Charles River Analytics Inc. [redacted] (D)

U.S. Army Missile Command Contract No. DAAH01-93-C-R327
Charles River Analytics Inc. Report No. R9305
Final Report

A Neural/Expert Based Client Server Architecture for MITE ITS

[redacted]

J. M. Mazzu, S.M. Botros, and A.K. Caglayan
Charles River Analytics Inc.
55 Wheeler Street
Cambridge, MA 02138

February 1994

Prepared for:

U.S. Army Missile Command
Redstrone Arsenal, AL 35898

Charles River Analytics Inc.

U.S. Army Missile Command Contract No. DAAH01-93-C-R327
Charles River Analytics Inc. Report No. R9305
Final Report

A Neural/Expert Based Client Server Architecture for MITE ITS

J. M. Mazzu, S.M. Botros, and A.K. Caglayan
Charles River Analytics Inc.
55 Wheeler Street
Cambridge, MA 02138

February 1994

Prepared for:

U.S. Army Missile Command
Redstrone Arsenal, AL 35898

SUMMARY

A MITE (multi-node, interactive, task-sharing, expert-instruction) system, in general, is responsible for determining task allocation strategies for members of a team who must work together over a computer network to accomplish an overall objective. The MITE system must adapt to changing skill and performance levels of the team members and continually reallocate the tasks to insure optimal team performance. In order to best allocate tasks among team members, the MITE system should maintain internal models of the member's capacity and knowledge relating to the tasks which that member may be allocated. The system should also provide expert recommendations and instruction for team members to improve performance.

In this Phase I study, while investigating the application of hybrid neural network / knowledge base strategies to the problem of MITE systems, we also look for foundation technologies that can be applied to current or future commercial products with high potential returns. Designing the MITE system with an object-oriented client/server architecture provides the necessary reusability of code objects for a variety of application domains (Rumbaugh, 1991). For example, the complete MITE system can be used for both Army weapons systems, such as Avenger, and large scale manufacturing applications. The task allocation objects can be used within MITE systems or for single user project scheduling. The neural network objects developed for the task allocation module can also be extracted and used for a variety of other optimization problems.

TABLE OF CONTENTS

LIST OF FIGURES

FOREWORD

This final report was prepared by Charles River Analytics Inc., Cambridge, MA for the U.S. Army Missile Command under Contract No. DAAH01-93-C-R327. It describes the results of contract efforts to develop a modular hybrid multi-node, interactive, task-sharing, expert-instruction (MITE) system with an object-oriented client/server architecture.

The program direction was provided by Ms. Carol Barclay at the U.S. Army Missile Command in Redstone Arsenal, AL.

The program was directed by Mr. James M. Mazzu of Charles River Analytics Inc. with Dr. Alper K. Caglayan as the program manager. Dr. Sherif M. Botros contributed to the project.

1. INTRODUCTION

1.1 Background

The hybrid integration of artificial neural networks (ANNs) and knowledge-based expert systems (KBs) is an ideal step in the development of intelligent systems. In general, the two methods complement each other such that ANNs provide *soft* constraints, while expert systems allow *hard* constraints Glover and Rao (1990). Specifically, ANNs perform nonlinear functions, pattern recognition capabilities, fault tolerance and parallel processing; while expert systems involve language processing, formal logic and rule interpretation. Here, we exploit the complementary strengths of neural networks and knowledge-based expert systems to investigate a modular hybrid multi-node, interactive, task-sharing, expert-instruction (MITE) system with an object-oriented client/server architecture.

A MITE system, in general, is responsible for determining task allocation strategies for members of a team who must work together over a computer network to accomplish an overall objective. The MITE system must adapt to changing skill and performance levels of the team members and continually reallocate the tasks to insure optimal team performance. In order to best allocate tasks among team members, the MITE system should maintain internal models of the member's capacity and knowledge relating to the tasks which that member may be allocated. The system should also provide expert recommendations and instruction for team members to improve performance.

Charles River Analytics has significant accomplishments in many areas directly related to the MITE system development. We have identified optimal task allocation methods for aircrews and developed pilot/crew models using neural networks and expert systems Zacharias and Gonsalves (1990); object-oriented task representation, rule-based situation assessment, domain simulation, and aircrew modeling Zacharias and Riley (1992). Additionally, our company has been in the forefront of hybrid neural network/expert system R&D over the last number of years. We have developed a hybrid neural network knowledge based structural monitoring system for smart structure applications Mazzu, Caglayan, Allen, et al. (1992), developed a hybrid neural network knowledge based system for robotic pose determination Mazzu and Caglayan (1992), multiple target recognition Gonsalves and Caglayan (1992) remote sensing Mazzu, Snorrason and Caglayan (1992), nuclear plant monitoring Mazzu, Caglayan and Gonsalves (1992), and real-time intelligent systems Caglayan, Walker and Riley (1992).

The results of our extensive research is evident in our successful commercialization of the first learning agent, **Open Sesame!®**, available on the Macintosh platform Charles River

Analytics Inc. (1992). Open Sesame!® is the first software assistant that learns by watching the user, what to do for the user. We have also successfully commercialized the industry's first hybrid development environment, NueX™, which facilitates the development of hybrid neural network / knowledge base systems.

1.2 Summary

In this Phase I study, while investigating the application of hybrid neural network / knowledge base strategies to the problem of MITE systems, we also look for foundation technologies that can be applied to current or future commercial products with high potential returns. Designing the MITE system with an object-oriented client/server architecture provides the necessary reusability of objects for a variety of application domains Rumbaugh, Blake, Premerlani, et al. (1991). For example, the complete MITE system can be used for both Army weapons systems, such as Avenger, and large scale manufacturing applications. The task allocation objects can be used within MITE systems or for single user project scheduling. The neural network objects developed for the task allocation module can also be extracted and used for a variety of other optimization problems.

In our study, we draw upon our achievements in adaptive modeling, task allocation, training systems, diagnostics and assessments, while incorporating the hybrid neural network/expert system methodology to provide a significant advancement in the development of a hybrid MITE systems. In particular, answers to the following questions are sought:

- What *object-oriented client/server architectures* will best support the development of hybrid MITE systems?
- What *hybrid neural network/knowledge-based strategies* can be developed for recognizing member characteristics, styles, and temporal response patterns in order to create adaptable models of team members, and to perform model comparison diagnostic functions?
- What *knowledge base structures* are necessary *to represent* domain expertise in the form of expert and domain models, which include *domain task definitions and requirements?*
- What domain expert knowledge is required for developing the expert system capabilities to provide intelligent neural pattern interpretation and member assessments?

- Can domain specific rules be automatically extracted from the neural networks in order to maintain an accurate *model-based representation of the member's knowledge and performance?*
- What hybrid strategies can be employed to insure the proper *allocation of tasks* based on the team member models located at each node *on the PC network?*

1.3 Outline of the Report

This report is divided into nine chapters. Chapter 2 formulates the problem of MITE systems and the executive operation levels. Hybrid neural network / knowledge base methodologies are described in Chapter 3. The MITE system is presented in object oriented methodology in Chapter 4. Chapter 5 describes in detail the optimal task allocation strategies developed. The MITE diagnostic and interpretation module is presented in Chapter 6 with adaptive models presented in Chapter 7. The full-scale research prototype requirements are presented in Chapter 8 while Chapter 9 presents the research and development conclusions and recommendations for future work.

2. FORMULATION OF THE PROBLEM

2.1 MITE systems

The objective is to obtain a major advantage in multi-node, interactive, task-sharing, expert-instruction (MITE) systems by using *hybrid neural network/rule-based expert systems* in an *object-oriented client/server architecture*, to capitalize on their complementary strengths within the optimal task allocation, domain/team member models and expert recommendations and instruction. Neural networks will use *quantitative knowledge* for pattern classification and estimation while expert systems represent and process *qualitative knowledge* in rule-bases.

The hybrid MITE System (Figure 2.1-1) extends the architecture of Warren, Goodman and Maciorowski (1993) to include Team and Scheduling Modules required for multi-node task sharing functionality, and provides each module with the hybrid neural network/expert system capabilities required to produce an intelligent adaptable system.

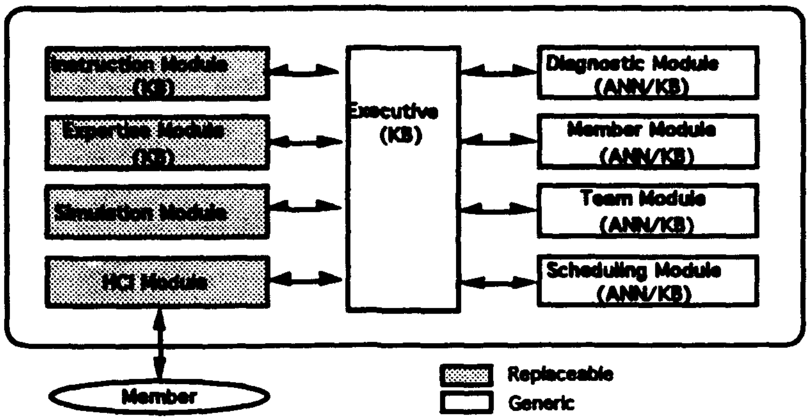

Figure 2.1-1: MITE System Architecture

Here, we take advantage of the *object-oriented methodology* to encapsulate the MITE functionality of instruction, expert domain knowledge, domain simulation, human-computer interface (HCI), performance diagnostics, adaptive member models, team objectives, and optimal task-scheduling. In general, the diagnostics compare the member model with the domain expert model to determine the team member's progress. For multi-node task sharing functionality, the diagnostics are monitoring the team performance in order to compare the team model with the domain model (part of the Expertise Module), and appropriately allocating the domain tasks through the Scheduling Module. The Executive maintains the mutli-node operation levels (ITS, TEAM-FM and TEAM-PM), while being responsible for all messages sent to and from the MITE Modules.

The high level domain and interface dependent modules are designed to contain replaceable code and rule base objects; this insures the maximum reusability of system components. Figure 3.2.1 shows that the Diagnostics, Member, Team and Scheduling Modules have been identified to incorporate the hybrid neural network/expert system (ANN/KB) strategies, while the Instruction, Expertise, and Executive will at least take advantage to the knowledge-based expert system technology (KB).

2.2 MITE Executive and Operational Levels

The system ***Executive*** (Figure 2.2-1) is responsible for maintaining the MITE *client/server* operation levels and coordinating all messages sent to and from the previously identified MITE modules. The Executive encapsulates an Executive Function Manager and a Message Manager to accomplish these objectives.

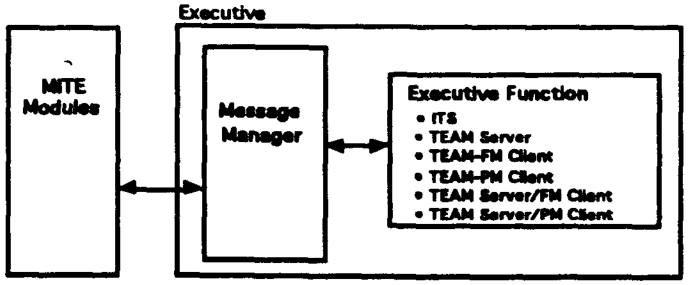

Figure 2.2-1: MITE Executive

For PC network operations, each computer's system Executive is responsible for maintaining its appropriate Executive functionality according to the overall MITE operation level. The MITE *operation levels* consist of **ITS, TEAM-FM** (Full-Up Mode), and **TEAM-PM** (Pseudo-Member Mode). In order to accomplish these overall operation levels, the Executive must be capable of six independent *functionalities*, including: ITS, TEAM Server, TEAM-FM Client, TEAM-PM Client, TEAM Server/FM Client, and TEAM Server/PM Client. The object-oriented methodology allows all MITE Modules to be ignorant of the Executive functionality while being experts on their particular modular functionality. Therefore, the same MITE Modules can be used with any of the PC network Executives operating with any of the six functionalities.

3. HYBRID NEURAL/EXPERT METHODOLOGY

3.1 Hybrid Methodology

The hybrid methodology exploits the complementary strengths of neural networks and expert systems (Figure 3.1-1) to create intelligent systems that can outperform either method alone. ANNs and expert systems both offer unique solutions to various technical problems. However, each have weaknesses which can be balanced by the strengths of the other. For instance, ANNs would not be appropriate for constraint based analysis, though expert systems would easily handle this. On the other hand, expert systems would have difficulty recognizing patterns from large data sets, while ANNs can process large data sets and continue to recognize patterns even when some of the data is missing. Therefore, the hybrid methodology takes advantage of complementary characteristics which maximize the functionality of both ANNs and expert systems, in order to obtain an intelligent system where the whole is greater than the sum of its parts.

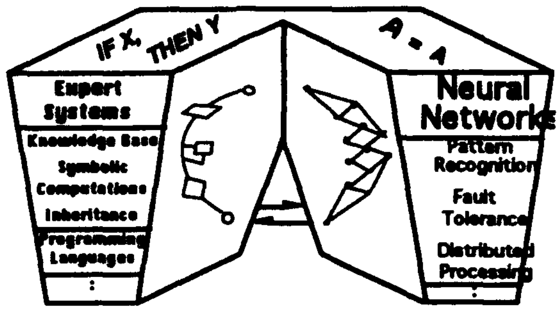

Figure 3.1-1: Hybrid Complementary Strengths

In general, expert systems are useful where the expertise of a domain expert is available and this expertise can be described as a set of rules and facts. Neural networks are useful where there is no real expertise or when the domain knowledge cannot easily be captured in terms of rules. Most real life problems fall somewhere in between these two realms, and therefore the hybrid integration of both technologies is desirable. Charles River Analytics has conducted extensive research in hybrid systems, applied the hybrid methodology to the development of intelligent systems for advanced applications, commercialized the industry's first hybrid development environment, **NueX™**, and the first commercially available learning agent for the Macintosh, **Open Sesame!®**.

3.2. Neural Networks Overview

Artificial neural networks (ANNs) Anderson and Rosenfeld (1988) represent nonalgorithmic class of information processing for using massively parallel distributed processing architectures. Stimulated by the efforts directed at understanding the interconnection of neurons in the human brain allowing the storage, retrieval, and processing of complex data, research over the last 25 years in artificial neural systems has produced solutions to complex problems in visual pattern recognition, combinatorial search, and adaptive signal processing.

Hopfield

In this section, we present a brief introduction to Hopfield neural networks and how they can be used to optimize static continuous valued functions. For a more detailed analysis of Hopfield networks, we refer the reader to Hertz, Krough and Palmer (1991) and Hopfield (1984).

A Hopfield network with continuous valued units consists of a fully connected network of units (neurons). The input to each unit is a weighted summation of the output of all the other units as shown in equation 1. The output of each unit is a monotonic function of the input, usually called the activation function.

$$V_i = f(u_i) = f(\sum_j w_{ij} V_j) \tag{1}$$

where u_i and V_i are the input and output of unit i respectively and f(.) is the activation function. The connection weights w_{ij} form a symmetric connectivity matrix. Three different methods have been proposed to update the output of the different units: asynchronous, synchronous and continuous. In the asynchronous update, each unit is updated randomly and independently from all other units. In the synchronous update, the units are updated simultaneously at each clock cycle. In the continuous update, the units are updated continuously based on equation 2.

$$\tau_l \frac{dV_i}{dt} = -V_i + f(u_i) = -V_i + f(\sum_j w_{ij} V_j) \quad (2$$

We are more interested in the continuous updating because it has been shown that it yields better results for combinatorial optimization problems. It has been shown, using Lyapunov stability theorems, that for a symmetric positive definite matrix of weights and a monotonic function f(.), the set of equations (2) converges to a unique equilibrium point.

To use a Hopfield network for the minimization of a multivariable cost function we follow the following steps Hertz, et al. (1991):

1. Define an energy function H(V) which is bounded from below. The minimum of this energy function should correspond to the minimum of the cost function to be optimized.

2. use $V_j = f(u_j)$, where f(.) is a monotonically increasing function.

3. use the update equation:

$$\tau_l \frac{du_i}{dt} = - \frac{dH(V)}{dV_i} \tag{3}$$

4. The weights of the network are determined by the function $\frac{dH(V)}{dV_i}$. If the energy function H(V) has a quadratic form :

$$H(V) = V^T W V \tag{4}$$

where **W** is a positive definite function, then the matrix **W** define the connectivity weight matrix.

Adaptive Resonance Theory

ART2-A Carpenter, Grossberg and Rosen (1991) is an algorithmic version of ART2 Carpenter and Grossberg (1987a) which is a dynamical system that can perform unsupervised classification of an arbitrary number of analog spatial patterns. ART2-A is a three layer network, where one performs preprocessing, one is a feature representation field, and the third is a category representation field with competitive learning.

Competitive Neural Networks

Competitive networks differ from Hopfield networks by the fact that some of the constraints that the network should satisfy are established through competition and need not be specified in the energy function. The processing units of such a network compete with each other and are arranged in such a way that the more powerful units (i.e. the ones receiving the largest inputs) win and are selected. The other units are inhibited. Competitive networks have been used for clustering and classification (for example ART networks) or also for constraint satisfaction in combinatorial optimization Looi (1992). An example of a simple competitive network is a winner-take-all network. In this network competition is established through lateral inhibition between the units in the output layer. At steady state, the output unit with the largest inputs is reinforced and becomes active and inhibits the rest of the output units. The book by Hertz, Krogh and Palmer Hertz, et al. (1991) contains numerous other examples of competitive networks.

3.3 Knowledge-Based Expert Systems Overview

An expert system is a computer program that can perform a task normally requiring the reasoning ability of a human expert. Expert systems are highly specialized according to their application domains. Although any program solving a particular problem may be considered to exhibit expert behavior, expert systems are differentiated from other programs according to the manner in which the domain specific knowledge is structured, represented, and processed to produce solutions. In particular, expert system programs partition their knowledge into the following three blocks: Data Base, Rule Base, and Inference Engine. Expert systems utilize symbolic and numeric reasoning in applying the rules in the Rule Base to the facts in the Data Base to reach conclusions according to the construct of reasoning specified by the Inference Engine.

Knowledge Representation

There are two basic types of knowledge that can be incorporated into expert systems: declarative knowledge and procedural knowledge. The kind of knowledge describing the relationships among objects is called declarative knowledge. The kind of knowledge prescribing the sequences of actions that can be applied to this declarative knowledge is called procedural knowledge. In expert systems, procedural knowledge is represented by production rules whereas declarative knowledge is represented by frames and semantic networks in addition to production rules.

While expert systems have been traditionally built using collections of rules based on empirical associations, interest has grown recently in knowledge-based expert systems which perform reasoning from representations of structure and function knowledge. For instance, an expert system for digital electronic systems troubleshooting is developed by using a structural and behavioral description of digital circuits Davis (1988). The objective of this approach to expert system implementation is to reason from first principles about the domain rather than from empirical associations. One of the key ideas in this approach is to use multiple representations of the digital circuit (both functional and physical structure) in troubleshooting applications. The approach is also similar to the multiple levels of abstraction in modeling of mental strategies for fault diagnosis problems Rasmussen (1985).

Inference Strategies

The inference control strategy is the process of directing the symbolic search associated with the underlying type of knowledge represented in an expert system: antecedents of IF-THEN rules, nodes of a semantic net, or collection of frames. In practical expert system applications, the blind search is an unacceptable approach due to the associated combinatorial explosion. Search techniques can be basically grouped into three: breadth-first, depth-first, and heuristic. The breadth-first search exhausts all nodes at a given level before going to the next level. In contrast, the depth-first exhausts all nodes in a given branch before backtracking to another branch at a given level. Heuristic search incorporates general and domain-specific rules of thumb to constrain a search.

Expert systems employ basically two types of reasoning strategies based on the search techniques above: forward chaining and backward chaining. In forward chaining, starting from what is initially known, a chain of inferences is made until a solution is reached or determined to be unattainable. For instance, in rule-based systems, the inference engine matches the left-hand side of rules against the known facts and executes the right-hand side of the rule that is activated. In contrast, backward-chaining systems start with a goal and searches for evidence to support that goal. Pure forward chaining is appropriate when there are multiple goal states and a single initial state whereas backward chaining is more appropriate when there is a single goal state and multiple initial facts. Many expert systems utilize both forward and backward chaining.

3.4 Review of Genetic Algorithms

Figuratively, genetic algorithms Goldberg (1989) are defined as search algorithms based on the mechanics of natural selection and natural genetics. Mathematically speaking, they are search algorithms that use random choice as a tool to guide a highly exploitive search through a coding of a parameter space. The focus of the algorithm is not perfection (or optimum) but satisfaction. Moreover, the algorithm is not meant to apply to one specific problem, but a set of problems for which little knowledge or "properties" of the problems are available. Consequently, the judgment of a genetic algorithm is not whether it can find an optimum point in one problem, but whether the algorithm can do a better job than other types of algorithms in a wide variety of problems.

The major differences between genetic algorithms and traditional search algorithms are: direct use of a coding, blindness to auxiliary information, search from a population, and randomized operators. Specifically, genetic algorithms abstract different kinds of search problems (decision, learning, optimization) into a common problem: exploitation of the best string representation among a set of feasible strings. Since the algorithm works at the coding level, and does not use functional properties of the problem, it cannot be fooled when there exist some tricky properties, or breaks down when no knowledge of properties is available. Moreover, genetic algorithms work with a population of possible search points instead of only one, therefore reducing the risk of being stuck at a local minimum. Finally, the transition rules in the algorithm are stochastic; the algorithm uses "calculated chances" to achieve the better solution.

Basic Genetic Operators

In genetic algorithms, the feasible values of search variables (states) are coded into strings of characters (genes), each encoding a piece of information about the variable. Associated with each string is a unique fitness value that indicates the performance of the string (search state). Given an initial population of strings, reproduction, crossover, and mutation are three basic operations that are used in genetic algorithms to guide the exploiting search. The reproduction corresponds to Darwin's "survival of the fittest", implying that strings with higher values have higher probabilities of contributing offspring strings.

Reproduction begins with determining each individual string's relative fitness within the current population. Once the relative fitness is determined, strings from the current population are *randomly* chosen to be parent strings in the reproduction process by using the strings' relative fitness as probability distribution function. After the reproduction strings are chosen, crossover and mutation operators are then conducted to produce new offspring strings.

Crossover is a "normal" reproduction process where two new strings are generated by "crossing over" their parent strings. As an example, if at random two parent strings are chosen to be 101 and 000 and crossing happens at the last code, the resulting offsprings due to the crossover will be 100 and 001. This random mixing of genetic codes provides a highly effective randomly search through the solution-space.

Mutation, on the other hand, is an "occasional" reproduction process where random alteration of the value of offspring string is performed. For example, from the parent strings 101 and 000, mutation may produce a new string 010, albeit generate a new genetic string of genes different from the parent strings.

With the above mentioned three operators, common information in the string representations of the problem are implicitly utilized and exploited during the search process. This concept consists of a similarity template (schema) and the building block hypothesis. Simply speaking, the schema is a template describing the similarities between the strings. The schema is implicitly used to guide an effective search based on a belief that the common genes among highly fit strings must be the "good" genes, which are used as building blocks in the formation of the next generation of genes.

The Fundamental Theorem of Genetic Algorithms, or *Schema Theorem* of Equation 3.1.1, governs the hypothesis that highly fit, low order, schemata with short defining length (known as building blocks) receive exponentially increasing trials in successive generations:

$$m(H,\ t+1) \geq m(H,\ t) \cdot \frac{f(H)}{\bar{f}}\left[1 - p_c \frac{\delta(H)}{l-1} - o(H)\, p_m\right] \tag{3.1.1}$$

where:

$f(H)$ = average fitness of strings representing schema H at time t

$\bar{f}$ = average fitness of entire population = $\frac{\sum f_i}{n}$

$m(H,\ t)$ = m examples of schema H at time step t

p_m = probability of mutation

p_c = probability of crossover

$o(H)$ = order of schema (number of fixed string positions)

$\delta(H)$ = defining length (distance between first/last fixed string position)

The merit of the genetic algorithms is their adaptability to various problems, and their little requirement on the knowledge of the problem--an opposite of the AI approach. Moreover, the focus of the genetic algorithms is not perfection (or optimum) but satisfaction. The algorithms are not meant to apply to one specific problem, but a set of problems for which little knowledge or "properties" of the problems are available. Consequently, the judgment of a genetic algorithm is not whether it can find an optimum point or not in one problem, but whether the algorithm can do a better job than other types of algorithms in a wide variety of problems.

4. MITE OBJECT ORIENTED ARCHITECTURE

The MITE system architecture is specified according to the object modeling techniques (OMT) presented within (Rumbaugh, 1991). This provides a solid object class hierarchy from which MITE system objects can inherit properties and behavior methods.

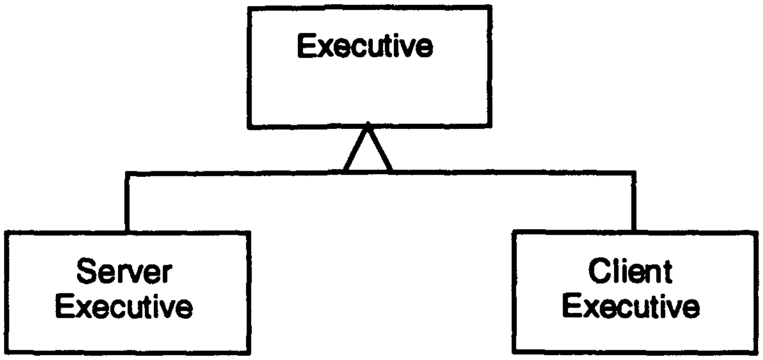

Figure 4-1: Executive Class Diagram

Figre 4-1 shows the two kinds of Executive classes, the Server Executive and the Client Executive. The △ notation (Rumbaugh,1991) indicates the "kind of" or inheritance heirarchy.

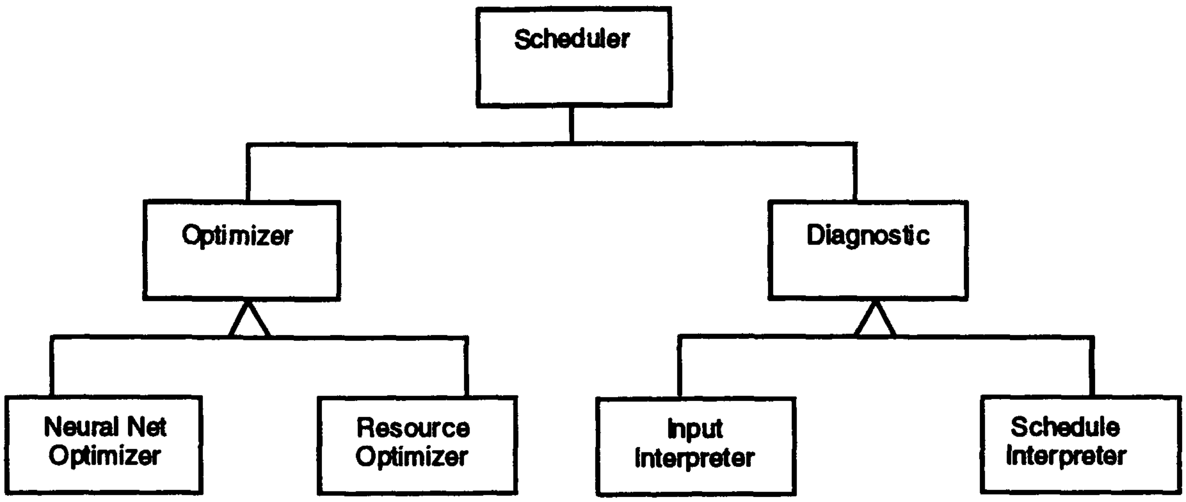

Figure 4-2: Scheduler Class Diagram

Figure 4-2 shows that the Scheduler class contains (or is associated with) an Optimizer class and a Diagnostic class. Each of which have subclasses as indicated.

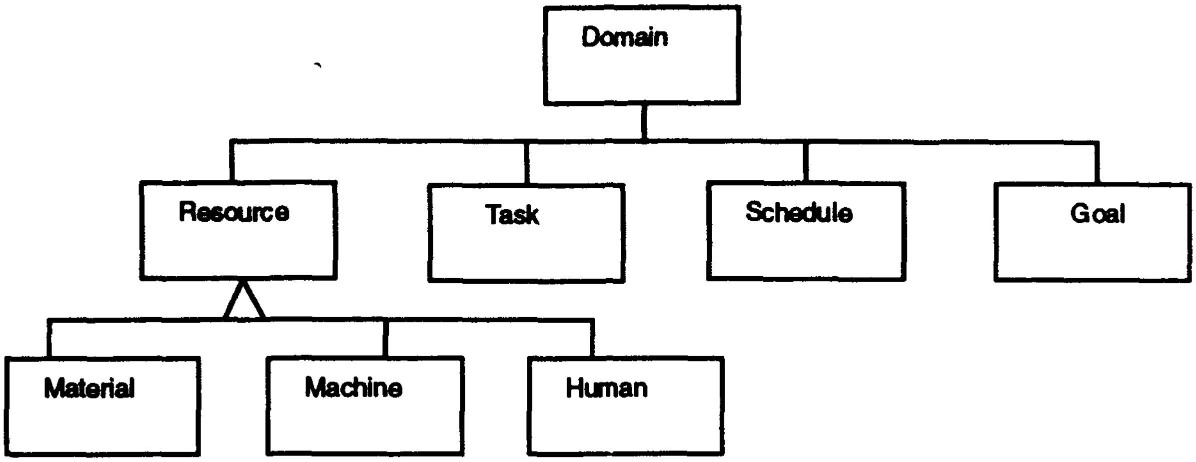

Figure 4.3: Domain Class Diagram

Figure XX shows the classes which are associated with the problem Domain class. These subclasses include Resource, Task, Schedule and Goal. the Resource class is devided into three specific kinds of resources, Material, Machine, and Human resources.

5. MITE OPTIMAL TASK ALLOCATION MODULE

5.1 An Intelligent Automated Resource Management and Task Planning System

In order for a large organization to achieve its goals effectively it has to intelligently manage its resources, and to identify and plan the different tasks it needs to perform to reach its goals in the most optimal way. The resources of the organization may include humans, machines and material. An intelligent resource management system should be able to adapt to changing environments and goals. Moreover, it may be able to give advice on which current constraints hinder performance and how to improve the performance by removing some of these constraints.

The functions that an intelligent resource management system may perform include:

- The optimal scheduling of tasks and allocation of resources subject to the different task and resource constraints.
- The identification and fixing of both resource bottlenecks and under-usage. For example the system has to recognize an impossible set of tasks given the timing and resource constraints. Moreover, it may suggest different alternatives that can make the performance of the tasks possible or more efficient. These alternatives may include, for example, the retraining of human resources and the addition or removal of resources.

- Reactive planning and scheduling to compensate for unforeseen disturbances. This is important for example when the creation or disappearance of resources or tasks is unpredictable, or the occurrence of some tasks is conditional on the occurrence of a particular event.

- Learning of task and resource attributes from previous experiences. For example learning of resource capacities and skills and task requirements and payoffs.

- The evaluation and pricing of resources based on their added value and the criticality of their role in the organization.

In this section, we will try to examine some of the above functions and suggest different possible methods to automate these procedures in order to achieve optimal performance.

5.2 Optimal Scheduling and Resource Allocation

A general scheduling and resource allocation problem is defined as follows:

Given a group of jobs that can be divided into smaller tasks and a set of limited resources (e.g. machines, people), which can perform the tasks with different efficiencies; and given some objectives that we would like to optimize and certain constraints that have to be met; how to schedule the tasks on the different resources to achieve these objectives optimally. This is a general problem that is present in almost any large organization. For example, similar problems arise in manufacturing, construction management, parallel computer processor scheduling, large network management, real-time systems and large project management. Military applications may include, for example, the optimal deployment of forces.

There are three components to the task allocation problem that govern the choice of a good schedule:

a. Objective / Goal

The overall objective or goal of the tasks. Examples of simple objectives include: maximizing payoff, minimizing tardiness or makespan and minimizing cost.

b. Tasks Definition:

The nature of the tasks and their attributes. Difficulties of tasks, expected times for task completions, required resources, task priorities, deadlines and task dependencies are among the important attributes that have to be taken into consideration when determining task scheduling.

c. **Resource Definition:**

The different resources available and their attributes. These may include cost of resources, their skills and efficiencies in performing the different tasks.

The different task and resource attributes represent both hard constraints that have to be met for a feasible task and resource allocation, and soft constraints that represent preferences. Examples of hard constraints include deadlines, task dependencies and the availability of the right resources. Soft constraints may include the preference that a task should be better performed at a particular time for best results, or that some tasks should be better done concurrently or within a given time window apart.

5.3 Solution of the optimal resource allocation problem

Assuming all the attributes of the tasks and resources are known a priori and are deterministic, there are many different ways for task scheduling and resource allocation. These methods can be grouped into two main groups:

- **AI heuristic search techniques**
- **optimization methods**

However, it must be emphasized that the optimal resource allocation problem belongs to a class of problems known as NP-hard. This means that there is no known algorithm which can find the globally optimal solution in a polynomial function of the size of the problem. In general the amount of time required to find the globally optimal solution is likely to grow exponentially with the size of the problem. This is due to the fact that there is no known way of knowing the optimal answer without enumerating almost all the possible alternatives. However, there exists a number of good search techniques which are able to find approximately optimal actions in a reasonable time.

Here we present some of the currently used search techniques for resource allocation and discuss their advantages and disadvantages. The following issues are addressed when comparing the different search algorithms:

- The time required to obtain a solution
- The quality of the solution obtained
- The repeatability of the solution
- Generality of the search procedure.

- Ability to adapt to changing knowledge base and changing objectives
- Scalability of the search method

5.3.1 AI Search methods

Artificial intelligence heuristic search techniques consist of first representing the scheduling problem as a tree structure, then use different heuristic search strategies to find a good path through the tree. Each node in the tree structure may represent, for example, a partially determined schedule and the branches originating from the nodes may represent the different available options at this particular state. A cost or utility may also be associated with each branch. To explore all possible paths from a starting node to a final node may be a huge task for even reasonably sized problems. AI search techniques employ different strategies to limit the search to the most promising paths. One such strategy, the branch-and-bound algorithm, uses domain specific knowledge to put lower bounds on the expected cost of the different paths, and only traverses the most promising ones Winston (1979). The paper by Cheng, Diamond and Lin Cheng, Diamond and Lin (1993) provides an example of the use of the branch-and-bound algorithm for schedule optimization. In that paper, domain specific knowledge is used to determine the sequence of nodes and the bounds on the costs. A recent review by Nelson and Toptsis Nelson and Toptsis (1992) describes different AI search algorithms that can be implemented both in serial and parallel machines.

Rule-based techniques for scheduling and schedule evaluation have also been explored by many researchers. One promising approach that has been recently explored is the use of fuzzy rules. Hatono et al. Hatomo, Suzuka, Umano, et al. (1992) propose a system for flexible manufacturing scheduling which employs fuzzy rules to assign tasks to resources. Fuzzy rules are used to determine the quality of a schedule. The degree of influence of each decision on the schedule evaluation is computed and assignments are changed based on their degrees of influence.

5.3.2 Optimization-based methods

The basic idea behind optimization-based methods is to represent the goal of the resource allocation problem using a utility or a cost function. Next, this function is optimized subject to different equality and inequality constraints that exist due to the task and resource attributes. Many optimization techniques have been explored for solving different types of scheduling and resource allocation problems with varying degrees of success (Rogers and White Jr (1991); Looi (1992); Uckun, Bagchi and Kawamura (1993); Eberhardt, Daud, Kerns, et al. (1991); Vaithyanathan and Ignizio (1992); Johnston and Adorf (Scheduling with Neural Networks- The Case of the Hubble Space Telescope)). Among these techniques, the most currently popular approaches are:

- Mathematical programming

- Simulated annealing
- Boltzman machines
- Genetic algorithms
- Hopfield neural networks and its variations
- Competitive neural networks.

In some of these optimization techniques such as genetic algorithms and simulated annealing, it is very hard to represent the constraints explicitly. This problem is circumvented by adding to the cost function a large penalty term if the constraints are not satisfied. For the case of genetic algorithms, the fitness of a chromosome is reduced proportional to the degree of constraint violation. The mutation and crossover rules generally do not incorporate the constraints. In general, the final solution obtained for methods which use a penalty to represent the constraints need not be a valid solution. Some optimization techniques, such as mathematical programming, simulated annealing and Boltzman machines, have the ability to converge to the globally optimal solution although this may not be practical in large applications due to the very long time it takes to converge to the global optima.

Optimization methods tend to be more general than heuristic search techniques and can be applied to any problem that can be formulated as a cost or utility optimization. However, due to the lack of problem specific knowledge, the optimization-based methods tend to take longer than heuristic search methods on a serial machine to obtain comparable solutions.

In this report, we develop an optimization-based scheduling and resource allocation technique which utilizes a type of competitive recurrent neural network Eberhardt, et al. (1991). We chose this particular network configuration due to its ability of representing many of the constraints explicitly.

5.4 Competitive Recurrent Neural Network Solution

Before presenting the algorithm, we will first formulate the scheduling problem that we would like to solve. The time horizon of the scheduling problem is divided into discrete units (e.g. seconds, minutes or days) which depend on the scale of the problem and the desired precision of the schedule required.

Define the following variables:

- N the number of tasks that need to be scheduled

- $y_i(k)$ a binary variable which represents the starting time of task i. A value of 1 means that the starting time of task i is scheduled at time unit k.
- t_i time units required to complete task i
- $R = \{r_1, r_2, ... r_M\}$ set of available resources
- $Q_i = \{r_{i1}, r_{i2}, ...\}$ set of resources required to perform task i
- $A_j = \{t_{j1}, t_{j2}, ...\}$ set of tasks that resource j can perform
- $s_j(k)$ utilization of resource j at time step k
- $b_i(k)$ a priority function for each task. This priority function should reflect task importance, deadlines and payoffs. For example, if a deadline for task i occurs at time unit k, $b_i(k + j), j > 0$ should be set to a large negative value, representing a large cost. High priority tasks should be represented by large positive values. Examples of different priority functions $b_i(k)$ are shown in figures 5.4-1 and 5.4-2:

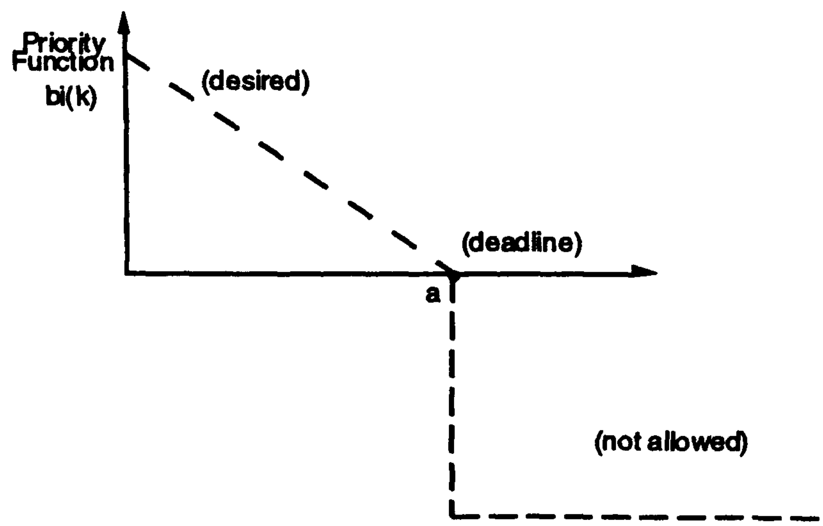

Figure 5.4-1: Deadline Priority Function

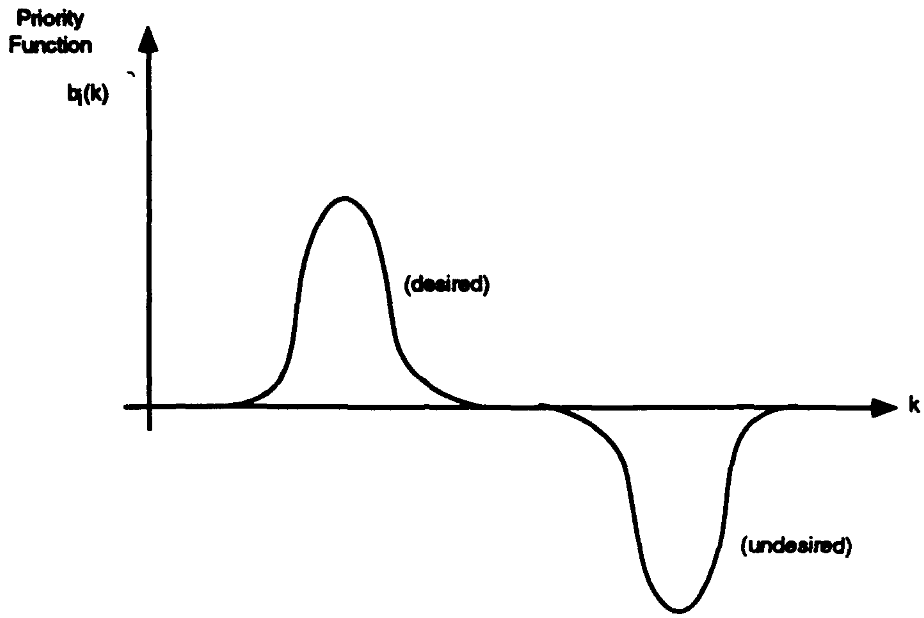

Figure 5.4-2: Time window Priority Function

- $W_{ij}(k, l)$ represent the desirability of the occurrence of task j at time unit l given that task i occurs at time unit k. For example, if task j depends on task i, and task i takes 2 units of time to complete from the starting time, then $W_{ij}(k, l)$ should be a large negative value for $l < k + 2$ and should be a positive value otherwise. This is shown schematically in figure 5.4-3. On the other hand, if it is desired to schedule two tasks some k time units apart, the corresponding W_{ij} should be set to a large positive value. If there are no dependencies, this variable may not be used.

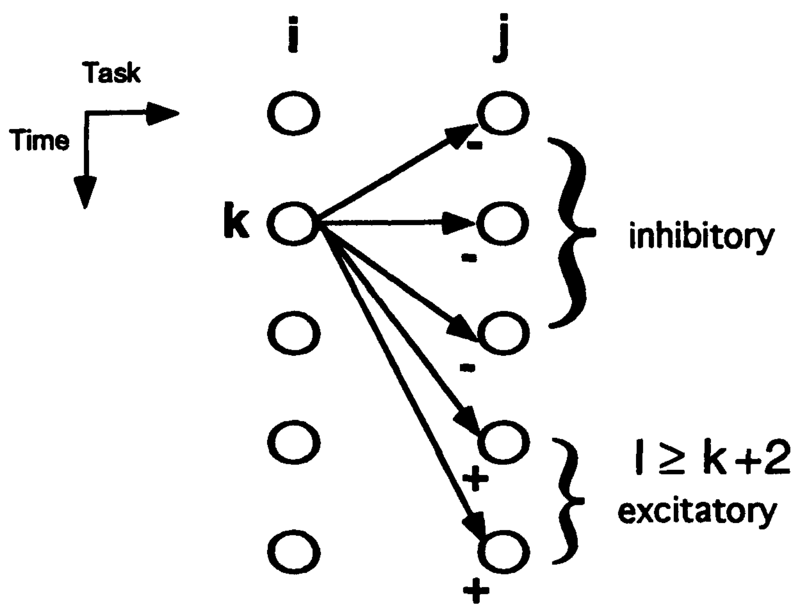

Figure 5.4-3: Desirability of Task Occurrence

There are two sets of constraints that are not represented by the parameters $b_i(k)$ and $W_{ij}(k,l)$. These constraints are:

$$\sum_k y_i(k) = 1 \qquad \text{for all } i = 1 \text{ to } N \qquad (1)$$

$$s_i(k) \leq r_i \qquad \text{for all } i = 1 \text{ to } M, \text{ for all } k \qquad (2)$$

Constraint (1) means that each task should only be scheduled once and constraint (2) means that the utilization of any resource at any time step should be less or equal to the available quantity of that resource.

5.2.1 Task Scheduling with Resource Constraints

The processing units of the competitive neural network are arranged in two different subnetworks as shown in figure 5.2.1-1. The output of the different units of the task scheduling subnetwork represent the variable $y_i(k)$. The columns of the task scheduling matrix are connected to a resource constraint violation matrix, the columns of which are feedback to the rows of the task scheduling matrix. The patterns of the connections and the connection strengths are determined from the utilization of each resource by the different tasks. This can be determined from the sets Q_i and time units t_i. The connections between the resource constraint violation network and the task scheduling network are designed in order to satisfy constraint 2. The units of the resource constraint violation subnetwork are called k-m Winner-Take-All (k-m WTA) units Eberhardt, et al. (1991). Constraint 1 can also be enforced using similar units.

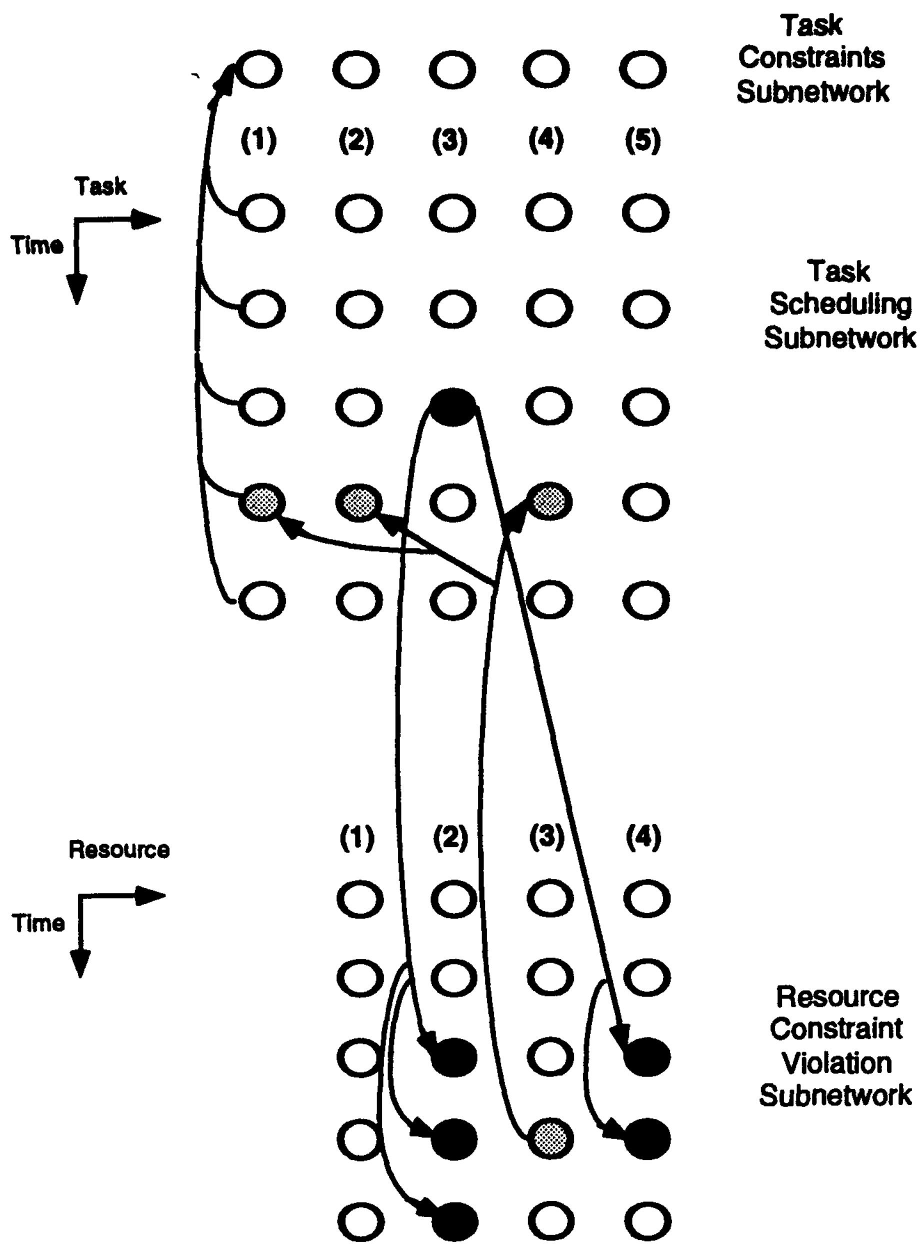

Figure 5.2.1-1: Competitive Recurrent Neural Network

Figure 3 shows only the resource constraint connections for Task 3 and shows only the connections which feedback from Resource 3. Inhibitory and excitatory connections within the Task Scheduling subnetwork are not shown.

The equation representing the output of each k-m WTA unit is shown in equation 3:

$$\tau_r \frac{dR_i}{dt} = g(\sum_A y_i(l), k_i, m_i) \qquad (3)$$

$$\text{where } g(\text{sum}, \text{min}, \text{max}) = \begin{cases} \text{max} - \text{sum} & \text{sum} > \text{max} \\ 0 & \text{min} \le \text{sum} \le \text{max} \\ \text{min} - \text{sum} & \text{min} > \text{sum} \end{cases}$$

The function g(sum, min, max) is called a windowing function. For the resource scheduling application, we do not specify a minimum of resource utilization and the function g( ) used should act like a threshold function. An example of a threshold function $g(Rsum_i(k), Rmax_i(k))$ is shown in figure 4 below, where $Rsum_i(k)$ represents the amount of resource i used at time k, and $Rmax_i(k)$ represents the maximum amount of resource i available at time k. Using the threshold function g( ), the rate of change of the output of the resource constraint violation unit is zero as long as the sum of the corresponding resources used does not exceed the maximum available. On the other hand, if the sum of the resources required to perform the schedule exceeds the available limit for a certain resource at any given time, the output of the corresponding resource constraint violation unit will increase until the task schedule is changed.

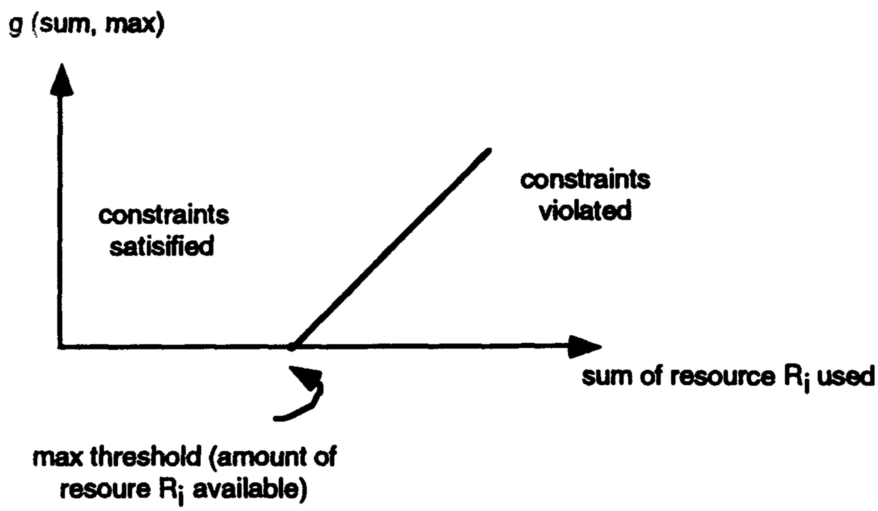

Figure 5.2.1-2: Windowing Function

The equations governing the dynamics of the task processing units are given by equation 4:

$$t_t \frac{dy_i}{dt} = - y_i(k) + f(a\, b_i(k) + bR_i(k) + gn_i(k) + qy_i(k) + \sum_{jl} W_{ij}(k,l) y_j(l)) \qquad (4)$$

Annealing is required to avoid local minima and to force the solution

into a zero one configuration. This is done by adjusting the parameter T of the sigmoidal function. The parameter T, also called temperature, is gradually decreased using for example an exponential schedule as shown in equation 5 :

$$T(t) = T_o \exp(-t / t_o) \qquad (5)$$

To control noise and settling time the hysteric annealing component $qy_i(k)$ is added to the task scheduling processing units Eberhardt, et al. (1991). The value of the variable q should be increasing with time in order to provide an increasing positive feedback to each processing unit and force the solution to a corner of the hypercube representing the output of the different units. An example of a possible schedule for the variable q is as shown by equation (6)

$$q(t) = (t / t_q)^2 \tag{6}$$

t_q is a time constant that controls the settling time. The bigger it is, the longer it takes for the solution to reach an equilibrium and the better is the quality of the solution.

Task Scheduling Evaluation

The Competitive Recurrent Neural Network should settle in a valid solution, that is a solution which satisfies constraints 1 and 2 (assuming such a solution exists) if the different time constants are chosen appropriately. The time constants for the resource processing units should be much larger than those of the task scheduling units to prevent oscillation. The time constant for the annealing schedules should be slow.

5.4 Optimal Resource Allocation

The formulation of the task scheduling problem does not yet take into account the efficiencies of each resource and its cost in performing the tasks. The collective output of each of the resource types has only been considered thus far. In order to allow the network to optimize over the resource cost and task efficiencies, it is necessary to represent tasks which can be accomplished by various resource combinations as separate nodes within the Task Scheduling subnetwork. (FIgure 5.4-1)

Resource	Cost	Task1 Efficiency
R_1	Medium	Medium
R_2	Medium	Medium
R_3	High	High
R_4	Low	Medium
R_5	Low	Low
R_6	Medium	Low

Figure 5.4-1: Resources which may be allocated to Task 1

For example a single task can be performed either by a skilled or an unskilled worker. Our method for implementing this is to consider a task that can be performed by two different resources as two different tasks with different properties (e.g. different desirability functions and different completion times). To illustrate this, we have identified three possible instances of Task 1:

- Task 1a: accomplished by R_1 and R_2

- Task 1b: accomplished by R_3
- Task 1c: accomplished by R_4, R_5 and R_6

Each instance combination carries with it a total cost based upon each resource's cost and task efficiency. In this case each processing unit in the task scheduling subnetwork will represent the scheduling of the particular task instances. We then have to add the additional constraint that the task has to be performed only by one of the available task instances. This can again be performed using a k-m WTA subnetwork. (FIgure 5.4-2)

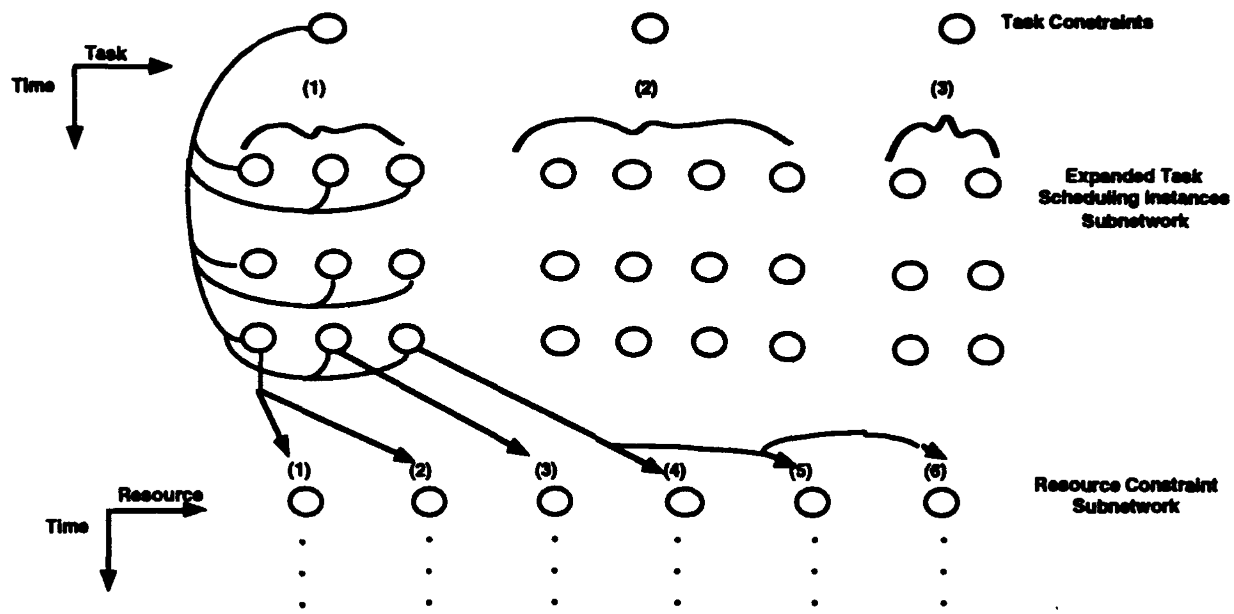

Figure 5.4-2: Expanded Subnetwork for Optimal Task Allocation

Optimal Resource Allocation Evaluation

For a limited number of task/resource combinations the above approach should not represent a computational problem. However, if the number of possible task/resource combinations is very high, this approach may not be practical. One way to solve the computational complexity problem in this case is to preprocess the task/resource combination and only enumerate the most attractive combinations. In this manner, heuristic and *fuzzy rules* are used to evaluate the different task/resource combinations.

6. MITE DIAGNOSTIC AND INTERPRETATION MODULE

If the competitive network fails to find a valid schedule, such a schedule may not exist due to some resource bottleneck or time limitation. In such a case we can examine the task schedule and resource constraint violation subnetworks to determine which items represent bottlenecks. For example if a particular resource could not be scheduled (i.e. the corresponding column of processing units of the scheduling subnetwork have all low values), we examine the inputs to each unit and determine the most influential one (i.e the input with the highest inhibitory value). We can then determine whether this is due to a particular resource scarcity, or an unfeasible deadline or both. The network interpretation may be performed using a network interpretation module which uses heuristic or fuzzy rules to make its decisions.

Associative networks together with some training using patterns/decisions pairs may also be applied. The function of the network interpreter is represented in block diagram below: (Figure 6-1)

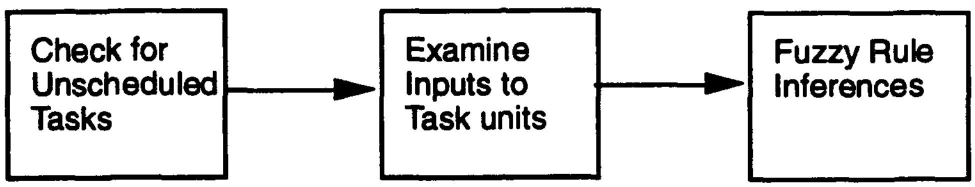

Figure 6-1: Network Interpretation

The three parameters P_1, P_2 and P_3 may represent for example inputs from task priorities, task dependencies or resource availability. The values for each parameter are fuzzified using different membership functions. The outputs D_1, D_2, ...D_n of the fuzzy system may represent different inferences. For example D_1 may represent that resources and dependency constraints are met at this time, but that deadline has passed.

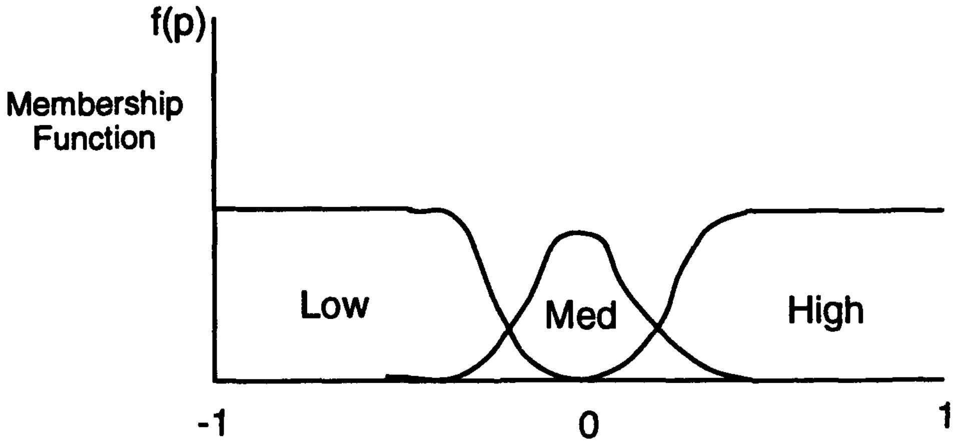

Figure 6-2: Fuzzy Membership Function

An example of a membership function with three different levels: low, medium and high is shown in figure 6-2.

In other cases, when resources are available, we may still be interested in improving some performance measure (e.g. minimize cost of resources, minimize time ...). This can be done by constructing a sensitivity of the performance measure to the different resource parameters. One possible way to perform this is to change a resource parameter, one at a time, and perform the optimization on the changed system and then compute the new performance measure. Resources should then be changed in the direction of improvement of performance. We can also use a simulation tool such as a petri net to examine the performance of the planned schedule and determined the performance sensitivity to different parameters Kapasouris, Serfaty, Deckert, et al. (1991).

7. ADAPTIVE RESOURCE AND TEAM MODELS

In many applications, resource and task attributes may not be known a priori. In such cases, these unknowns have to be inferred from prior performances on similar tasks. This can be done using either clustering techniques, such as an adaptive resonance network (ART network), or also using supervised feedforward backpropagation networks. The inputs to these networks will be some resource (or task) identifiers and the output should be the expected resource attributes.

To create the *Team Model*, ANNs are separated into high and low level classifiers Glover and Rao (1990)); a high level classifier network will process raw input data, while low level networks create or process feature sets (Figure 7-1).

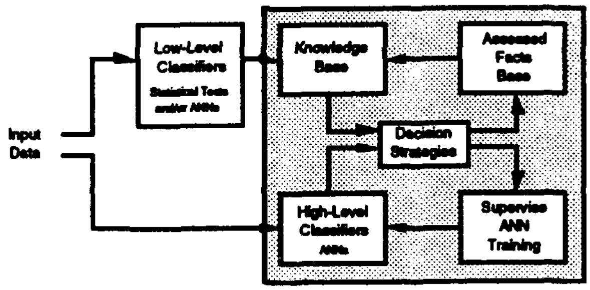

Figure 7-1: Adaptive Hybrid System for Team Modeling

Evolutionary Neural Paradigm:

In our research, we investigated the application of genetic algorithms (GAs) to perform the necessary model learning capabilities. A genetic algorithm was applied to the evolution of neural network paradigms. The GA operators of Reproduction, Crossover and Mutation were applied to the selection of neural network nodes, forward propagation functions and weight change functions.

The following structures were used to represent the neural network population:

Network:	NetID , TotalFitness
NeuronGene:	NeuronID, NeuronID, NeuronID, NeuronID,.....
Neuron:	NetID , NeuronID, Fitness , Activate, Stat
Connects:	NeuronID, NeuronID, NeuronID, NeuronID,.....
Propagates:	NeuronID, NeuronID, NeuronID, NeuronID,.....
Data:	Activate, Activate, Activate, Activate,.....
Memory:	Weight , Weight , Weight , Weight ,.....
Activation:	Object , Operator, Object , Operator,.....
Remember:	Object , Operator, Object , Operator,.....

Each Network in the population consists of a collection of generic Neurons which are identified with the network's NeuronGene. This NeuronGene is evolved using the GA techniques. After the network gene has been established for each network in the population, the training data is propagated through the network using each neuron's Activation gene string. Next the error signal is propagated back through the network using each neuron's Remember gene string.

The Operator and Objects which make up the neuron's Activation and Remember gene strings are specified as follows:

Operators: +,-,/,*,^,∫,Δ
^ = power
∫ = summation
Δ = change weights delta (Remember)

Objects: d,w,#
d = data
w = weights (Memory)
# = constant

The Δ operator appears only in the Activation gene string and serves to initiate the use of the Remember gene string which will be used to modify the neuron's Memory string containing connection weights.

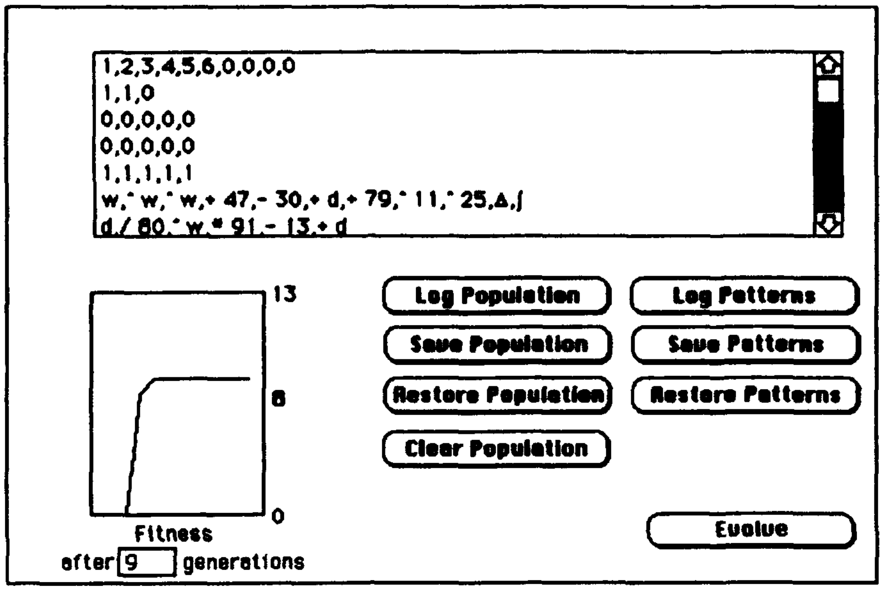

Figure 7-2: Genetically Evolving Neural Paradigm

Figure 7-2 shows a log of a sample genetically evolving neural network. The fitness of the network population is illustrated after 9 generations of network evolution. This GA method of evolving neural networks looks to be promising for problems which known networks solutions are not available. The GA will evolve a population of networks that will be capable of solving the problem on which it was trained.

7. IDENTIFICATION OF TASK-SHARING DOMAINS

7.1 Pedestal Mounted Stinger (PMS)/Avenger Team

The PMS/Avenger Air Defense Weapon System is a lightweight, day/night, limited adverse weather fire unit for countering the threat of low altitude high-speed fixed wing or rotary wing aircraft. Boeing (1988) The fire unit incorporates two turret mounted STINGER missile pods, a .50 caliber machine gun, Forward Looking Infrared (FLIR), Laser Range Finder (LRF) and Identification Friend or Foe (IFF). The fully rotatable, gyro-stabilized turret is mounted on the M998 High-Mobility Multipurpose Wheeled Vehicle (HMMWV). It can shoot missiles or machine gun on-the-move or from a stationary position with gunner in turret or at a remote location. It has on-board communications for both radio and intercom operations.

The MITE system can be applied to a team of Avenger units which must all operate together to accomplish a set objective. This requires the identification of the Tasks and Resources involved in the Avenger system in addition to specification of the team objective or goal. Here, we investigate each major part of the Avenger system (as specified in Boeing (1988)) and identify the corresponding Tasks and Resources as required for the task scheduling and optimal resource allocation modules.

Turret

Description: Pedestal Mounted STINGER/Avenger (PMS/Avenger) fire unit includes a rotating turret mounted on a based which is secured to HMMWV bed hard points. It may be removed from one HMMWV and installed on another PMS modified HMMWV in less than twenty minutes. PMS/Avenger turret has a cabin in which a gunner is positioned to operate the system. Located on each side of the turret is a launch beam each holding a missile pod.

Identified Resources:	R_1:	PMS/Avenger fire unit		
	R_2:	Turret		
	R_3:	Human Gunner		
Identified Tasks:	T_1:	Turret Installation	*requires:*	R_2, R_3
	T_2:	System Operation	*requires:*	R_1, R_3

Missile Pods

Description: Each pod, also called Standard Vehicle Mounted Launchers (SVML), holds up to four missiles and has an upper and lower access panel which is hinged or removed during missile reload. Each pod contains two argon bottles pressurized to 6000 PSi which can provide argon for approximately 40 missile cooldowns.

Identified Resources:	R_4:	Missile Pod (SVML)		
	R_5:	Missile		
	R_6:	Argon Bottle		
Identified Tasks:	T_3:	Missile Reload	*requires:*	R_3, R_5
	T_4:	Missile Cooldown	*requires:*	R_3, R_6

Missiles

Description: Missiles are installed in Standard Vehicle Mounted Launchers, without gripstocks but with slings attached. Missile electrical and coolant connectors are self-mated when missiles are installed and latched in pods. Missiles may be removed from pod and fired in MANPADS role if fire unit becomes disabled. Grip stocks and Battery Coolant Units (BCUs) are stored in a carrying case on the fire unit for use in converting missiles to MANPAD configuration.

Identified Resources:	R_5:	Missile		
	R_7:	Missile Sling		
	R_8:	Battery Coolant Units (BCU)		
Identified Tasks:	T_5:	Missile Installation	*requires:*	R_3, R_5

T6:	Missile Firing	*requires:*	R3,R5
T7:	MANPAD Conversion	*requires:*	R3,R5,R8
T8:	MANPAD Firing	*requires:*	R3,R5

Power

Description: Turret azimuth and elevation movement is accomplished by electric motors powered by batteries carried in base of fire unit connected slip ring assembly. HMMWV power system (generator and battery set) are paralleled with PMS/Avenger battery set. Fire unit battery is charged by 100 amp HMMWV generator through NATO power cable connected to base of fire unit.

Identified Resources: R9: Battery
R10: Generator

Identified Tasks: T9: Charge Battery *requires:* R9,R10

Machine Gun

Description: A .50 caliber machine gun for coverage of STINGER dead zone and for self-protection is attached to right hand launch beam below missile pod. It is shock mounted and fires at a maximum rate of 1100 rounds per minute. Two hundred rounds of ammunition are provided in a removable magazine. Gun is remotely armed from controls in turret and fired from either the turret or remote control unit.

Identified Resources: R11: Machine Gun
R12: Removable Magazine
R13: Remote Control Unit

Identified Tasks:

T10:	Arm Gun	*requires:*	R3,R11,R12
T11:	Fire Gun	*requires:*	R3,R11 ***or*** R13,R11

In this case, the Fire Gun Task 11 is composed of two task instances. The first relates to when the gun is fired manually the second relates to the firing from the remote control unit.

Optical Sight

Description: A projected reticle optical sight includes a transparent sight glass through which gunner looks to acquire, track, and perform target engagement. Driven reticles indicate aiming point of missile seeker at missile uncage. This confirms that missile seeker is locked onto same target gunner is tracking. Optical sight is attached to torque tube and follows elevation aiming point of pods.

Identified Resources: R14: Optical Sight

	R15:	Missile Seeker		
Identified Tasks:	T12:	Acquire Target	*requires:*	R3,R14
	T13:	Track Target	*requires:*	R3,R14
	T14:	Engage Target	*requires:*	R3,R15,R5

Sensor Package

Description: Sensor package includes a Laser Ranger Finder (LRF), and Forward Looking Infrared (FLIR) with an Automatic Video Tracker (AVT). This provides PMS/Avenger with target acquisition, automatic tracking, and ranging capability in battlefield environment, at night, and in adverse weather.

Identified Resources:	R16:	Laser Ranger Finder (LRF)		
	R17:	Forward Looking Infrared (FLIR)		
	R18:	Automatic Video Tracker (AVT)		
Identified Tasks:	T15:	Determine Range	*requires:*	R3,R16
	T16:	Automatic Tracking	*requires:*	R3,R18

Forward Looking Infrared

Description: FLIR receiver is attached to left launch beam directly below missile pod. Receiver follows elevation aiming point of pod. Gunner switches to narrow field of view. Gunner switches to narrow field-of-view using left footswitch in turret or a pushbutton switch on remote control unit to complete target acquisition. Switching to rain mode and increasing gain setting enhances FLIR acquisition capability during inclement weather.

Identified Tasks:	T17:	Rain Mode Switch
	T18:	Increase Gain

Laser Range Finder (LRF)

Description: LRF is a CO_2, dual aperture, eye-safe laser which is mounted on left launch beam below pod and beside the FLIR receiver. Range data from LRF is processed by PMS/Avenger computer and directly displayed on the control display terminal. This data also provides fire permit signal for missile and gun use.

Identified Resources:	R19:	Control Display Terminal
Identified Tasks:	T19:	Permit Fire
	T20:	Automatic Tracking

Automatic Trackers

Description: There are two auto-track systems available with PMS/Avenger system; Automatic Video Tracker (AVT) and missile autotrack. Tracking box for AVT is presented on FLIR display and may be activated whenever target is centered in tracker box. Tracker is controlled by right thumb switch on handstation which locks tracker onto target and automatically moves turret in azimuth and elevation until engagement is complete or tracker is deactivated. Missile auto-tracker locks on target using error signals from missile seeker once it is uncaged. To enable missile auto-tracker, OPERATE MODE track switch on gunner's console or remote control unit must be set to AUTO position. Once missile seeker is uncaged, seeker error signals provide input to drive turret automatically in azimuth and elevation. Missile auto-track is enabled only if FLIR AVT is deactivated.

Identified Resources:	R20:	Missile Autotrack		
	R21:	Error Signals		
Identified Tasks:	T19:	Lock On	*requires:*	R21
	T20:	AVT Deactivate	*requires:*	R18

Gyro Stabilization

Description: Turret drive is stabilized to automatically maintain missile pod aiming direction regardless of vehicle motion. Gyro detect changes in azimuth and elevation of carrier vehicle and provide drive signals to maintain pod pointing direction. Turret drive control can be operated manually from turret or remote control unit handstations or either automatic tracking system can be activated to maintain pointing direction of pods. Stabilized mode of operation is particularly helpful in maintaining pod pointing direction during on-the-move operations. Stabilized mode of operation is accomplished through switch action on gunner's console.

Identified Resources:	R22:	Gyro		
	R23:	Drive Signals		
Identified Tasks:	T19:	Maintain Pod Direction	*requires:*	R22,R23

Remote Control Unit (RCU)

Description: A RCU is provided as part of system allowing crew to conduct engagements from remote positions up to 50 meters from fire unit. RCU is normally mounted in HMMWV cab with 50 meter cable stowed behind HMMWV seats. RCU has a duplicate handstation, FLIR monitor and a pointer for visual aircraft acquisition. Also, all controls, indicators and communications necessary to acquire, track, identify and engage targets with both gun and missiles are provided.

Identified Resources:	R_{24}:	Remote Control Unit (RMU)		
Identified Tasks:	T_{20}:	Remote Acquire	*requires:*	R_{20}, ...
	T_{21}:	Remote Tracking	*requires:*	R_{20}, ...
	T_{22}:	Remote Identify	*requires:*	R_{20}, ...
	T_{23}:	Remote Engage	*requires:*	R_{20}, ...

Communications

Description: Radio transmissions can be initiated in the turret by either use of a footswitch or control switch on the CVC helmet. In the HMMWV cab, the driver can key the radio with control switch on the CVC helmet. When RCU is emplaced, either crewman can key the radio with control switch on the CVC helmet

Identified Resources:	R_{25}:	Radio

Identification Friend or Foe (IFF)

Description: IFF is provided by use of STINGER AN/PPX-3B Interrogator and PMS/Avenger unique IFF antenna mounted on front of turret. IFF interrogator is mounted in turret and is connected to an interface unit.

Identified Resources:	R_{26}:	STINGER AN/PPX-3B Interrogator
	R_{27}:	IFF antenna

Handstations

Description: Handstations are provided in turret and on RCU for turret control and conduct of engagements. Handstations contain all controls to activate turret rotation and pod elevation, select gun or missile, activate and uncage missile, activate laser, control FLIR auto-track, and fire weapons.

Identified Resources:	R_{28}:	Handstations		
Identified Tasks:	T_{24}:	Activate Rotation	*requires:*	R_{28}, ...
	T_{25}:	Select Weapon	*requires:*	R_{28}, ...
	T_{26}:	Uncage Missile	*requires:*	R_{28}, ...
	T_{27}:	Activate Laser	*requires:*	R_{28}, ...

Control Display Terminal (CDT)

Description: CDT is provided in turret for gunner interface with computer and to display essential information. Display provides messages indicating system faults, directing gunner actions, informs gunner of laser range readout in meters, percentage of battery charge remaining, azimuth and clock position of turret, and elevation of pods. CDT is used by gunner to: orient turret to Primary Target Line (PTL), set no-fire zones, perform BIT, read elapsed time indicator displays, and to input air density for machine gun lead angle and super elevation.

Identified Resources:	R29:	CDT		
	R30:	Computer		
Identified Tasks:	T28:	Orient Turret	*requires:*	R3, R29, ...
	T29:	Set Zone	*requires:*	R3, R29, ...
	T30:	Perform BIT	*requires:*	R3, R29, ...

Stinger RMP Missile

Description: Stinger RMP missile is an improvement on the basic Stinger missile in the RMP is unlikely to lock on false targets such as flares. The IEA on board the PMS has a programmable missile footprint that can be change to match mission threat.

Identified Resources:	R31:	RMP Missile		
Identified Tasks:	T31:	Program Footprint	*requires:*	R30, R31, ...

Fire Unit Operation Modes

There are five modes of operation for the fire unit, displayed in figure 7.1-1:

System Equipment		Mode			
	OFF	COMM	SAFE	RUN	ENGAGE
Lighting	Off	On/Off	On/Off	On/Off	On/Off
Heater/Ventilator	On/Off	On/Off	On/Off	On/Off	On/Off
Communications					
Intercom	Off	On	On	On	On
Radios	Off	On	On	On	On
Termiflex	Off	Off	On	On	On
FLIR	Off	Off	On/Off	On/Off	On/Off
IFF	Off	Off	On	On	On
LRF	Off	Off	On/Off	On/Off	On/Off
Computer	Off	On	On	On	On
Missile Electronics	Off	Off	Off	Off	On
Machine gun	Off	Off	Off	Off	On
Optical Sight	Off	Off	Off	Off	On

Turret Drive	Off	Off	Off	On	On
ECA Drive	Off	Off	On	On	On
Motors Status	Off	Off	Off	On/Off	On/Off
Lights	Off	Off	On	On	On
Canopy	Open/Closed	Open/Closed	Open/Closed	Closed	Closed

Figure 7.1-1: Fire Unit Mode Status

Knowledge of each mode state is necessary for the MITE system, and can be represented in an fuzzy rule base. Additionally the following mode Tasks are identified:

Identified Tasks:

T32: Off Mode
T33: Comm Mode
T34: Safe Mode
T35: Run Mode
T36: Engage Mode

7.2 General Project Management MITE Application

The MITE system investigated here can be applied to a variety of task scheduling and resource optimization problems which require a team of users working together over a computer network. Most large development projects consist of team members with corresponding costs and skills, who must use available resources to perform tasks, in order to accomplish an overall project goal.

8. FULL RESEARCH PROTOTYPE REQUIREMENTS

Charles River Analytics is identifying an object oriented intelligence framework which will provide fundamental building block objects for the development of intelligent agent systems which will operate across various networks and computer platforms. The framework includes neural networks, genetic algorithms, fuzzy logic, and proprietary Open Sesame!® learning capabilities.

The task scheduling and resource optimization technology developed within this MITE contract are being integrated into the design and development of our future intelligent agent products. Complete implementation of the fundamental MITE intelligence modules should be accomplished within the intelligence framework currently under development. This will allow the MITE system to be applied to a number of commercial applications in addition to specific defense related Avenger team applications. Furthermore, the classes and objects implemented for the MITE system will be capable of reuse and recombination for the development of optimization applications that have not yet been considered.

9. CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

In our Phase I study, we developed a multi-node, interactive, task-sharing, expert-instruction (MITE) system which takes advantage of the hybrid neural network / knowledge base approach within an object oriented architecture. In particular, our investigation has shown the following:

- We have demonstrated that the MITE system architecture can be designed according to object oriented methodologies, and have developed a class diagram which illustrates an object class hierarchy for MITE systems.

- We have identified the important classes and associated object properties needed for representation of team objectives, tasks to perform, and required resources.

- We have shown how competitive neural network architectures will be applied to the problem of task scheduling, how those architectures must be expanded to provide optimal resource allocation, and demonstrated how the automatically generated schedule is interpreted by a fuzzy logic rule base.

- We have identified various task-sharing domain problems which are applicable to the hybrid MITE system. The PMS / Avenger system is investigated for potential communication and task scheduling coordination between remotely located Avenger teams which must work together to accomplish a common objective. Corresponding tasks and resources are identified for application of the MITE system to the Avenger team.

- We have implemented a prototype of a genetically evolving neural network paradigm by applying the genetic algorithm (GA) operators to gene strings which represent neuron connection paths, activation functions and weight change functions.

9.2 Recommendations

On the basis of our successful Phase I results, we recommend the following:

- The development of a full-scale research prototype of the hybrid multi-node, interactive, task-sharing, expert-instruction (MITE) system. This involves expanding the object class diagrams, developing state transition scenarios, and implementing the object oriented architecture in our in-house developed hybrid development framework. The object oriented intelligence framework provides fundamental building block objects for neural networks, genetic algorithms, fuzzy logic, and Open Sesame!® learning capabilities.

- The refinement of the hybrid neural network and knowledge base architecture developed in Phase I in order to handle the faults which may occur during multi-node network operations. Our experience with fault-tolerant systems will provide the basis for this effort.

- The incorporation of additional neural network paradigms in order to improve the modeling capabilities the system requires in order to properly reallocate resources after diagnosing performance of the resources on the tasks.

- The in depth application of the MITE system to the team of networked Avenger weapons, in addition to a extended development effort to apply the MITE task scheduling methods to general project management and resource allocation.

10. REFERENCES

Anderson, J.A. and Rosenfeld, E. 1988. *Neurocomputing—Foundations of Research*. Cambridge, MA: The MIT Press.

Boeing. 1988. *PMS/Avenger Handbook.*

Caglayan, A.K., Walker, J.L. and Riley, E.W. 1992. *SDL-An Object Oriented Compiler Approach to Real-Time Knowledge Base Systems*. Charles River Analytics Inc. R89011 (August).

Carpenter, G.A. and Grossberg, S. 1987a. "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine." *Computer Vision, Graphics, and Image Processing*, Vol. 37: 54-115.

Carpenter, G.A., Grossberg, S. and Rosen, D.B. 1991. "ART2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition." *Neural Networks*, Vol. 4.

Charles River Analytics Inc. 1992. *SDL User's Manual.* Charles River Analytics Inc. (March).

Cheng, T.C.E., Diamond, J.E. and Lin, B.M.T. 1993. "Optimal Scheduling in Film Production to Minimize Talent Hold Cost." *Journal of Optimizatoin Theory and Applications*, Vol. 79, No. 3: pp. 479-492.

Davis, K. 1988. "Diagnostic Expert System for the B1B." *IEEE AES Magazine,* (April).

Eberhardt, S.P., Daud, T., Kerns, D.A., et al. 1991. "Competitive Neural Architecture for Hardware Solution to the Assignment Problem." *Neural Networks*, Vol. 4: pp. 431-442.

Glover, C.W. and Rao, N. 1990. "Hybrid Neural Network and Rule–Based Pattern Recognition System Capable of Self–Modification." *Applications of Artificial Intelligence VII*, Vol. 1293.

Goldberg, D.E. 1989. *Genetic Algorithms*. Reading, MA: Addison-Wesley Publishing Company, Inc.

Gonsalves, P.G. and Caglayan, A.K. 1992. *A Hybrid Neural Network/Expert System Approach to Multiple Target Recognition*. Charles River Analytics Inc., Cambridge, MA. Report No. R91121 (January).

Hatomo, I., Suzuka, T., Umano, M., et al. 1992. "Towards Intelligent Scheduling for Flexible Manufacutring: Application of Fuzzy Inference to Realizing High Variety of Objectives." *Japan/USA Symposium on Flexible Automation*, Vol. 1 - ASME.

Hertz, J., Krough, A. and Palmer, J.P. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA. Lecture Notes: Santa Fe Institute Studies in the Sciences of Complexity: 1991. Volume 1:

Hopfield, J.J. 1984. "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons." *Proceedings of the National Academy of Sciences*, Vol. 81: 3058-3092.

Johnston, M.D. and Adorf, H.M. Scheduling with Neural Networks- The Case of the Hubble Space Telescope. "Computers in Operations Research." *19*, Vol. 3/4, No. pp. 209-240.

Kapasouris, P., Serfaty, D., Deckert, J., et al. 1991. "Resource Allocation and Performance Evaluation in Large Human-Machine Organizations." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3.

Looi, C.K. 1992. "Neural Network Methods in Combinational Optimization." *Computers in Operations Research*, Vol. 19, No. 3/4: pp.

Mazzu, J.M. and Caglayan, A.K. 1992. "A Hybrid Neural Network System for Robotic Object Recognition and Pose Determination." *International Simulation Technology Conference*. Houston, TX.

Mazzu, J.M., Caglayan, A.K., Allen, S.M., et al. 1992. *A Hybrid Neural Network Knowledge Base System for Intelligent Structural Monitoring*. Wright-Patterson AFB. R89171 (July).

Mazzu, J.M., Caglayan, A.K. and Gonsalves, P.G. 1992. *A Neural Network Approach to Nuclear Plant Monitoring*. Charles River Analytics Inc. R91141 (May).

Mazzu, J.M., Snorrason, M.S. and Caglayan, A.K. 1992. *A Hybrid Neural Network Expert System Approach to Remote Sensing*. Charles River Analytics. R91391 (August).

Nelson, P.C. and Toptsis, A.A. 1992. "Unidirectional and Bidirectional Search Algorithms." *IEEE Software*, Vol. March: pp. 77-83.

Rasmussen, J. 1985. "The Role of Hierarchical Knowledge Representation in Decision Making and System Management." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 2.

Rogers, R.V. and White Jr, K.P. 1991. "Algebraic Mathematical Programming and Networks Models of the Deterministic Job-Shop Scheduling Problem." *IEEE Tranactions on Systems, Man and Cybernetics*, Vol. 21, May/June, No. 3.

Rumbaugh, J., Blake, M., Premerlani, W., et al. 1991. *Object Oriented Modeling and Design*. Prentice Hall, NJ.

Uckun, S., Bagchi, S. and Kawamura, K. 1993. "Managing Genetic Search in Job Shop Scheduling." *IEEE Expert*, Vol. October: pp. 15-24.

Vaithyanathan, S. and Ignizio, J.P. 1992. "A Stochastic Neural Network for Resource Constrained Scheduling." *Computers in Operations Research*, Vol. 19, No. 3/4: pp. 241-254.

Warren, K.C., Goodman, B.A. and Maciorowski, S.M. 1993. *A Software Architecture for Intelligent Tutoring Systems*. The Mitre Corporation. (August).

Winston, P.H. 1979. *Artificial Intelligence*. Reading, MA: Addison-Wesley.

Zacharias, G.L. and Gonsalves, P.G. 1990. *Optimal Aircrew Task Allocation Method.* Charles River Analytics Inc. R90091 (Dec).

Zacharias, G.L. and Riley, E.W. 1992. *Timeline Analysis for Behavioral Representation of Datalink Operations*. Charles River Analytics Inc. R90331 (June).