables of the elementary information processors. STM has the capacity of 5-9 objects. Long-Term Memory contains all the Go knowledge, production systems and programs that relate to actions and purposes. COMPOSER works very intensively with the Go board, which is its external memory.

Discrimination Net The Go knowledge that is categorized by Go terms and programs are organized by discrimination-net like hierarchical structures. Especially, the production systems, programs that are related to Go terms are stored in the template at a terminal node of the discrimination net. The Go knowledge database is organized into several faculties, such as *fuseki*, midgame, endgame and positional judgement. Each faculty manages its own discrimination net, whose terminals are Go terms and possibly with its cue and template. A recognized pattern that is labelled by a Go term can also trigger a familiar sequence of play that is stored in the template.

Positional Judgement Positional judgement is a program that is called from time to time. It is a program designed to check the potential territory and the balance of a region and the whole board. It deals with several specific decision problems that are tied with the own purpose. It works intensively with Joseki archive and Go terms

[24] There is a connection with Craik and Lockhart (1972).

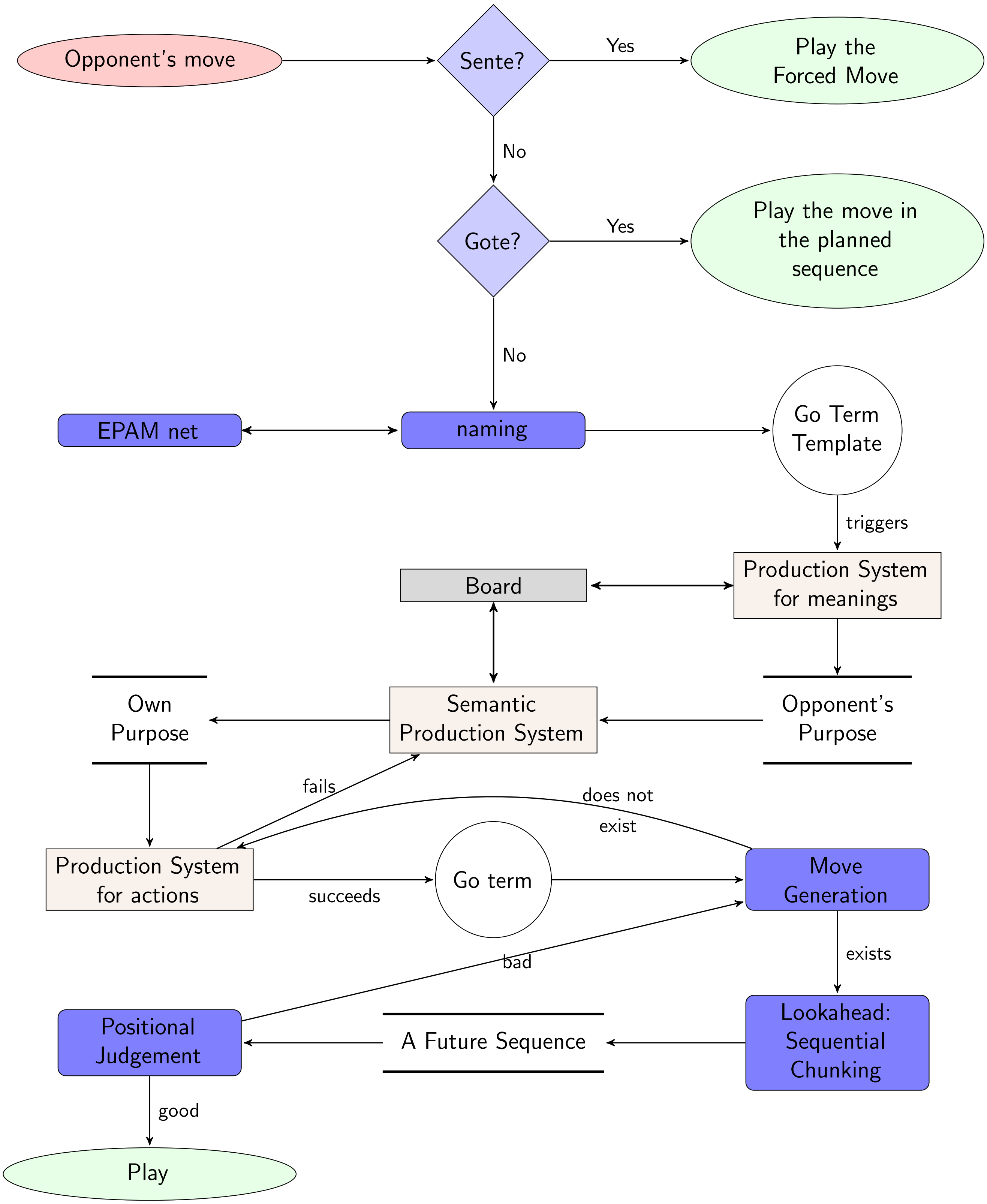

Figure 5.7: The Procedure of COMPOSER

(mostly the adjective Go terms and the terms related to situational judgement e.g. *aji*). It has a database of wrong examples and very good examples.

Macro-Processes Not only is the Go knowledge classified by the different phases of the game but also the macro-processes have at least three departments: Fuseki, Mid-Game and Endgame. There is no clear cut among the departments. Roughly speaking, as soon as some capturing race appears on the board, then the macro-process will mark it as the end of Fuseki; and when the whole board becomes fully decomposable despite some *dame*, it begins with endgame. The macro-processor is aware of the phase of the game, so it will deliver the *board situation* accordingly. Joseki dictionary belongs to Fuseki department, but it is not only documented with huge amount of joseki sequences[25], but they are also organized into a discrimination net according to their properties and components. The endgame problems under the assumption that the board is fully decomposable can be effectively solved by using Combinatorial Game Theory, therefore it does not concern COMPOSER for the moment.

Attention The board situation that the macro-processor reports is filtered with its dynamic attention. The board can be geographically decomposed to 9 regions: 4 corners, 4 sides, and the center. I believe the center can be further divided into 4 small squares. Depends on the flow of the game, e.g. whether it is in an episode of a fight on a region and whether the central planer would like to continue the fight, the processor will control the focus on the board, i.e. whether it should be global or local. This decision can be hinted by the location of the latest opponent's move. When it is focused on one of the regions, the positional judgement about the rest of the board will be more abstract, i.e. only the most general report about the other regions are retrieved. The abstract report may go like: White's region, Black's moyo is good.

Opponent Modelling Understanding opponent's purpose is a reflection on the player's knowledge. Therefore, there is no need for extra system of modelling the opponent. The process of yielding opponent's purpose is characterized as a kind of production system in this model. Likewise, generating a Go term that corresponds to the own purpose should be similar to a duality or a mapping to the production system mentioned above. In other words, a player need not to spend as much time to construct a brand new production system for yielding actions from own purpose as he/she did for the production which yields opponent's purpose from a Go term. It is likely that the opponent's purpose is misunderstood though. It is often a losing point of a game.

Figure 5.7 illustrates a general procedure of making a move by COMPOSER in the course of a game. This procedure elaborates the model proposed in Saito and Yoshikawa (1996) (p.73). The EPAM net and positional judgement vary according to different states of the game which is supervised by the macro-processor. For example, if the

[25] There are about 20,000 joseki sequences in theory, and only up to 500 of them appear more often in the real games.

game is in the opening state, then the EPAM net is *fuseki* and *joseki* dominated. The first decision in Figure 5.7 asks whether it is a move that the player has to answer (a *sente*), otherwise the player will undergo a loss or bad situation. When it is decided to answer, then the possibility is usually only one. The second decision basically asks whether the opponent's move follows an early planned sequence in mind. If yes, the player can simply play out the next move in the plan.

COMPOSER will become an executable algorithm when Go-specific contents are inserted into it. It contains several states of process. In general, a state receives an input from the previous state and operates on the information, and then passes on the output to the next state. In some cases, the output will be examined to see whether this is satisfactory. When an output is not satisfactory, the procedure that generates it will be assigned to regenerate a new output. These evaluation processes are controlled by the internalised and endogenously evolving aspiration levels, so that the system *satisfices* and will not go on to infinite loops. Satisficing here can be thought of as a stopping rule.

There is a central planner or macro-processor which is in charge of directing the phases of the game and providing a board that is layered with hierarchical information. Only the local region that is the temporary focus will be perfectly revealed to the processes triggered at different state of the computation. The rest of the board will be marked as, for example, White's territory or unsettled battle. Only when the micro-processors require more information, deeper information can be revealed. In this way, the micro-processors's burden is reduced by being provided the radical focus of the board.

One remarkable component of COMPOSER is its double-level production systems: purpose/action level and semantic level. When COMPOSER recognizes and labels the opponent's move with a Go term that is one of the terminal nodes in the discrimination net. The term triggers a *purpose-producing* production system, whose condition part calls for information on the board, and whose action part contains the purpose of that move. After the purpose of the opponent is generated, it is delivered to the *semantic* production system which generates appropriate own purpose according to the climate of the game as well. The own purpose will in turn be the input of the *action* production system which generates relevant a Go term for the move generation. Since different production systems are evoked at any given point of time and the program's attention is changing, I can fairly claim that the problem space on which the program works is adapting too.

5.5 Concluding Remarks

In this chapter, we explored the research on the verbal reports of Go players, their behaviour and performance. The emphasis was on the abilities of players in using and reasoning by means of Go terminology. These observations were incorporated into the construction of an pseudo IPS, COMPOSER, whose structures are faithful to the premise of the theory of human problem solving.

Go terminologies play an important role in decision making of Go when compared with their counterparts in other domains. Besides, Go-terms should not be mistaken as mere counterparts of chunks in the case of Chess. The role, use and properties of Go-terms are far more general. They are not only huge in magnitude, but also organised in hierarchy. From this observation we suggest that Go is a semantically richer domain, this aspect is revealed by a higher-level production system of COMPOSER.

The result of this chapter can be extended to several directions.

1. To verify COMPOSER with more comprehensive and detailed verbal reports and insert domain-specific knowledge into COMPOSER and turn it into a computer program.

2. To extend the characteristics of COMPOSER in the investigations of problem solving in more complex economic problems (the problems faced by economic agents not the problems defined by the economists) whose goals and scope are more ambiguously defined in general.

As for as obtaining better data, the ideal way to fairly collect and record the games and extract the players decision process is to have them playing in the context of tournament. The tournaments should be done with the computerized setting and the players are physically separated, so that the players' thinking aloud protocols can be collected without intervening the game. Apart from recording the whole game, there should also be the record of the time spent on each move. Besides, there should be a third expert who observes the whole match and later on comment on the whole game. We should interview the players after the games are finished, if possible.

The knowledge representation of Go shows that many implicit regulations known by players have emerged from extremely simplistic elements of the game and its rule. This aspect of Go, in my opinion, can reflect the subtlety and heuristics in economic decision making.

Chapter 6

Conclusion

In this thesis, a distinction between Classical and Modern Behavioural Economics was made and it was suggested that Information Processing Systems are a reasonable and more general formalisation to model human decision makers who are facing complex and uncertain environments. Modern Behavioural Economics has now developed into a solid line of research that continues to provide alternative models and evidence from experimental environments. Despite that, an important premise remains unchallenged by MBE - i.e, optimization, except for some occasional protests from here and there. Classical Behavioural Economics, pioneered by Herbert Simon serves as a fundamentally different, multi-disciplinary platform to observe and theorize about decision making in the face of complexity and uncertainty. By putting oneself in the shoes of a Modern Behavioural Economist, it is often difficult to see the general impossibility of optimization. For MBE, decision makers' problems are still formulated as (modified) mathematical optimization problems, taking into account psychological and emotional factors. This thesis discussed the underpinnings of CBE from a computability theory point of view, which serves as a useful lens to understand the ideas that Simon and Turing have advocated.

The recurring and central concepts of this thesis were *bounded rationality*, *satisficing*, and *heuristics*. These notions were examined and elaborated by making a distinctions between CBE and MBE, the algorithmic underpinnings of the former - in the light of computability theory and computational complexity theory, and the characteristics of Information Processing Systems in Human Problem Solving, respectively. It concludes that in the general cases of decision making, the *best* solution often cannot be found *procedurally*, but a good solution may. Put formally, optimization as a method to find the solutions is merely a special case of satisficing, which appears to be a natural procedure of human decision makers who suffer from severe limitations in information processing.

This thesis argues that CBE should exploit the richness of the concept of Information Processing Systems in Economic environments to understand decision making. This involves both construction of IPS and studying its theoretical properties. It will also help us to look for ways to extend and move beyond the well-structured problems in Human Problem Solving to more complex economic problems. The game of Go is being put forward as a candidate platform for this transition. There are several directions in which this paradigm can move forward - both in terms of the breath of issues or deeper investigations.

Organisational Decision Making The logic behind the construction of an IPS that is able to mimic Go players' behaviour can be suitably modified with domain specific information and applied to organisational decision making. One way is to externally observe the members' behaviour in the naturally hierarchical organisations. The other way is to extract the external task environment and the internal problem space that the members work with by analysing their verbal reports.

Near Decomposability Near Decomposability, which may appear to be remote from CBE, is however an integral concept in this tradition. Although it is not discussed in great detail in this thesis, it has implicitly appeared as an essential property in various forms and ideas in our discussions earlier. The presence of decomposability in problems can help simplify the problem space from the problem solvers' point of view. Unfortunately, the problems in the real world are very rarely fully decomposable. As a result, the assumption of decomposability that is often employed in economics can lead to bad approximations. Near decomposability, which is also mathematically defined on matrices like the case of decomposability, provides the flexibility that goes beyond to simplify the problems for the observer and the solver. It allows interesting and complex phenomena and evolution to emerge from the weak channel between two sub-systems that is considered negligible in the short run. In the future, we can explore the property of near decomposability as a form of heuristics for a problem solver to simplify task environments and the problem space. It can be useful in investigating different aspects of problem solving across domains and its role in complexity reduction and learning faced by the problem solvers.

Understanding Learning An important lesson obtained from building the pseudo IPS for playing Go is that human reasoning and behaviour is not a mere outcome of a search in huge knowledge database that is stored in a problem solver's mind. Instead, as an organism that is able to think, reason and learn, we have the innate abilities to associate and transform information or knowledge into heuristics that facilitate us in coping with many complex situations. When a general architecture which mimics such innate abilities is built, then we are ready to "teach" the machines to become different kinds of experts. This is also true for organizations, which are constantly learning and evolving.

Our insight is in some ways consistent with Turing's suggestion about a machine which is able to learn by receiving education:

> "If a machine were able in some way to 'learn by experience' it would be much more impressive. If this were the case there seems to be no real reason

> why one should not start from a comparatively simple machine, and, by subjecting it to a suitable range of 'experience' transform it into one which was much more elaborate, and was able to deal with a far greater range of contingencies. ... Let us suppose that it is intended that the machine shall understand English, and that owing to its having no hands or feet, and not needing to eat, not desiring to smoke, it will occupied its time mostly in playing game such as Chess and *Go*, and possibly Bridge. ... As I see it, this education process would in practice be an essential to the production of a reasonably intelligent machine within a reasonably short space of time. The human analogy alone suggests this. "
>
> Turing (1951), p.257-8, italics added

Alan Turing has shown us the ability of the Universal Turing Machine in simulating all the programmables (i.e. It is ready to learn); This seemingly simple, almost playful, mathematical construction is shown to be capable of computing all that is intuitively calculable. Simon on the other hand has established the Human Problem Solving approach, in which IPS (that are capable of universal computation) are capable of providing procedural characterizations of the heuristics involved in human problem solving. This thesis has taken a small step to incorporate the relevant aspects of their contributions and put them in to one coherent algorithmic framework for understanding economic decisions. This involves building IPSs that encapsulate domain specific knowledge in different areas of economics. This vision - of Classical Behavioural Economics - I believe, presents a more general behavioural approach to investigate economic problems using algorithmic and constructive methods.

Appendix A

Go terminology

The Go terms used in this chapter favour the Japanese version (romanized Japanese). Some of the English terms directly adopt the Japanese counterparts. The definitions are from Nam (2004). The texts in the parentheses are the English translations, and if there is no parentheses behind the term, it means the Japanese and English terms are the same.

- **Aji** (Aji/Potential Trouble): Unpleasant possibilities remaining for one player in a position, after a local sequence has been played out.
- **Atari**: An immediate thread to capture, which leaves the opponent's stone(s) with only a single liberty.
- **Atsui** (Thick): A characteristic of stones that are strong and solid with no weakness or potential trouble, so they have influence in a certain direction.
- **Dame** (Neutral Point): An empty point on the board, which is not a part of either player's territory and has no prospects of becoming territory.
- **Fuseki** (Opening): The initial stage of the game where the players place stones in preparation for middle-game fighting and for making territory.
- **Geta** (Net): A capturing technique that blocks all avenues of escape of the opponent's stones without touching them.
- **Gote**: i) A move which does not require the opponent's answer. ii) A position in which one is forced to answer the opponent's last move.
- **Joseki** (Joseke/Pattern): A formulaic sequence of moves which is established for giving equal outcomes to both players.
- **Kake** (Press): To prevent the opponent's stones from coming out toward the center by blocking them from above.

- **Keisei Handan** (Positional Judgement): Evaluating the state of the game or estimating the territorial balance.
- **Kifu** (Game Record): A diagram that shows the moves of a game.
- **Komi** (compensation): A set number of points given to White for making up his loss which results from Black's taking the first turn in an even game. It is usually set 5.5 or 6.5 points in modern Go.
- **Moyo** (Framework): A territorial outline, made up of several strategic points that can become actual territory as the game continues.
- **Shicho** (Ladder): A sequence in which one keeps giving atari to the opponent's stones until the stones are driven to the edge of the board or to friendly stones and captured.
- **Seki**: A situation in which two combined opposing groups without two eyes cannot kill each other because the internal liberties cannot be filled, so both are considered to be alive.
- **Semeai** (Capturing race): A local skirmish in which the stones of both sides are surrounded and each have a most one eye. Each player is reducing the liberties of their opponent's stones. The first one to capture the opponent's stones wins the race and his own stones live.
- **Sente**: i) A move that requires the opponent's answer. ii) A state in which one an enforce the opponent to answer every move in a certain sequence and then play elsewhere. iii) The privilege of not having to answer the opponent's last move and being able to choose freely where to play next.
- **Tenuki** (Play Elsewhere): To ignore the opponent's last move and make a move in another part of the board.
- **Tesuji**: An important or key place in a local position.

Bibliography

Akerlof, G. A. and R. J. Shiller (2009). *Animal Spirits: How Human Psychology Drives the Economy, and Why It Matters for Global Capitalism*. Princeton: Princeton University Press.

Baylor, G. W. J. and H. A. Simon (1966). A Chess Mating Combinations Program. In *AFIPS Conference Proceedings, 1966 Spring Joint Computer Conference*, Volume 28, pp. 431–477. Washington D.C.: Spartan Books.

Berlekamp, E. and D. Wolfe (1994). *Mathematical Go*. Wellesley, MA: A K Peters.

Bernstein, A. (1958). A Chess-playing Program for the IBM-704. *Chess Review*, 208–209.

Boorman, S. A. (1969). *The Protracted Game: a Wei-Ch'i Interpretation of Maoist Revolutionary Strategy*. New York: Oxford University Press.

Boring, E. G. (1946). Mind and Mechanism. *The American Journal of Psychology 59*(2), 173–192.

Bossomaier, T., J. Traish, F. Gobet, and P. C. R. Lane (2012). Neuro-cognitive Model of Move Location in The Game of Go. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–7. IEEE.

Broadbent, D. E. (1958). *Perception and Communication*. New York: Pergamon Press.

Bruner, J. S., J. J. Goodnow, and G. A. Austin (1956). *A Studying of Thinking*. New York: Wiley.

Burmeister, J. M. (2000). *Studies in Human and Computer Go: Assessing the Game of Go as a Research Domain for Cognitive Science*. Ph. D. thesis, School of Computer Science and Electrical Engineering & School of Psychology, The University of Queensland, Australia.

Burmeister, J. M., Y. Saito, A. Yoshikawa, and J. Wiles (2000). Memory Performance of Master Go Players. In H. van den Herik and H. Iida (Eds.), *Games in AI Research*, pp. 271–286. The Netherlands: Van Spijk, Venlo.

Burmeister, J. M. and J. Wiles (1995). The Challenge of Go as a Domain for AI Research: A Comparison between Go and Chess. In *Intelligent Information Systems, 1995. ANZIIS-95. Proceedings of the Third Australian and New Zealand Conference*, pp. 181–186. IEEE.

Camerer, C. F. (2003). *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton: Princeton University Press.

Camerer, C. F. (2007). Neuroeconomics: Using Neuroscience to Make Economic Predictions. *The Economic Journal 117*(519), C26–C42.

Camerer, C. F., G. Loewenstein, and M. Rabin (Eds.) (2004). *Advances in Behavioral Economics*. Princeton: Princeton University Press.

Chase, W. G. and H. A. Simon (1973a). The Mind's Eye in Chess. In W. G. Chase (Ed.), *Visual Information Processing. Proceedings of the Eighth Annual Carnegie Symposium on Cognition, Held at the Carnegie-Mellon University Pittsburgh, Pennsylvania, May 19, 1972*. London: Academic Press.

Chase, W. G. and H. A. Simon (1973b). Perception in Chess. *Cognitive Psychology 4*(1), 55–81.

Chomsky, N. (1956). Three Models for the Description of Language. *IRE Transaction on Information Theory 2*(3), 113–124.

Chomsky, N. (1959). On Certain Formal Properties of Grammars. *Information and Control 2*(2), 137–167.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics 58*(2), 345–363.

Church, A. (1938). The Constructive Second Number Class. *Bulletin of the American Mathematical Society 44*(4), 224–232.

Cook, S. A. (1971). The Complexity of Theorem-proving Procedures. In *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, pp. 151–159.

Craik, F. I. M. and R. S. Lockhart (1972). Levels of Processing: A Framework for Memory Research. *Journal of Verbal Learning and Verbal Behavior 11*(6), 671–684.

Davis, J. B. (2012). Economics Imperialism under the Impact of Psychology: The Case of Behavioral Development Economics. Working paper series, Social Science Research Network (SSRN). Marquette University and University of Amsterdam.

Davis, M. (1958). *Computability and Unsolvability*. New York: McGraw-Hill Book Company.

de Bondt, W. F. M. and R. H. Thaler (1989). Anomalies: A Mean-Reverting Walk Down Wall Street. *The Journal of Economic Perspectives 3*(1), 189–202.

de Finetti, B. (1937 (1964)). Foresight: Its Logical Laws, Its Subjective Sources. *Annales de l'Institut Henri Poincaré, translated by Henery E. Kybrug, Jr ; reprinted in Studies in Subjective Probability, edited by Henry E. Kyburg, Jr. and Howard E. Smokler 7*, 93–158. Translated by Henery E. Kybrug, Jr.

de Groot, A. (1965). *Thought and Choice in Chess*. The Hague, The Netherlands: Mouton.

de Groot, A. and F. Gobet (1996). *Perception and Memoery in Chess*. Assen: Van Gorcum.

Devlin, K. (1988). *Mathematics: The New Golden Age*. New York: Columbia University Press.

Edwards, W. (1954). The Theory of Decision Making. *Psychological Bulletin 51*(4), 380–417.

Edwards, W. (1961). Behavioral Decision Theory. *Annual Review of Psychology 12*(1), 473–498.

Edwards, W. (1962). Subjective Probabilities Inferred From Decisions. *Psychological Review 69*(2), 100–135.

Erbach, D. (1989). Report from the Workshop on New Directions in Game-tree Search. *Computer Go 10*, 8–9.

Ericsson, K. A. and H. A. Simon (1980). Verbal Reports as Data. *Psychological Review 87*(3), 215–251.

Ericsson, K. A. and H. A. Simon (1993). *Protocoal Analysis: Verbal Reports as Data* (Revised ed.). Cambridge, MA: A Bradford Book, The MIT Press.

Feferman, S. (2009). Gödel, Nagel, minds, and machines. *Journal of Philosophy CVI*(4), 201–219.

Feigenbaum, E. A. (1959). An Information Processing Theory of Verbal Learning. Technical Report P-1817, The Rand Corporation, Santa Monica, California, USA.

Feigenbaum, E. A. and H. A. Simon (1962). A Theory of The Serial Position Effect. *The British Journal of Psychology 53*(3), 307–320.

Feigenbaum, E. A. and H. A. Simon (1984). EPAM-like Models of Recognition and Learning. *Cognitive Science 8*(4), 305–336.

Froot, K. A. and R. H. Thaler (1990). Anomalies: Foreign Exchange. *The Journal of Economic Perspectives 4*(3), 179–192.

Gigerenzer, G. and R. Selten (Eds.) (2001). *Bounded Rationality: The Adaptive Toolbox*. Cambridge, MA: The MIT Press.

Glimcher, P. W., M. C. Dorris, and H. M. Bayer (2005). Physiological Utility Theory and the Neuroeconomics of Choice. *Games and Economic Behavior 52*(2), 213–256.

Gobet, F., A. de Voogt, and J. Retschitzki (2004). *Moves in Mind: the Psychology of Board Games*. New York: Psychology Press, Taylor & Francis Group.

Gobet, F. and H. A. Simon (1996). Templates in Chess Memory: A Mechanism for Recalling Several Boards. *Cognitive Psychology 31*(1), 1–40.

Goodwin, R. M. (1947). Dynamical Coupling with Especial Reference to Markets Having Production Lags. *Econometrica 15*(3), 181–203.

Guetzkow, H. and H. A. Simon (1955). The Impact of Certain Communication Nets upon Organization and Performance in Task-Oriented Groups. *Management Science 1*(3-4), 233–250.

Hasan, H. and L. Warne (2008). Go*Team Experimentation Results: Research, Train and Sustain. Human Dimensions of NCW Sub-task Report. Technical report, Joint Operations Division, DSTO Defence Science and Technology Organisation, Australia.

Hayek, F. A. (1945). The Use of Knowledge in Society. *The American Economic Review 35*(4), 519–530.

Heukelom, F. (2012). A Sense of Mission: The Alfred P. Sloan and Russell Sage Foundations' Behavioral Economics Program, 1984-1992. *Science in Context 25*(2), 263–286.

Hommes, C. H. (2006). Heterogeneous Agent Models in Economics and Finance. In L. Tesfatsion and K. L. Judd (Eds.), *Handbook of Computational Economics*, Volume 2, Chapter 23, pp. 1109–1186. Amsterdam: Elsevier B.V.

Howson, C. (2009). Logic and Finite Additivity: Mutual Supporters in Bruno de Finetti's Probability Theory. In M. C. Galavotti (Ed.), *Bruno de Finetti: Radical Probabilitist*, pp. 41–58. London: College Publicaions.

Kahneman, D., J. L. Knetsch, and R. H. Thaler (1991). Anomalies: The Endowment Effect, Loss Aversion, and Status Quo Bias. *The Journal of Economic Perspectives 5*(1), 193–206.

Kahneman, D. and R. H. Thaler (2006). Anomalies: Utility Maximization and Experienced Utility. *The Journal of Economic Perspectives 20*(1), 221–234.

Kahneman, D. and A. Tversky (1979). Prospect Theory: an Analysis of Decisions Under Risk. *Econometrica 47*(2), 313–327.

Kao, Y.-F., V. Ragupathy, K. V. Velupillai, and S. Zambelli (2012). Noncomputability, Unpredictability, Undecidability and Unsolvability in Economics and Finance Theories. *Complexity 18*(1), 51–55.

Kao, Y.-F. and K. V. Velupillai (2012a). Herbert Simon. In G. Faccarello and H. Kurz (Eds.), *Handbook of the History of Economic Analysis*,. Cheltenhem: Edward Elgar. forthcoming.

Kao, Y.-F. and K. V. Velupillai (2012b). Origins and Pioneers of Behavioural Economics. *Interdisciplinary Journal of Economics and Business Law 1*(3), 47–73.

Kao, Y.-F. and K. V. Velupillai (2013). Behavioural Economics: Classical and Modern. *The European Journal of the History of Economic Thought*. forthcoming.

Khachiyan, L. G. (1979). A Polynomial Algorithm in Linear Programming. *Doklady Akademiia Nauk SSSR 244:S*, 1093–1096. translated in Soviet Mathematics Doklady 20:1 (1979), p. 191-194.

Kleene, S. C. (1936a). General Recursive Functions of Natural Numbers. *Mathematische Annalen 112*(1), 727–742.

Kleene, S. C. (1936b). λ-definability and Recursiveness. *Duke Mathematical Journal 2*(2), 340–353.

Lamont, O. A. and R. H. Thaler (2003). Anomalies: The Law of One Price in Financial Markets. *The Journal of Economic Perspectives 17*(4), 191–202.

LeBaron, B. (2006). Agent-Based Computational Finance. In L. Tesfatsion and K. L. Judd (Eds.), *Handbook of Computational Economics*, Chapter 24, pp. 1187–1233. Amsterdam: Elsevier B.V.

Lee, C. M. C., A. Shleifer, and R. H. Thaler (1990). Anomalies: Closed-End Mutual Funds. *The Journal of Economic Perspectives 4*(4), 153–164.

Lichtenstein, D. and M. Sipser (1980). GO is Polynomial-Space Hard. *Journal of the Association for Computing Machinery 27*(2), 393–401.

Ljungqvist, L. and T. J. Sargent (2004). *Recursive Macroeconomic Theory*. Cambridge, MA: The MIT Press.

Loewenstein, G. and R. H. Thaler (1989). Anomalies: Intertemporal Choice. *The Journal of Economic Perspectives 3*(4), 181–193.

March, J. G. (1978). Bounded Rationality, Ambiguity and the Engineering of Choice. *The Bell Journal of Economics 9*, 587–608.

Matiyasevich, Y. (1994). *Hilbert's Tenth Problems*. Cambridge MA: The MIT Press.

McCall, J. J. (1970). Economics of Information and Job Search. *The Quarterly Journal of Economics 84*(1), 113–126.

Miller, F. D. (1984). Aristotle on rationality in action. *The Review of Metaphysics 37*(3), 499–520.

Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review 63*(2), 81–97.

Minsky, M. (1975). A Framework for Representing Knowledge. In P. H. Winston (Ed.), *The Psychology of Computer Vision*, pp. 211–277. New York: McGraw-Hill Book Company.

Moret, B. M. and H. D. Shapiro (1991). *Algorithms from P to NP: Volume I Design & Efficiency*, Volume I. Boston: Addison-Wesley.

Moriarty, K. (1996). The Meaning of Life: Extensional Semantics in the Game of Go. In I. Rauch and G. F. Carr (Eds.), *Semiotics around the World: Synthesis in Diversity*, Volume 1. Berlin: Mouton de Gruyter, Walter de Gruyter & Co.

Nagahara, Y. (1972). *Strategic Concept of Go*. Japan: The Ishi Press Go Series.

Nam, C. (2004). *Contemporary Go Terms: Definitions & Translations*. Seoul, Korea: Oromedia Publishing Co.

Nelson, R. R. and S. G. Winter (1977). In Search of Useful Theory of Innovation. *Research Policy 6*, 36–76.

Nelson, R. R. and S. G. Winter (2002). Evolutionary theorizing in economics. *The Journal of Economic Perspectives 16*(2), 23–46.

Newell, A., J. C. Shaw, and H. A. Simon (1958a). Elements of the Theory of Human Problem Solving. *Psychological Review 65*(3), 151–66.

Newell, A., J. C. Shaw, and H. A. Simon (1958b). The Processes of Creative Thinking. Technical report, Rand Corporation, Santa Monica, CA.

Newell, A. and H. A. Simon (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, INC.

Newell, A. and H. A. Simon (1976). Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM 19*(3), 113–126.

Polya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method*. Princeton: Princeton University Press.

Post, E. L. (1947). Recursive Unsolvability of a Problem of Thue. *The Journal of Symbolic Logic 12*(1), 1–11.

Ramsey, F. P. (1978 (1926)). Truth and Probability. In D. H. Mellor (Ed.), *The Foundations of Mathematics and Other Logical Essays*, pp. 58–100. London: Routledge and Kegan Paul.

Reitman, J. S. (1976). Skilled Perception in Go: Deducing Memory Structures from Inter-Response Times. *Cognitive Psychology 8*(3), 336–356.

Reitman, W., J. Kerwin, R. Nado, J. Reitman, and B. Wilcox (1974). Goals and Plans in a Program of Playing Go. In *ACM' 74 Proceedings of the 1974 annual conference*, Volume 1. New York: ACM.

Reitman, W. and B. Wilcox (1975). Perception and Rrepresentation of Spacial Relations in a Program for Playing Go. In *ACM' 75 Proceedings of the 1975 annual conference*, pp. 37–41. New York: ACM.

Roberts, R. B. and I. P. Goldstein (1977). The FRL Manual. Technical report, Massachusetts Institute of Technology Artificial Intelligence Laboratory.

Robson, J. M. (1983). The Complexity of Go. In R. E. A. Mason (Ed.), *Information Processing 83, IFIP Congress 1983*, Paris, pp. 413–417. Amsterdam: Elsevier Science Publisers B.V.

Rubinstein, A. (2008). Comments on Neuroeconomics. *Economics and Philosophy 24*(3), 485–494.

Rustichini, A. (2005). Neuroeconomics: Present and Future. *Games and Economic Behavior 52*(2), 201–212.

Saito, Y. and A. Yoshikawa (1995). Do Go Players Think in Words? Interim Report of the Analysis of Go Players Protocols. In H. Matsubara (Ed.), *Proceedings of the Second Game Programming Workshop in Japan*, Kanagawa, pp. 118–127. Computer Shogi Association.

Saito, Y. and A. Yoshikawa (1996). An Analysis of Strong Go-players' Protocol. In H. Matsubara (Ed.), *Proceedings of the Third Game Programming Workshop in Japan*, Kanagawa, pp. 66–75. Computer Shogi Association.

Saito, Y. and A. Yoshikawa (2000). Go as Testbed for Cognitive Sciences Studies. In H. van den Herik and H. Iida (Eds.), *Games in AI Research*, pp. 252–269. The Netherlands: Van Spijk, Venlo.

Savage, L. J. (1954). *The Foundations of Statistics*. New York: John Wiley & Sons.

Savitch, W. J. (1970). Relationships Between Nondeterministic and Deterministic Tape Complexities. *Journal of Computer and System Sciences 4*(2), 177–192.

Schumpeter, J. (1954). *History of Economic Analysis*. London: George Allen & Unwin Ltd.

Shannon, C. E. (1950). XXII. Programming a Computer for Playing Chess. *Philosophical Magazine 41*(314), 256–275.

Shapiro, S. (2003). Mechanism, Truth, and Penrose's New Argument. *Journal of Philosophical Logic 32*(1), 19–42.

Siegel, J. J. and R. H. Thaler (1997). Anomalies: The Equity Premium Puzzle. *The Journal of Economic Perspectives 1*(1), 191–200.

Simon, H. A. (1944). Decision-Making and Administrative Organization. *Public Administration Review 4*(1), 16–30.

Simon, H. A. (1947). *Administrative Behavior* (Second ed.). New York: The Free Press.

Simon, H. A. (1952). On the Definition of the Causal Relation. *The Journal of Philosophy 49*(16), 517–528.

Simon, H. A. (1953a). A Behavioral Model of Rational Choice. Technical report, The RAND Corporation, Santa Monica, CA.

Simon, H. A. (1953b). Casual Ordering and Indentifiability. In W. C. Hood and T. Koopmans (Eds.), *Studies in Econometric Methond*. New York: John Wiley & Sons, Inc.; London: Chapman & Hall, Limited.

Simon, H. A. (1955). A Behavioral Model of Rational Choice. *Quarterly Journal of Economics 69*(1), 99–118.

Simon, H. A. (1956). Rational Choice and the Structure of the Environment. *Psychological Review 63*(2), 129–38.

Simon, H. A. (1957). *Models of Man*. New York: John Wiley & Sons, Inc.

Simon, H. A. (1959). Theories of Decision-Making in Economics and Behavioral Science. *The American Economic Review 49*(3), 253–283.

Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society 106*(6), 467–482.

Simon, H. A. (1967). Motivational and Emotional Controls of Cognition. *Psychological Review 74*(1), 29–39.

Simon, H. A. (1972). Theories of Bounded Rationality. In C. B. McGuire and R. Radner (Eds.), *Decision and Organization*, pp. 161–176. Amsterdam: North-Holland.

Simon, H. A. (1973). The Organization of Complex Systems. In H. H. Pattee (Ed.), *Hierarchy Theory - The Challenge of Complex Systems*. New York: George Braziller.

Simon, H. A. (1974). How Big is a Chunk? Complex Information Processing Papers 242, Carnegie Mellon University, Pittsburgh, PA.

Simon, H. A. (1975). The Functional Equivalence of Problem Solving Skill. *Cognitive Psychology 7*(2), 268–288.

Simon, H. A. (1976). From Substantive to Procedural Rationality. In S. J. Latsis (Ed.), *Method and Appraisal in Economics*, pp. 129–148. Cambridge: Cambridge University Press.

Simon, H. A. (1978a). Information-Processing Theory of Human Problem Solving. In W. K. Estes (Ed.), *Handbook of Learning and Cognitive Processes*, Volume 5, pp. 271–295. Mahwah, NJ: Lawrence Erlbaum.

Simon, H. A. (1978b). On How to Decide What to Do. *The Bell Journal of Economics 9*(2), 494–507.

Simon, H. A. (1979). Rational Decision Making in Business Organization. *The American Economic Review 69*(4), 493–513.

Simon, H. A. (1983a). Discovery, Invention, and Development: Human Creative Thinking. *Proceedings of the National Academy of Sciences of the United States of America 80*(14), 4569–4571.

Simon, H. A. (1983b). *Reason in Human Affairs*. Oxford: Basil Blackwell.

Simon, H. A. (1987). Causality in Economic Models. In J. Eatwell, M. Milgate, and P. Newman (Eds.), *The New Palgrave: A Dictionary of Economics*, Volume 1, pp. 382–383. London: Macmillan Press Ltd.

Simon, H. A. (1990a). Invariants of Human Behavior. *Annual Review of Psychology 41*(1), 1–19.

Simon, H. A. (1990b). Prediction and Prescription in Systems Modeling. *Operations Research 38*(1), 7–14.

Simon, H. A. (1991a). Bounded Rationality and Organizational Learning. *Organization Science 2*(1, Special Issue: Organizational Learning: Papers in Honor of (and by) James G. March), 125–134.

Simon, H. A. (1991b). Cognitive Architectures and Rational Analysis: Comment. In K. VanLehn (Ed.), *Architectures for Intelligence: The 22nd Carnegie Mellon Symposium on Cognition*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., Publishers.

Simon, H. A. (1991c). *Models of My Life*. Cambridge, MA: The MIT Press.

Simon, H. A. (1993). Strategy and Organizational Evolution. *Strategic Management Journal 14*(Special Issue: Organizations, Decision Making and Strategy), 131–142.

Simon, H. A. (1995). Near Decomposability and Complexity: How a Mind Resides in a Brain. In *The Mind, the Brain, and Complex Adaptive System. Santa Fe Institiue Studies in the Sciences of Complexity*, Volume XXII of *Santa Fe Institiue Studies in the Sciences of Complexity*. Boston: Addison-Wesley.

Simon, H. A. (1996a). Machine as Mind. In P. Macmillan and A. Clark (Eds.), *Machines and Thought - The Legacy of Alan Turing*, Volume 1, Chapter 5, pp. 81–101. Oxford: Oxford University Press.

Simon, H. A. (1996b). *The Science of the Artificial* (3rd ed.). Cambridge, MA: The MIT Press.

Simon, H. A. (1997). *An Empirically Based Microeconomics - The Raffaele Mattioli Lectures*. Cambridge: Cambridge University Press.

Simon, H. A. (2001). Science Seeks Parsimony, Not Simplicity: Searching for Pattern in Phenomena. In A. Zellner and H. A. Keuzenkamp (Eds.), *Simplicity, Inference and Modelling: Keeping it Sophisticatedly Simple*. Cambridge: Cambridge University Press.

Simon, H. A. (2002). Near Decomposability and the Speed of Evolution. *Industrial and Cooperate Change 11*(3), 587–599.

Simon, H. A. and A. Ando (1961). Aggregation of Variables in Dynamic Systems. *Econometrica 29*(2), 111–138.

Simon, H. A. and M. Barenfeld (1969). Information-processing Analysis of Perceptual Process in Problem Solving. *Psychological Review 76*(5), 473–483.

Simon, H. A. and K. Gilmartin (1973). A Simulation of Memory for Chess Positions. *Cognitive Psychology 5*(1), 29–46.

Simon, H. A. and Y. Iwasaki (1988). Casual Ordering, Comparative Statics, and Near Decomposability. *Journal of Econometrics 39*(1), 149–173.

Simon, H. A. and K. Kotovsky (1963). Human Acquisition of Concepts for Sequential Patterns. *Psychological Review 70*(6), 534–546.

Simon, H. A. and A. Newell (1971). Human Problem Solving: The State of the Theory in 1970. *American Psychologist 26*(2), 145–159.

Simon, H. A. and N. Rescher (1966). Cause and Counterfactual. *Philosophy of Science 33*(4), 323–340.

Simon, H. A. and J. Schaeffer (1992). The Game of Chess. In R. J. Aumann and S. Hart (Eds.), *Handbook of Game Theory with Economic Application*, Volume 1. Amsterdam: Elsevier Science Publishers.

Sipser, M. (1997). *Introduction to the Theory of Computation*. Boston: PWS Publishing Company.

Smith, V. (1976). Experimental Economics: Induced Value Theory. *The American Economic Review 66*(2), 274–279.

Stokey, N. L. and R. E. Lucas (1989). *Recursive Methods in Economic Dynamics*. Cambridge, MA: Harvard University Press. (with collaboration of Edward C. Prescott).

Thaler, R. (1987a). Anomalies: Seasonal Movements in Security Prices II: Weekend, Holiday, Turn of the Month, and Intraday Effects. *The Journal of Economic Perspectives 1*(2), 169–177.

Thaler, R. H. (1987b). Anomalies: The January Effect. *The Journal of Economic Perspectives 1*(1), 197–201.

Thaler, R. H. (1990). Anomalies: Saving, Fungibility, and Mental Accounts. *The Journal of Economic Perspectives 4*(1), 193–205.

Thaler, R. H. (Ed.) (2005). *Advances in Behavioral Finance*. Princeton: Princeton University Press.

Tourlakis, G. J. (1984). *Computability*. Virginia: Reston Publishing Company, Inc.

Turing, A. (1953). Digital Computers Applied to Games. Manuscript, Modern Archive Centre, King's College, Cambridge.

Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society 42*(2), 230–265.

Turing, A. M. (1937). Computability and λ-definability. *The Journal of Symbolic Logic 2*(4), 153–163.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind 59*(236), 433–460.

Turing, A. M. (1951). Intelligent Machinery, A Heretical Theory. *Philosophia Mathematica 4*(3), 256–260.

Turing, A. M. (1952). The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society, Series B, Biological Sciences 237*(641), 37–72.

Turing, A. M. (1954). Solvable and Unsolvable Problems. *Science News 31*, 7–23.

Turing, S. (2012 (1959)). *Alan M. Turing: Centenary Edition*. Cambridge, UK: Cambridge University Press.

Tversky, A. (1972). Elimination by Aspects - A Theory of Choice. *Psychological Review 79*(4), 281–299.

Tversky, A. and D. Kahneman (1992). Advances in Prospect Theory: Cumulative Representation of Uncertainty. *Journal of Risk and Uncertainty 5*(4), 297–323.

Tversky, A. and R. H. Thaler (1990). Anomalies: Preference Reversals. *The Journal of Economic Perspectives 4*(2), 201–211.

Velupillai, K. V. (2000). *Computable Economics*. Oxford: Oxford University Press.

Velupillai, K. V. (2010a). *Computable Foundations for Economics*. London: Routledge.

Velupillai, K. V. (2010b). Foundations of Boundedly Rational Choice and Satisficing Decisions. *Advances in Decision Sciences 2010*, 16 pages.

Velupillai, K. V. and Y.-F. Kao (2013). Computable and Computational Complexity Theoretic Bases for Herbert Simons Cognitive Behavioral Economics. *Cognitive System Resarch*. forthcoming.

Velupillai, K. V. and S. Zambelli (2011). Computing in Economics. In J. Davis and W. Hands (Eds.), *The Elgar Companion to Recent Economic Methodology*. Cheltenham: Edward Elgar Publishing.

Wang, H. (1970). *Logic, Computers, and Sets*. New York.: Chelsea Publishing Company.

Yoshikawa, A., T. Kojima, and Y. Saito (1999). Relations between Skill and the Use of Terms - An Analysis of Protocols of the Game of Go. In H. J. van den Herik and H. Iida (Eds.), *Computers and Games (CG'98), Lecture Notes in Computer Science 1558*. Berlin: Springer-Verlag.

Zobrist, A. L. (1969). A Model of Visual Organization for the Game of Go. In *AFIPS '69 (Spring) Proceedings of the May 14-16, 1969, spring joint computer conference*, pp. 103–112. New York: ACM.