

A BOLTZMANN MACHINE FOR THE ORGANIZATION OF INTELLIGENT MACHINES

Michael C. Moed and George N. Saridis

Center for Intelligent Robotic Systems for Space Exploration (CIRSSE)
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

1. INTRODUCTION

In our present technological society, there is a major need to build machines that would execute intelligent tasks operating in uncertain environments with minimum interaction with a human operator. Although some designers have built smart robots, utilizing heuristic ideas, there is no systematic approach to design such machines in an engineering manner.

Recently, cross-disciplinary research from the fields of computers, systems, AI and information theory has served to set the foundations of the emerging area of the Design of Intelligent Machines (Saridis, Stephanou 1977).

Since 1977 Saridis has been developing a novel approach, defined as Hierarchical Intelligent Control, designed to organize, coordinate and execute anthropomorphic tasks by a machine with minimum interaction with a human operator. This approach utilizes analytical (probabilistic) models to describe and control the various functions of the Intelligent Machine structured by the intuitively defined principle of Increasing Precision with Decreasing Intelligence (IPDI) (Saridis 1979).

This principle, even though resembles the managerial structure of organizational systems (Levis 1988), has been derived on an analytic basis by Saridis (1988). The impact of this work is in the engineering design of intelligent robots, since it provides analytic techniques for universal production (blueprints) of such machines.

The purpose of the paper is to derive analytically a Boltzmann machine suitable for optimal connection of nodes in a neural net (Fahlman, Hinton, Sejnowski, 1985). Then this machine will serve to search for the optimal design of the Organization level of an Intelligent Machine.

In order to accomplish this, some mathematical theory of the intelligent machines will be first outlined. Then some definitions of the variables associated with the principle, like machine intelligence, machine knowledge, and precision will be made (Saridis, Valavanis 1988). Then a procedure to establish the Boltzmann machine on an analytic basis will be presented and illustrated by an example in designing the organization level of an Intelligent Machine. A new search technique, the Modified Genetic Algorithm, is presented and proved to converge to the minimum of a cost function. Finally, simulations will show the effectiveness of a variety of search techniques for the Intelligent Machine.

2. THE MATHEMATICAL THEORY OF INTELLIGENT CONTROLS

In order to design intelligent machines that require for their operation control system with intelligent functions such as simultaneous utilization of a memory, learning, or multilevel decision making in response to "fuzzy" or qualitative commands, Intelligent Controls have been developed by Saridis (1977, 1983). They utilize the results of cognitive systems research effectively with various mathematical programming control techniques (Birk & Kelley, 1981).

The theory of Intelligent Control systems, proposed by Saridis (1979) combines the powerful high-level decision making of the digital computer with advanced mathematical modeling and synthesis techniques of system theory with linguistic methods of dealing with imprecise or incomplete information. This produces a unified approach suitable for the engineering needs of the future. The theory may be thought of as the result of the intersection of the three major disciplines of Artificial Intelligence, Operations Research, and Control Theory. This research is aimed to establish Intelligent Controls as an engineering discipline, and it plays a central role in the design of Intelligent Autonomous Systems.

Intelligent control can be considered as a fusion between the mathematical and linguistic methods and algorithms applied to systems and processes. In order to solve the modern technological problems that require control systems with intelligent functions such as simultaneous utilisation of a memory, learning, or multilevel decision making in response to "fuzzy" or qualitative commands. Intelligent Control is the process of implementation of an Intelligent Machine and would require a combination of "machine intelligent functions" for task organisation purposes with system theoretic methods for their execution.

The control intelligence is hierarchically distributed according to the Principle of Increasing Precision with Decreasing Intelligence (IPDI), evident in all hierarchical management systems. They are composed of three basic levels of controls even though each level may contain more than one layer of tree-structured functions (Figure 1):

1. The organization level.
2. The coordination level.
3. The execution level.

The Organisation Level is intended to perform such operations as planning and high level decision making from long term memories. It may require high level information processing such as the knowledge based systems encountered in Artificial Intelligence. These require large quantities of knowledge processing but require little or no precision.

The functions involved in the upper levels of an intelligent machine are imitating functions of human behavior and may be treated as elements of knowledge-based systems. Actually, the activities of planning, decision making, learning, data storage and retrieval, task coordination, etc. may be thought of as knowledge handling and management. Therefore, the flow of knowledge in an intelligent machine may be considered as the key variable of such a system.

Knowledge flow in an intelligent machine's organisation level represents respectively:

1. Data Handling and Management.
2. Planning and Decision performed by the central processing units.
3. Sensing and Data Acquisition obtained through peripheral devices.
4. Formal Languages which define the software.

Subjective probabilistic models or fuzzy sets are assigned to the individual functions. Thus, their entropies may be evaluated for every task executed. This provides an analytical measure of the total activity.

Artificial Intelligence methods also applicable for the processing of knowledge and knowledge rates of the organisation level of an intelligent machine have been developed by Meystel (1985) and his colleagues.

The Coordination Level is an intermediate structure serving as an interface between the organisation and execution level.

It is involved with coordination, decision making and learning on a short term memory, e.g., a buffer. It may utilise linguistic decision schemata with learning capabilities defined in Saridis and Graham (1984), and assign subjective probabilities for each action. The respective entropies may be obtained directly from these subjective probabilities.

The Execution Level executes the appropriate control functions. Its performance measure can also be expressed as an entropy, thus unifying the functions of an "intelligent machine".

Optimal control theory utilises a non-negative functional of the states of a system in the states space, and a specific control from the set of all admissible controls, to define the performance measure for some initial conditions $(x(t), t)$, representing a generalised energy function. Minimisation of the energy functional yields the desired control law for the system.

For an appropriate density function $p(x, u(x, t), t)$ satisfying Jaynes' Maximum entropy principle (1957), it was shown by Saridis (1988) that the entropy $H(u)$ for a particular control action $u(x, t)$ is equivalent to the expected energy or cost functional of the system. Therefore, minimisation of the entropy $H(u)$ yields the optimal control law of the systems.

This statement establishes equivalent measures between information theoretic and optimal control problems and unifies both information and feedback control theories with a common measure of performance. Entropy satisfies the additive property, and any system composed of a combination of such subsystems can be optimised by minimising its total entropy. Information theoretic methods based on entropy may apply (Conant 1976).

Since all levels of a hierarchical intelligent control can be measured by entropies and their rates, then the optimal operation of an "intelligent machine" can be obtained through the solution of mathematical programming problems.

The various aspects of the theory of hierarchically intelligent controls may be summarized as follows:

The theory of intelligent machines may be postulated as the mathematical problem of finding the right sequence of decisions and controls for a system structured according to the principle of increasing precision with decreasing intelligence (constraint) such that it minimizes its total entropy.

The above analytic formulation of the "intelligent machine problem" as a hierarchically intelligent control problem is based on the use of entropy as a measure of performance at all the levels of the hierarchy. It has many advantages because of the tree-like structure of the decision making process, and brings together functions that belong to a variety of disciplines.

3. KNOWLEDGE FLOW AND THE PRINCIPLE OF IPDI

The concept of entropy used in this paper may be generalised if one introduces theory of evidence for the cases that Intelligent Machines are endowed with judgment, a very human property.

The general concepts of Intelligent Control Systems are the fundamental notions of Machine Intelligence, Machine Knowledge, its Rate and Precision. The definitions useful in order to derive the principle of IPDI are presented in (Saridis, Moed 1988).

Analytically, the relations may be summarised as follows:

Knowledge (K) representing a type of information may be represented as

$$K = -\alpha - \ln p(K) = (\text{Energy}) \quad (1)$$

where $p(K)$ is the probability density of Knowledge.

From equation (1) the probability density function $p(K)$ satisfies the following expression in agreement with Jaynes' principle of Maximum Entropy (1957):

$$p(K) = e^{-\alpha - K}; \quad \alpha = \ln \int_x e^{-K} dx \quad (2)$$

The Rate of Knowledge R which is the main variable of an intelligent machine with discrete states is

$$R = \frac{K}{T} = (\text{Power})$$

It was intuitively thought (Saridis 1983), that the Rate of Knowledge must satisfy the following relation which may be thought of expressing the principle of Increasing Precision with Decreasing Intelligence

$$(MI) : (DB) \longrightarrow (R) \quad (3)$$

A special case with obvious interpretation is, when R is fixed, machine intelligence is largest for a smaller data base e.g. complexity of the process. This is in agreement with Vamos' theory of Metalanguages (1986).

It is interesting to notice the resemblance of this entropy formulation of the Intelligent Control Problem with the ϵ -entropy formulation of the metric theory of complexity originated by Kolomogorov (1956) and applied to system theory by Zames (1979). Both methods imply that an increase in Knowledge (feedback) reduces the amount of entropy (ϵ -entropy) which measures the uncertainty involved with the system.

An analytic formulation of the above principle has been derived from simple probabilistic relation among the Rate of Knowledge, Machine Intelligence and the Data Base of Knowledge. The entropies of the various functions come naturally into the picture as a measure of their activities.

4. THE DESIGN OF THE ORGANIZATION LEVEL OF AN INTELLIGENT MACHINE AS A BOLTZMANN MACHINE

In the current literature of parallel architectures for Machine Intelligence, the Boltzmann machine represents a powerful, neural network based architecture that allows efficient searches to optimally obtain the combination of certain hypotheses of input data and constraints (Fahlman, Hinton, Sejnowski 1985).

The Boltzmann architecture may be interpreted as the machine that searches for the optimal interconnection of several nodes (neurons) representing different primitive events in order to produce a string defining an optimal task. Such a device may prove extremely useful for the design of the Organisation Level of an Intelligent Machine (Saridis, Valavanis 1988) (Figure 2).

We associate the state of each node with a binary random variable $x_i = \{0, 1\}$, with a priori probabilities $p(x_i = 1) = p_i$, $p(x_i = 0) = 1 - p_i$, where 1 represents the firing of neuron i , and 0 indicates neuron i idle. The state vector of the network, $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ is an ordered set of 0's and 1's describing the state of the machine in terms of firing/idle nodes, for an n node machine. The neurons of the machine can be visible, or hidden (Hinton, Sejnowski 1986). It is possible to extract the string of primitive events representing the optimal task by examining the state vector of the visible nodes in the network in steady state response to a given input.

The standard formulation of the Boltzmann machine uses Energy as a cost function which is minimised to find the optimal state of the machine. However, in (1) we defined knowledge as a form of Energy. This is not the function to be minimised in the Intelligent Machine. Instead, we will be minimising the Energy of Flow of Knowledge (F), which is the amount of knowledge which must flow through the machine in order to accomplish a particular task. This is found by:

$$F = R * T \quad (4)$$

where T is the total time of knowledge flow. By minimising F , the Intelligent Machine reduces the amount of Energy required to make a decision.

5. ENTROPY AS A MEASURE OF UNCERTAINTY

Entropy is used as a measure of uncertainty in the intelligent machine. The entropy manifests itself in the interaction and interconnection of nodes in the network. We can define energy of flow of knowledge into node i by:

$$F_i = \alpha_i - \frac{1}{2} \sum_j w_{ij} x_j x_i \quad (5)$$

and the probability the machine is in a state where Energy = F_i by:

$$P(F_i) = e^{-\alpha_i + \frac{1}{2} \sum_j w_{ij} x_j x_i} \quad (6)$$

where:

$w_{i,j}$ is the interconnection weight between nodes i and j

$w_{ii} = 0$

α_i is a probability normalising factor which insures $.5 \leq P(F_i) \leq 1$

Unlike the Boltzmann machine, this formulation does not remove α_i when $x_i = 0$. Instead, the machine operates from a base entropy level which it tries to reduce.

By bounding $P(F_i)$ by 0.5, we find that the entropy of being in a state where Energy = F_i increases as F_i increases. In other words, as the Energy increases, the uncertainty increases as well.

The Entropy of Knowledge Flow in the machine can be formulated as:

$$H(F) = - \sum_i P(F_i) \ln\{P(F_i)\}$$

Therefore:

$$H(F) = \sum_i \left(\alpha_i - \frac{1}{2} \sum_j w_{ij} x_i x_j \right) e^{-\alpha_i + \frac{1}{2} \sum_j w_{ij} x_i x_j}$$

6. SEARCH TECHNIQUES FOR THE INTELLIGENT MACHINE

Three random search techniques are compared here which may be used to find the minimum entropy in the Organisation Level of an intelligent machine. By examining the active visible neurons in the minimum entropy state of the network, one can determine the sequence of primitive events which produce a string

defining an optimal task for an intelligent machine. The techniques presented here allow escape from local entropy minima, which lead to incorrect task decisions, by randomly selecting states while searching for the global entropy minimum.

6.1 A Genetic Algorithm Search Technique

A technique which minimizes a system cost function is the Genetic Algorithm (Holland 1975). In contrast to other random search techniques, the Genetic Algorithm (GA) maintains a population of points in the space while searching for the optimum.

Here we present a modified GA which will converge in probability to the minimum cost. The standard GA has been changed by inserting spacer steps of an algorithm which is known to converge in probability, Expanding Subinterval Random Search.

Spacer steps are defined as follows: Suppose B is an algorithm which together with a descent function Z and solution set T converges in probability. We can define an algorithm C by $C(x) = \{y : Z \leq Z(x)\}$. In other words, C applied to x can give any point so long as it does not increase the value of Z , the current cost. B represents the spacer step, and the complex process between spacer steps is C . Thus, the overall process amounts to repeated applications of the composite algorithm CB . CB will converge in probability if B is repeated infinitely often and C does not increase the value of the current cost (Luenberger 1984).

We introduce the concept of "immigration" to imbed ESRS into GA. Infinitely often, we insert a randomly generated point into the GA search which forms the spacer step. The frequency of insertion is called the "immigration rate." By changing the "immigration rate," the algorithm adjusts its focus from global to local searches. This rate may be fixed dependent on the complexity of the search space, or may vary while the search is in progress. A high "immigration rate" will force random search. A low rate will cause the GA. Parallels can be drawn to Simulated Annealing which starts as a near random search, and eventually becomes gradient descent. For the modified GA, the "immigration rate" is analogous to thermal energy in Simulated Annealing. The modified algorithm described in detail below converges in probability to the minimum cost.

6.1.1 Standard Genetic Algorithm

In general, each point in the space is represented by a binary string and has an associated cost dictated by the system cost function for that point. Since the makeup of the population is changed each iteration to emphasize members (points) which minimize the cost function, a near-uniform population will develop corresponding to a local minima in the cost function.

The following notation is used:

- P = population of members (points)
- P' = new population of members
- $|P|$ = number of members in P
- P_k = k th member of the population P
- $P_k(m)$ = m th bit of P_k
- J_K = Cost of P_k
- S_k = probability of member k being selected from current population
- J_{\max} = max cost of any possible string in P
- n = length of P_k in bits

Each iteration of the search algorithm proceeds as follows:

Repeat:

1. Compute $J_k, \forall P_k \in P$
2. Let $J'_k = J_{\max} - J_k, \forall k$. Compute $S_k = \frac{J'_k}{\sum_i J'_i}, \forall k$.

Repeat:

- 3.1 Randomly select P_j, P_k from P based on S_j, S_k .
- 3.2 Randomly generate an index i between $1..n$. Exchange the right string halves of P_j, P_k (i.e., $P'_j(i..n) = P_k(i..n)$ and $P'_k(i..n) = P_j(i..n)$). This is called "crossover" or "mating."

3.3 Place P'_j, P'_k in P' . Return P_j, P_k to P
 until $|P^j| = |P|$.

4. Set $P = P'$
 until $P_k \in P$ and P_k has minimum cost.

In an attempt to prevent population convergence on a local minima ("premature convergence"), a "mutation" operator is added to the system. With a new generation of the population, each bit of every member has a small probability of inverting. The inversion adds diversity to the population and promotes search in previously unexplored regions of the space in an attempt to find the global cost minimum.

Particular aspects of this algorithm make it a powerful search tool. The "crossover" mechanism forces search on an n-dimensional hypercube by discovering and promoting particular substrings (called "building blocks") which perform well. These "building blocks" combine to discover the topology of the search space, which may not be known initially. Since the algorithm uses a population of points, many planes of the hypercube can be searched at once, leading to "implicit parallelism." Further, since members within a population are independent, a new population may be formed by "mating" in parallel. Steps 3.1-3.3 can be blocked together and generate two new members in parallel with other "mating" blocks. These features as well as others are described in depth in (Goldberg 1989). Applications of this algorithm are presented in (De Jong 1975, Grefenstette et al 1985, Davis and Coombs, 1987).

Heuristic algorithms within GA have been developed to avoid convergence at local minima (Maldin 1984, Suh and Van Gucht 1987). The "SIGH" system (Ackley 1987) uses active and passive subpopulations to escape local minima. When particular members of the population are performing poorly, they become passive until the active subpopulation converges. If this convergence is premature, the passive members are activated, bringing diversity and new structure to the search.

6.1.2 Modified Genetic Algorithm

Unfortunately, many of the heuristically driven GA searches perform well for a small set of functions, and prematurely converge for functions outside that set. However, it can be shown that under certain conditions, the GA will converge in probability to the global minimum of the cost function.

The conditions are as follows:

1. Instead of (or in addition to) the "mutation" operator, an "immigration" operator is used. Introduce a randomly generated member P_i to P' every M populations for some integer $M > 0$.
2. If $P_k \in P$ and $\forall P_i \in P, P_k \geq P_i$ then $P_k \in P'$.

Step 1 inbeds ESRS into the GA, where ESRS is algorithm B as described by (Luenberger 1984) and stated above. Step 2 insures $C(x) = \{y : Z(y) \leq Z(x)\}$ where C is the GA algorithm. Therefore, CB, the modified GA, converges in probability to the cost minimum.

As one can see, these conditions do not bind the algorithm severely. The "immigration" rate (immigrations/population), $1/M$, is related to the "mutation" rate (mutations/bit) as follows:

$$1/M = (\text{mutations/bit}) * (\text{members/population})$$

In fact, the "immigration" of new members may be probabilistic, with probability $1/M$.

6.2 Simulated Annealing

One random search technique commonly used to find the global minimum cost in a Boltzmann Machine is Simulated Annealing. This technique simulates the annealing process of metal by probabilistically allowing uphill steps in a state-dependent cost function while finding the global cost minimum, or ground state. The algorithm allows control of the search randomness by a user specified parameter, T . In true metal annealing, this cost function is the Energy of the system, E , and T is the annealing temperature (Kirkpatrick et al. 1983). This method can easily be adapted for finding the minimum entropy of the Organization Level of an intelligent machine.

Given a small random change in the system state $X_i = \{x_1, x_2, \dots, x_n\}$ to X'_i and the resulting entropy change, ΔH , if $\Delta H \leq 0$, the change is accepted. If $\Delta H > 0$, the probability that the new state is accepted is:

$$p(X_{i+1} = X'_i) = e^{-\Delta H/K_B T} \quad (16)$$

where K_B is the Boltzmann Constant and T is a user set parameter. By reducing T along a schedule, called the annealing schedule, the system should settle into a near-ground state as T approaches 0.

Another method for simulated annealing is discussed in (Hinton, Sejnowski 1986). Using this method, if the entropy change between X_i and X'_i is ΔH , then regardless of the previous state, accept state X'_i with probability:

$$p(X_{i+1} = X'_i) = \frac{1}{1 + e^{-\Delta H/T}} \quad (17)$$

Since an intelligent machine consists of a set of binary states, it should be noted that in both of the above methods, X'_i is Hamming distance 1 from X_i (Kam et al. 1985).

The process of simulated annealing escapes local minima through its probabilistic random search, and probabilistically converges to the global cost minimum under certain conditions (Geman, Geman 1984). The next technique, Expanding Subinterval Random Search, probabilistically guarantees convergence within a δ neighborhood to the global minimum of a specified cost function.

6.3 Expanding Subinterval Random Search

A third technique for finding the global minimum value for a cost function for a dynamic system is Expanding Subinterval Random Search as described in (Saridis 1976). Using entropy as the cost function and given a state X_i , one may define the following random search algorithm for an appropriately selected μ :

$$X_{i+1} = \begin{cases} X'_i & \text{if } H(X'_i) - H(X_i) \leq 2\mu \\ X_i & \text{if } H(X'_i) - H(X_i) > 2\mu \end{cases} \quad (18)$$

where $H(Y)$ is the entropy induced by state $Y = (y_1, y_2, \dots, y_n)$ and X'_i is a randomly selected state vector generated from a prespecified independent and identically distributed density function, defined by (5).

It is shown that:

$$\lim_{n \rightarrow \infty} \text{Prob} [H(X_n) - H_{\min}^* < \delta] = 1 \quad (19)$$

where H_{\min}^* is the global minimum entropy of the network. The existence of H_{\min}^* is proven in the cited work.

This method can be used on-line to find the global minimum entropy in the Organisation Level of an intelligent machine.

7. EXPERIMENTAL RESULTS

7.1 Simulation of Search Techniques

A net was created which recognised strings of 15 bit binary numbers. The net was formulated using the standard Energy methods found in (Hinton, Sejnowski 1986). Energy was used instead of Entropy in these simulations for two reasons. First, to compare the results of this simulation to the results of simulations by other researchers, a standard measure had to be used. Second, the method for creating regions of attraction in an Entropy based net is still being investigated.

The net had three Energy minima, corresponding to states (001010100100100, 110110110001101, 00111101100010). The respective Energy for these three states were (0.8, 0.6, 1.0). Each simulation technique attempts to find the global Energy minimum of the net, which was 0.6. The cases presented here show best and worst performance of each technique over 10 trials. Other cases which varied the depth and width of the Energy wells are presented in (Saridis and Moed, 1988). For this experiment, the wells were narrow.

The modified Genetic Algorithm was performed as presented in Section 6.1. The population was set at 20 members. Each member was 15 bits long, so the number of bits in each population was 300. The "immigration rate" was set to 0.5 which corresponds to a mutation rate of 0.025.

Simulated Annealing was performed using the acceptance criteria in (17). The system was cooled in accordance with:

$$\frac{T_1(t)}{T_0} = \frac{1}{\log(10 + t)}$$

where $T_1(t)$ = temperature at time t

T_0 = initial temperature.

The net state changed in Hamming distance 1 increments.

Expanding Subinterval Random Search (Saridis 1976) was slightly modified to reinforce the probabilistic selection of node states which reduced the Energy in the net. The probability of a node being active

was initially 0.5. When the Energy was reduced during search, the probability of the node being reactivated became

$$P(x_i = 1) = P(x_i = 1) + [1.0 - P(x_i = 1)] * 0.1$$

if the node was active, or

$$P(x_i = 1) = P(x_i = 1) - P(x_i = 1) * 0.1$$

if the node was inactive.

Figures 3-8 present the best and worst performance of each algorithm over 10 trials. Modified GA found the minimum Energy string between the 20th and 180th population. Since there were 20 strings per population, this indicates that between 400 and 3600 points had to be generated. The best performance by Simulated Annealing required over 5500 iterations. The worst performance did not converge in 12000 iterations (the most attempted). As a guideline, the best performance of the random search ESRS was slightly over 2000 iterations. The worst performance did not converge in 12000 iterations. The results of these limited experiments force a closer examination of the Modified Genetic Algorithm as a search technique for minimizing the Energy in a Boltzmann machine.

8. CONCLUSIONS

A mathematical theory for intelligent machines was proposed and traced back to its origins. The methodology was developed to formulate the "intelligent machine", of which an intelligent robot system is a typical example, as a mathematical programming problem as using the aggregated entropy of the system as its performance measure. The levels of the machine structured according to the Principle of Increasing Precision with Decreasing Intelligence can adapt performance measures easily expressed as entropies. This work establishes an analytic formulation of the Principle, provides entropy measures for the account of the underlying activities, and integrates it with the main theory of "Intelligent Machines". Optimal solutions of the problem of the "intelligent machine" can be obtained by minimizing the overall entropy of the system.

This formulation was proven to be applicable to the derivation and design of parallel architectures for Machine Intelligence. The Boltzmann machine was analytically derived from the definitions of knowledge flow and Jaynes' principle of maximum entropy. The Modified Genetic Algorithm was presented as a search technique which converged in probability to the minimum of a specified cost function. Three techniques, the Modified Genetic Algorithm, Simulated Annealing, and Expanding Subinterval Random Search were described as methods to find the global minimum Energy of a Boltzmann Machine. Simulations using these search techniques were conducted, and results indicate that the modified Genetic Algorithm may be an efficient method to find the minimum Energy.

REFERENCES

- Ackley, D. H., (1987), "A Connectionist Machine for Genetic Hillclimbing," Kluwer Academic Publishers.
- American Heritage Dictionary of the English Language*, (1969).
- Birk, J. R. and Kelley, R. B., (1981), "An Overview of the Basic Research Needed to Advance the State of Knowledge in Robotics," *IEEE Trans. on SMC*, SMC-11, No. 8, pp. 575-579.
- Conant, R. C., (1976), "Laws of Information Which Govern Systems," *IEEE Trans. on SMC*, SMC-6, 4, 240-255, April 1976.
- Davis, L., Coomb, S., (1987), "Genetic Algorithms and Communication Link Speed Design: Theoretical Considerations," Proceedings of an International Conference on Genetic Algorithms and their Applications, Cambridge, MA, 252-256.
- De Jong, K. A., (1975). "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Doctoral Dissertation, University of Michigan.
- Fahlman, S. E., Hinton, G. E., Sejnowski, T. J. (1983), "Massively Parallel Architectures for AI: NETL, THISLE and Boltzmann Machines," Proceedings of National Conference on AI, Menlo Park, CA.
- Geman, S. and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, pp. 721-471, November 1984.
- Goldberg, D. E., (1989), "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley.

- Grefenstette, J. J., Gopal, R., Rosmaita, B. J., Van Gucht, D., (1985), "Genetic Algorithms for the Traveling Salesman Problem," Proceedings of an International Conference on Genetic Algorithms and their Applications, Pittsburgh, PA, 160-168.
- Hayes-Roth, et al., (1983), *Building Expert Systems*, Addison-Wesley, New York.
- Hinton, G. E., Sejnowski, T. J. (1986), "Learning and Relearning in Boltzmann Machines," pp. 282-317, in *Parallel Distributed Processing*, ed, D. E. Rumelhart and J. L. McClellan, MIT Press.
- Holland, J. H., (1975), "Adaptation in Natural and Artificial Systems," The University of Michigan Press, Ann Arbor, Michigan.
- Hopfield, J.J. (1982), "Neural Networks and Physical Systems with Emergent Collective Computation Abilities," Proc. National Acad. Sci., U.S.A, Vol. 79, pp. 2554-2558, April 1982.
- Jaynes, E. T., (1957), "Information Theory and Statistical Mechanics," *Physical Review*, 106, 4.
- Kam, M., Guez, A. and Cheng, R. (1987), "On Stable Points and Cycles in Binary Neural Networks," Proceedings of the IEEE Intl. Symp. on Intelligent Control, pp. 321-326, Philadelphia, PA, January.
- Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M. P. (1983), "Optimization by Simulated Annealing," *Science*, Vol. 220, Number 4598, May 13, 1983.
- Kolmogorov, A. N., (1956), "On Some Asymptotic Characteristics of Completely Bounded Metric Systems," Dokl Akad Nank, SSSR, Vol. 108, No. 3, pp. 385-9.
- Luenberger, D. L., (1984), "Linear and Nonlinear Programming," Second Addition, Addison-Wesley.
- Maudlin, M. L., (1984), "Maintaining Diversity in Genetic Search," Proceedings of the National Conference on Artificial Intelligence, 247-250.
- Meystel, A., (1985), "Intelligent Motion Control in Anthropomorphic Machines," Chapter in *Applied Artificial Intelligence*, S. Andriole Ed. Pentecellis Books, Princeton, NJ.
- Saridis, G. N. (1977), "Expanding Subinterval Random Search for System Identification and Control," IEEE Trans. on Auto. Control, pp. 405-412, June.
- Saridis, G. N., (1979), "Toward the Realization of Intelligent Controls," *IEEE Proceedings*, Vol. 67, No. 8.
- Saridis, G. N., (1985), "Control Performance as an Entropy," *Control Theory and Advanced Technology*, 1, 2.
- Saridis, G. N., (1988), "Entropy Formulation of Optimal and Adaptive Control," IEEE Transactions on AC, Vol. 33, No. 8, pp. 713-721.
- Saridis, G. N. and Graham, J. H., (1984), "Linguistic Decision Schemata for Intelligent Robots," *Automatica*, Vol. 20, No. 1, 121-126.
- Saridis, G. N., Moed, M. C., (1988), "Analytic Formulation of Intelligent Machines as Neural Nets," Proceedings of the IEEE Conference on Intelligent Control, Washington, DC, August 1988.
- Saridis, G. N., Stephanou, H. E., (1977), "A Hierarchical Approach to the Control of a Prosthetic Arm," *IEEE Trans. on SMC*, Vol. SMC-7, No. 6, pp. 407-420.
- Saridis, G. N. and Valavanis, K. P., (1988), "Analytical Design of Intelligent Machines," *Automatica the IFAC Journal*.
- Suh, J. Y., Van Gucht, D., (1987), "Incorporating Heuristic Information into Genetic Search," Proceedings of an International Conference on Genetic Algorithms and their Applications, Cambridge, MA, 100-107.
- Vamos, T., (1986), "Metalanguages - Conceptual Models. Bridge Between Machine and Human Intelligence," Working Paper E/37, Hungarian Academy of Science.
- Zames, G., (1979), "On the Metric Complexity of Causal Linear Systems, ϵ -entropy and ϵ -dimension for Continuous Time," *IEEE Trans. Automat. Control*, Vol. AC-24, pp. 222-230, April.

ACKNOWLEDGMENT

This work was supported by NASA Grant NAGW-1333.

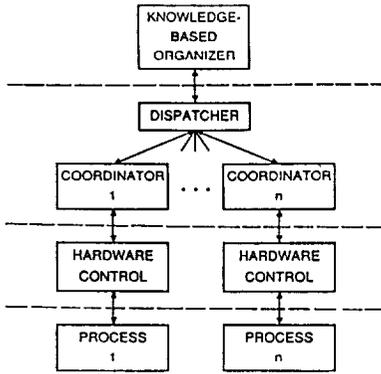


Figure 1. Hierarchical Intelligent Control System

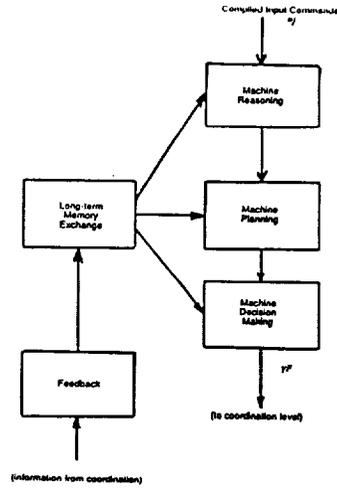


Figure 2. Block Diagram of the Organization Level

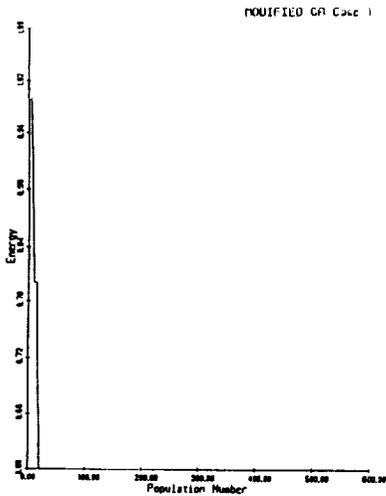


Figure 3. Best Case

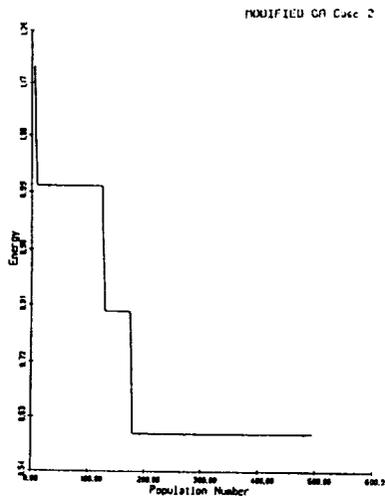


Figure 4. Worst Case

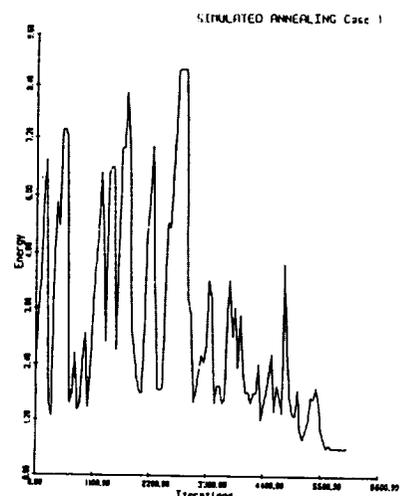


Figure 5. Best Case

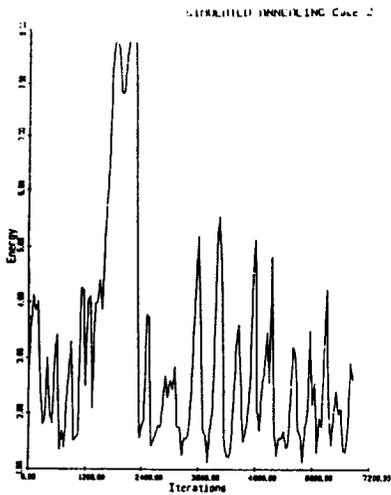


Figure 6. Worst Case

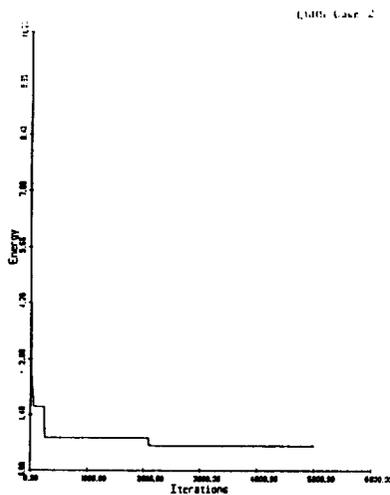


Figure 7. Best Case

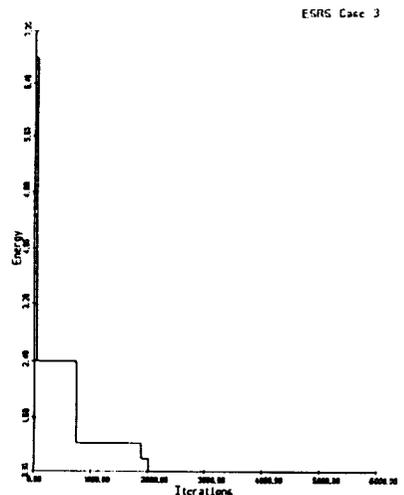


Figure 8. Worst Case