Algorithmic Entropy, Phase Transition, and Smart Systems

E.V. Krishnamurthy

Computer Sciences Laboratory,
Australian National University, Canberra, ACT 0200, Australia
`abk@discus.anu.edu.au`

Abstract. A smart system exhibits the three important properties: (i) interactive, collective, coordinated and parallel operation (ii) self-organization through emergent properties (iii) adaptive and flexible operation. A hierarchy based on metric entropy is suggested among the computational systems that transcend from the unsmart to the smart system through a phase transition like phenomenon. Understanding smart systems is useful to solve hard-optimization problem inspired by the self-organizing processes found in nature. Such systems will be valuable to create artificial systems made up of exotic matter to solve specific problems in particular domains of interest with a high efficiency.

1 Introduction

Smart systems have no formal definitions, although such a system seems to be a precursor to the living system, Brooks [4,5]. As such no suitable formal model is available for smart systems. Therefore, we define the smart systems as those systems having the following important properties:

1. Interactive, Collective, Coordinated, and Highly Efficient Parallel Operation
They can interact with the environment (hence called open). Also they collectively and cooperatively perform actions, coordinating their actions when there is competition, to obtain maximal efficiency.

2. Self Organization and Emergence
The total dynamic behaviour of the system cannot be inferred from the dynamic behaviour of its components and new properties emerge abruptly. These new properties of the system are not predictable, in advance from the properties of the individual interactions. In particular, under emergence, the many degrees of freedom arising due to its component parts collapse into a fewer new ones with a smaller number of globally relevant parameters; see Makishima [27].

3. Adaptive, Fault-Tolerant, and Flexible Operation
Smart systems are always flexible to change-they can modify their past behaviour by a learning process and adapt to environmental changes, available resources, as well as, tolerating failures or non-cooperation of some of their components.

Our purpose in this paper is to restrict our consideration to the following issues:
1. Is the behaviour of smartness analogous to the critical phenomenon (phase transition or percolation)? Can we obtain suitable parameters to describe this phenomenon?

2. Is there a hierarchy of degree of smartness among computational systems?
3. Can we simulate smart systems to solve intractable problems?

2   Conrad's Principles

Before we answer the above questions, we recall the principles proposed by Conrad [7]. These have a direct bearing on defining smart systems. According to Conrad, a general purpose computing system cannot have all of the three following properties:

1. Structural programmability (algorithmizability)
2. High computational efficiency
3. High evolutionary adaptability and flexibility

Properties 1 and 2 are mutually exclusive. That is, we cannot have high computational efficiency in a programmable system. Properties 1 and 3 are mutually exclusive in the region where maximum computational efficiency exists. That is, whenever maximal computational efficiency is required, the system should be highly adaptive rather than structurally programmable (algorithmic). Based on Conrad's principles, we can say that self-organization is not possible in an algorithmizable system.

A further support to Conrad's principles is provided by Wegner [35]. Wegner argues that the algorithmic machines are less powerful than machines that can interact with the environment and such interacting systems turn out to be much more efficient in coupling material resources to problem solving. As we know, efficient coupling to environment makes enormous difference for problems such as pattern recognition, survival etc. As a further support to Wegner's hypothesis, in this paper we will show that such interaction introduces positive entropy into the system to turn a rigid algorithmic system into a smart system. An example of a smart system is visual perception which is computationally most intensive. While a digital computer makes a large number of sequential decisions, the biological visual system uses a holistic approach, collective or Gestalt .The system makes use of its computational resources very efficiently, whether it is measured by speed of calculation or energy considerations. The computations are resistant to damage to hardware. Also emergent properties - such as self-awareness seems to be present. Further, there is no structural hardware diagram or equivalently a structured program that does it, Haken [15], Hameroff [16], Hopfield [20].

3   Phase Transition Model for Smart System

As mentioned in the introduction, smart systems need to possess the three properties:

1. Interactive, collective and parallel operation
2. Self organization through emergence
3. Adaptive and flexible operation

Properties 1 and 2 hold near the critical point in a phase transition in a physical system [9] or in a percolation model, Stauffer and Aharony [33], near the percolation threshold. Note that the percolation model is more general than the phase transition model, since it deals with the more abstract geometric properties rather than the thermodynamic properties used in phase transition models. These two models lie at the

border of order and disorder and are concerned with the collective and parallel inter-action among the microscopic objects that result in scale invariance and universality. Scale invariance corresponds to a power law behaviour over a wide range of control parameter; the exponent involved in this power law is called the critical exponent.

By universality, we mean that the set of exponents found in diverse systems group themselves into distinct classes, with the property that all systems falling in the same class have the same exponents. This suggests that there are common features among the underlying microscopic mechanisms responsible for the observed scale invariant behaviour. The idea of universality is that apparently dissimilar systems show consid-erable similarities near their critical points. Accordingly, all percolation/ phase transi-tion problems can be divided into a small number of different classes depending upon the dimensionality of the system and the symmetries of the order state. Within each class, all phase transitions have identical behaviour in the critical region, only the names of the variables are changed. Thus universality describes the relationship among different phase transitions.

Also percolation / phase transition models reflect the cooperative, as well as, com-petitive behaviour among the microscopic objects. The modelling uses a suitable geometric structure with a local computation that results in a global change. Further, the percolation / Phase transition systems have the property 3, namely, adaptive and flexible behaviour to tolerate failures, since they both deal with the formation of clus-ters or creating paths among distant neighbours, even if some of these neighbours are non-cooperative. This ultimately can lead to an emergent behaviour through self-organized criticality.

From the above arguments we see that to model a smart system, the phase transi-tion or percolation model will be of value since the smart system seems to lie between order and disorder.This observation has been made earlier by Langton [24], Wolfram [31], who suggest that life exists at the edge of chaos, and by Zak et al [37] who sug-gest that instability is necessary to create intelligence.

Prigogine [30,31] emphasises why non-unitary evolution based on star- hermitean operator is needed to deal with open systems, as well as, systems which are far-from equilibrium and attain stability. Prigogine emphasises that the irreversible (non-unitary) process play a fundamental role in biological systems. According to Prigog-ine, future and past play the same role in time- reversible unitary systems; hence they cannot explain the emergence of new dynamical patterns involved in the intelligent or smart biological systems.

Hence, the three required properties of smart systems are difficult to realise within the unitary transformation or time reversible evolutionary systems. It looks as though hysteresis, long term memory and time arrow have a direct role in turning the systems smarter. In fact, it is due to the time arrow and memory, the smart system remembers and orders the temporal events as earlier and later, without the explicit sequential addressing mode used in conventional programming. This leads to a kind of self-awareness of past, present and future leading to a psychological time arrow. Break-down of time symmetry seems essential for constructing special purpose systems made up of exotic phases of matter that can function similar to the Nature beyond the computable domain.

4 Role of Metric and Algorithmic Entropy

Metric entropy plays a similar role like the ordinary entropy does in thermodynamics or information theory to measure the disorder in a system. Metric entropy is defined as the average information per measurement obtained from an (denumerable) infinite sequence of finite precision – identical measurements made on a time evolving system from a time minus infinity to plus infinity. If the metric entropy is zero it implies that the measurement sequence, begun in the remote past, ceases to provide any more additional information after a finite time T. This means adequate information has been acquired to predict the properties of that system completely and unambiguously. This also means that the information gained by measurement equals the entropy decrease in the system so that no uncertainty remains. However, if the metric entropy is positive this means the entropy in the system has not decreased as much as the information obtained by measurement and still the uncertainty remains. It can also be due to the fact that the system is producing entropy faster than what we can measure.

Metric entropy is related to another entropy called "algorithmic entropy" or Kolmogorov-Chaitin entropy, Chaitin [6]. Unlike in ordinary entropy, in algorithmic entropy we use a rule or a program as the basis for the creation of order or disorder in objects that are created from a given input. We then express the order and disorder of the created objects in terms of the program length and the length of the output it generates. For this purpose, we measure the bit-length of the minimal program (or functional rule) that generates a required output sequence from a given input sequence.

As an example, consider the creation of a totally erratic infinite sequence. This sequence cannot be described as an output sequence of some input sequence by using a simple program whose length is smaller than the desired output length. Analogously, whenever the predictions of a physical theory can be obtained completely, such a theory is representable by a Turing machine [10]. The analogy between measurement procedure and algorithm is now clear. In both cases there is a well-defined input and we want a well-defined output after a finite time that can be described in finite time and space. This analogy permits us to relate metric entropy to the algorithmic entropy. Also it enables us to use the algorithmic complexity measure to describe the complexity of the measurement procedure. That is the bit-length of the minimal program that describes the functional rule to generate a required output from a given input, can be used to define the complexity of the measurement procedure, as well as, algorithms.

We can define the algorithmic entropy measure thus: Let $K(n)$ define the bit length of a minimal program that can output any n bit finite subsequence using a functional rule. Also let $K = \text{Lim } n \rightarrow \infty [K(n)/n]$. We note that $0 \leq K \leq 1$. Thus when $K > 0$ the length of the program required to generate a desired output sequence goes on increasing or the rule is not finite and turns out to be more and more complex. However, for $K = 0$ the program length is much shorter than the required output thus defining a deterministic, orderly and predictable output according to an algorithmic rule. Thus K can be used to measure metric entropy.

In general, systems with zero entropy are tractable (predictable) and those with positive entropy are not predictable. Positive entropy systems cannot be described in any simpler way than they are and there no formal grammatical rules that can be used to understand them. Well- structured objects (e.g., Context free grammars, regular grammars and serial-parallel orders) provide for easy description through functional

rules and hence have zero metric entropy. The systems with zero metric entropy are classified "Turing or algorithmically expressible". Such zero-entropy systems are amenable for accurate measurements in a finite time and their orbital sequences are algorithmically predictable leaving no uncertainty in measurement. Also, the information gained by measurement equals the entropy decrease in the system. In positive entropy systems, the rate of evolution can be faster than what we can measure to gain information.

Remark: In Gell-Mann [12], p.101, the algorithmic entropy is called **"depth"** and the metric entropy is called **"crypticity"**; the former measures the amount of (labour) time required to go from the compressed spatial information or minimal program to the full spatial description, while the latter measures the amount of labour (time) involved to compress the full spatial description into a minimal program.

4.1 Algorithmic Information and K Entropy

Consider $K = \text{Lim } t \rightarrow \infty [K(t)/t)]$; where K(t) is the algorithmic information needed to record a piece of the trajectory in the interval of time t. Thus for positive K-entropy, in the long run the recording of information for evolution increases unbounded and the evolution cannot be followed deterministically, unless a disproportionately long or infinite time is devoted to this task beyond a critical time
t = T(c). As we approach T(c), the recording is critically slowed down -much like in phase transition phenomena. At this time the motion is no longer deterministic and the forward and backward evolution are not reversible resulting in a spontaneous breakdown of time - reversal symmetry [10]. In contrast, when K= 0 the evolution can be recorded deterministically, without breaking the time reversal symmetry.
Thus a phase -transition like situation arises between the Turing expressible and Turing non-expressible systems leading to a critical point behaviour. By the universality principle used in phase transition-like phenomenon it is possible that critical exponents and scaling factors exist for such systems so that we can say when the system can become smart. We will discus this aspect again later.

4.2 Hierarchy among Machines

Based on Metric-Algorithmic entropy we can establish a hierarchy among the computational systems (See Fig. 1). That is the metric entropy seems to play an important scale-free role as in percolation or phase transition in deciding whether the system is smart. Essentially, the zero entropy machines cannot cope up with the complexity of solving a problem leading to a critical point corresponding to intractability and non-computability. At this point a phase transition like phenomena can help towards emergence of new properties resulting in positive entropy and self -organization, Bak [1]. Thus we can classify the two major classes of machines, ordinary (O) and dissipative (P) based on metric entropy as below.

O. Ordinary or Zero Metric Entropy Machines:
Completely structured, Deterministic, Exact behaviour (or Algorithmic) Machines.
This class contains: Finite State machines, Push down-stack machines, Turing Machines (Deterministic) that halt and Exactly integrable Hamilton flow machines.

Such machines are information-lossless; their outputs contain all the required information as dictated by the programs and the information gained by running the program exactly equals the entropy decrease in the program, Gell-mann [12], p.224.

P. Positive Metric Entropy Machines:

These are Partially Structured, Non-deterministic, Probabilisitic, Average behaviour, ergodic systems that preserve shape and volume in phase space , mixing systems where shape in phase space is distorted, Kolmogorov -flow (K-flow) systems with the motion in phase space unpredictable, nonequilibrium systems exhibiting macro and emergent behaviour – such as Chemical and Biological machines and living systems (Fig. 1). The positive entropy machines do not obey the unitary transformation law. Their evolution results in increasing entropy. Their actions are irreversible. That is the time-reversal symmetry is spontaneously broken. Only such machines can become smart; but they will have to experience an arrow of time. The measurements on these systems will not reveal all the required information.

4.3 Combining Zero and Positive Entropy Machines to Create Smartness

Prigogine [30,31] suggests the use of non-unitary transformation, called star-Hermitean operators to extend the capabilities of computational systems to reflect average behaviour. That is we require the tools of both equilibrium and non-equilibrium quantum statistical mechanics to create smart systems that lie at the critical point between order and disorder. We also recall from Gal Or [11] that irreversibility is essential for creating smart matter. In his book Gal-Or asserts that all processes in nature can be understood by an optimal mixture of order and disorder. Thus to create smart systems we need to combine the above two different classes of machines, Zak et al [37], Gell-Mann[12].

5   Simulating a Mixture of Entropy in Systems

A way to simulate a mixture of the zero and positive entropy machines is by choosing the mode of application and the action set of a rule-based program to be either deterministic, nondeterministic, probabilistic or fuzzy. Rule application policy in a production system [23] can be modified by:

(i) Assigning probabilities/fuzziness for applying the rule
(ii) Assigning strength to each rule by using a measure of its past success
(iii) Introducing a support for each rule by using a measure of its likely relevance to the current situation.

The above three factors provide for competition and cooperation among the different rules. In particular, the probabilistic rule system can lead to emergence and self-organized criticality. Thus, we may attempt to enlarge the capabilities of class O machines by simulating special features of class P machines - using nondeterminism, randomness, approximation, probabilities, equilibrium-statistical mechanical (e.g. simulated annealing) and non-equilibrium statistical mechanical(e.g. genetic algorithms and Ant algorithm [8]).
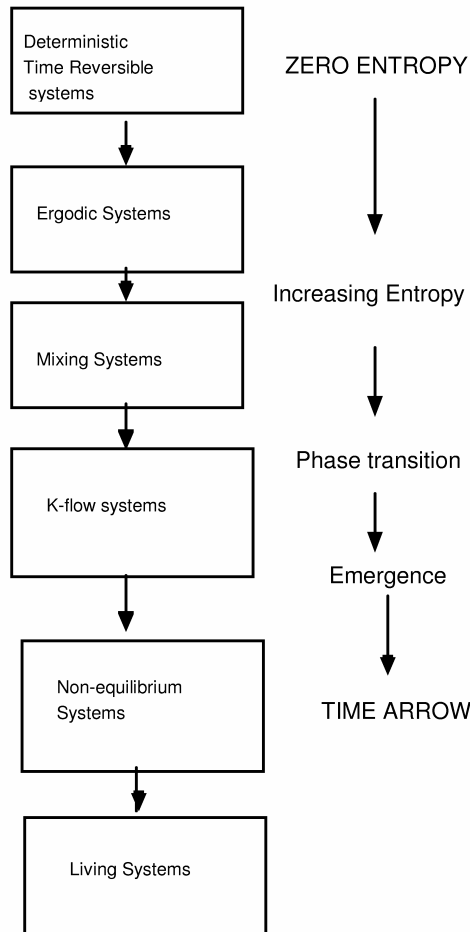
```
┌─────────────────┐
│ Deterministic   │        ZERO ENTROPY
│ Time Reversible │
│  systems        │              │
└────────┬────────┘              │
         │                       │
┌────────▼────────┐              │
│                 │              ▼
│ Ergodic Systems │
│                 │       Increasing Entropy
└────────┬────────┘
         │                       │
┌────────▼────────┐              ▼
│                 │
│ Mixing Systems  │       Phase transition
│                 │
└────────┬────────┘              │
         │                       ▼
┌────────▼────────┐
│                 │        Emergence
│ K-flow systems  │
│                 │              │
└────────┬────────┘              ▼
         │
┌────────▼────────┐        TIME ARROW
│ Non-equilibrium │
│ Systems         │
│                 │
└────────┬────────┘
         │
┌────────▼────────┐
│                 │
│ Living Systems  │
│                 │
└─────────────────┘
```

Fig. 1. Hierarchy of systems

6 Phase Transitions in AI Systems

Phase transition like situation have been observed experimentally in a wide variety of heuristics used for search problems in NP class or beyond. This phenomenon is called a knife-edge phenomenon (name given in AI); Hubermann and Hogg [21], Gent and Walsh [13] , Lau and Okagaki [25],Walsh [34], Zhang and Korf [38]. Here the over-constrained problems tend to become more constrained and under- constrained prob-lem tends to become less constrained as the search becomes deeper. In between is the knife-edge, a region in which critically constrained problems (that is those at the solvability phase transition) independently of the depth at which one examines the search tree. This knife edge phenomenon is also known in Artificial life (AL) as the Edge of chaos, Langton et al [21]. Since self- organized criticality is associated with

the phase transition, attempts are under way to exploit this feature to solve hard optimization problems. Attempts have been made to model far from equilibrium and nonequilibrium processes to solve hard optimization problem, Bak [1]. These attempts have largely been experimental, Boettcher [2], Boettcher and Percus [3]; Hubermann and Hogg [21], Pemberton and Zhang [29], Hogg and Williams[18], Hogg and Kephart [17]; however, some theoretical attempts have been made by Gomes and Selman [14], Mezard et al [28].

6.1   Simulating Nonequilibrium System: Evolutionary Optimization

In nature highly specialized complex structures emerge when their most inefficient elements are selectively driven to extinction. Evolution progresses by selecting against the few most poorly adapted species, rather than by expressly breeding those species best adapted to the environment. The experimental approach by Boettcher and Percus [3] uses the extremal optimization (EO) processes in which the least fit variables are progressively eliminated; Sneppen et al [32], Holland [19]. The EO process uses a different strategy in comparison to simulated annealing. In simulated annealing the system is forced to equilibrium dynamics by accepting or rejecting local changes. EO, however, takes the system to a far from equilibrium position and persistent selection against the worst fitness lead to near-optimal solution. Also EO differs from Genetic algorithm (GA); whereas GA keeps track of entire gene pools of states from which to select and breed an improved generation of solutions, EO operates only with local updates on a single copy of the system, with improvements achieved instead by elimination of the bad. EO also differs from the greedy strategy which aims at improving the solution at each step and as a result falls into a local optimum. EO, however, can fluctuate between good and bad solutions and can enable us to cross barriers and approach new regions in configuration space.

6.2   Boolean Satisfiability Problem and Phase Transition

Another important approach in this direction is due to Korkin [22]. Korkin considers the solution of the boolean satisfiability problem or SAT near the midpoint of the phase transition curve from satisfiable to the nonsatisfiable to illustrate the self-organization..Based on this we can create a self organized evolutionary process for solving K-SAT when the system is in nonequilibrium state near the phase transition.

6.3   Do Critical Exponents and Scaling Laws Exist for Smartness?

Although the different studies show the existence of a phase -transition like phenomena and power law relationships do hold, as yet, we cannot confidently predict the critical probability and power law exponents without experimentation for each individual complex problem. Although scaling and universality are widely accepted experimentally in many different areas (including Biology, Medicine, Physics and Social sciences and in random networks, including the World-Wide-Web), they are yet to be established mathematically. Also, practical computation for evaluating such parameters is difficult. Thus the only approach seems to be available at the present is simulation. This gives rise to various types of scaling laws depending upon the nature of the problem. Studies on nonequilibrium physics, Prigogine [31], Lebowitz [26] can help us to understand these aspects.

7 Matter and Life

Brooks [4,5] examines the more general questions than those raised in this paper, namely, the relationship between matter and life, what is that makes the matter alive and what do we lack in our understanding to create living machines. He argues that this inability to bridge matter and life may arise due to four factors:

1. Our models may not have adequately correct parameters
2. Our models may not be complex enough to bring out the distinction and may lack unimagined features of life.
3. Lack of computing power
4. Lack of "new stuff", namely mathematics and physics

Since smart system is a precursor to living system, we should expect a phase transition like situation arising to distinguish living systems from non-living systems.

8 Conclusion

We described some important properties that a smart system need to posses and the entropy conditions required for an ordinary system to become smart resembling a phase-transition. We also a presented a review of the current attempts to create smart heuristics by using the knife-edge phenomenon encountered in solving intractable problems. Although, percolation model is suitable for understanding a smart system, we are unable to obtain quantitative parameters, e.g., scaling laws and critical exponents that can help us distinguish a smart system from unsmart systems. Computational science along with Physics, Biology and Mathematics will play a major role in the development of a new science as envisaged by Wolfram [36].

References

1. Bak, P.: How Nature works: The Science of self-organized criticality, Springer Verlag, New York (1996)
2. Boettcher, S.: Extremal Optimization of graph partitioning at the percolation threshold, J.Phys A.: Math.Gen **32**, (1999) pp. 5201–5211.
3. Boettcher. S and A. Percus, A.: Nature's way of Optimizing, Artificial Intelligence, **119** (2000) 275–286.
4. Brooks, R.: The relationship between matter and life, Nature, **409** (2001) 409–411.
5. Brooks, R.: Steps towards living machines, pp. 72–94, in Lecture Notes in Computer Science, Vol. **2217**, Springer Verlag, New York (2001) 72–94,
6. Chaitin, G. "Algorithmic Information Theory*, M.I.T Press, Cambridge, Mass (1987)
7. 7. Conrad, M.: Molecular Computing, Advances in Computers, **31**, Academic Press, New York (1990). 235–325
8. 8. Dorigo, M et al.: Antalgorithms, Lecture Notes in Computer Science, **2463** Springer Verlag, NewYork (2002)
9. Emch, G and Liu, C: The Logic of Thermostatistical Physics, Springer, New York, (2002)
10. Faisal, F.H.M: Quantum Chaos, Algorithmic Paradigm and Irreversibility, in Quantum Future, Ph. Blanchard and A. Jadczyk (ed), Springer Verlag, New York (1997), 47–57
11. Gal-Or, B: Cosmology, physics and philosophy, Springer Verlag, New York (1981)
12. Gell-Mann, M: The Quark and the Jaguar, W.H. Freeman, New York (1994).

13. Gent, I and Walsh, T: Phase transitions from real computational problems, Proc. 8th Symposium on AI, (1996) 356–364.
14. Gomes, C.P and Selman, B., Satisfied with Physics, Science, **297** (2002) 784–785.
15. 15 . Haken, H.: Synergetic Computers and Cognition, Springer Verlag, New York (1991)
16. Hameroff, S.R,: Ultimate Computing, North Holland, Amsterdam (1987)
17. Hogg,T and Kephart, J.O.: Phase transitions in higher dimensional pattern classification, Computer Systems Science and Engineering, **5** (1990**)** 223–232.
18. Hogg, T and Williams, C.P : The hardest constraint problem: a double phase transition, Artificial Intelligence, **69** (1994) 359–377
19. Holland, J.H.: Emergence-chaos to order, Addison Wesley, Reading, Mass. (1998)
20. Hopfield, J: Physics, Biological Computation and Complementarity, in The lesson of quantum theory*,* North Holland, Amsterdam (1986) 295–314.
21. Hubermann, B.A. and Hogg T .: Phase transitions in Artificial Intelligence Systems, Artificial Intelligence, **33** (1987) 155–171.
22. Korkin, M.: Self-organized Evolutionary Process in sets of interdependent variables near the mid point of phase transitions in K-satisfiability, Lecture Notes in Computer Science, **2210**, Springer Verlag, New York (2001), 225–235
23. Krishnamurthy, E.V and Krishnamurthy,V.: Rule-based Programming paradigm: A formal basis for biological, chemical and Physical computation, Biosystems. **49** (1999) 205–228
24. Langton, C.E et al: Life at the Edge of chaos, in Artificial Life II, Addison Wesley, Reading,Mass., (1992) 41–91
25. Lau, M and Okagaki, T.: Applications of Phase transition theory in visual recognition and classification, Journal of visual communication and Image representation, **59**(1994) 88–94.
26. Lebowitz, J.L.: Microscopic Origins of irreversible Macroscopic Behaviour, Physica **A 263** (1999).516–527
27. Makishima, S. : Pattern Dynamics: A theory of Self-Organization, Kodansha Scientific, Tokyo (2001)
28. Mezard, M., Parisi, G., and Zecchina, R.:Analytic and algorithmic solution of random satisfiability problems, Science, **297** (2002) 812–815.
29. Pemberton, J and Zhang,W (1996); Epsilon Transformation: Exploiting phase transitions to solve combinatorial optimization problems*,* Artificial Intelligence, **81** (1996) 297–325
30. Prigogine, I: From being to becoming, W.H. Freeman and Co, San Fransisco, (1980)
31. Prigogine, I.: Laws of Nature, Probability and time symmetry breaking, *Physica* **A 263** (1999). 528–539.
32. Sneppen, K and Bak, P,: Evolution as a self-organized critical phenomena, Proc. Natl. Acad. Sci., **92** (1995) 5209–5213.
33. Stauffer, D and Aharony, A: Introduction to Percolation theory, Taylor & Francis, London (1993)
34. Walsh, T.: The constrainedness in knife edge, Proc. AAAI-98, (1996) 406–411.
35. Wegner, P.: Why Interaction is more powerful than algorithms, Comm. ACM **50**, (1997) 81–91.
36. Wolfram, S.: A new kind of science, Wolfram Media, Inc, Champaign, Ill (2002)
37. Zak, M, Zbilut, J.P and Meyers, R.E .: From Instability to Intelligence, Springer Verlag, New York (1997).
38. Zhang, W and Korf, R.: A study of complexity transitions on the asymmetric travelling salesman problem, Artificial Intelligence, **81** (1996) 223–239.