Systems Research and Behavioral Science
Syst. Res. **17**, 149–162 (2000)

■ *Research Paper*

Systems Research, Genetic Algorithms and Information Systems

Sohail S. Chaudhry[1]*, Michael W. Varano[1] and Lida Xu[2]

[1]*Department of Decision and Information Technologies, College of Commerce and Finance, Villanova University, Villanova, Pennsylvania, USA*
[2]*Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China*

Darwinian evolution and genetics have spawned a class of computational methods called evolutionary algorithms, and in particular, genetic algorithms. These evolutionary strategies provide new opportunities and challenges with ever-increasing applications in industry. In this paper, we propose that the proper context for a basic unifying theory of evolution for the emerging debate on the similarities and differences between biotic evolution and evolutionary algorithms is systems science. Recent changes in technology, coupled with developments in the field of artificial intelligence, promote the growth of enabling technologies, such as intelligent systems, in which we integrate genetic algorithms. Genetic algorithms are integrated with other artificial intelligence tools using a cooperating intelligent subsystem, which is integrated into the information systems of the organization. A portfolio of examples illustrating the evolving and expanding applications of genetic algorithms is included, as well as our computational experience with several commercially available genetic algorithm software. Copyright © 2000 John Wiley & Sons, Ltd.

Keywords systems science; systems thinking; evolution; genetic algorithms; information systems; intelligent systems; computational intelligence; soft computing

INTRODUCTION

Organizations recognize that to remain competitive in today's business environment it is imperative to have a knowledge-based view of the firm. The knowledge-processing methods and technology that codify, organize and perform reasoning in knowledge-based systems are evolving into specialized components and tools that support managerial decision-makers. Knowledge-based decision support is provided by current, and evolving, applied artificial intelligence systems, such as, expert systems, fuzzy systems, neural networks and genetic algorithms, among others (Hayes-Roth and Jacobstein, 1994).

Recent changes in technology, coupled with development in the array of artificial intelligence techniques, have promoted the growth of enabling technologies, such as intelligent systems, and their applications in business, industry and

* Correspondence to: Sohail S. Chaudhry, Department of Decision and Information Technologies, College of Commerce and Finance, Villanova University, 800 Lancaster Avenue, Villanova, PA 19085, USA.

science. Fundamental principles from Darwinian evolution and genetics have spawned a class of computational methods called evolutionary algorithms and, in particular, genetic algorithms. Genetic algorithms are useful as search methods for problem-solving and modeling evolutionary systems. Evolutionary strategies provide both new opportunities and challenges as an emerging computational tool in a variety of applications and problem domains.

The focus of this study is that in system science we find conceptual foundations for evolutionary algorithms and, in particular, genetic algorithms. Genetic algorithms are an evolving important component of artificial intelligence which are used in intelligent systems. The complexity of business problems requires that intelligent systems become subsystems of the organizational information systems.

The paper begins with the discussion of evolutionary principles found in systems science linking biotic evolution and evolutionary algorithms. This is followed by a discussion of the structure and mechanism of genetic programming as evolutionary computing. Genetic algorithms are discussed to illustrate the advantages of artificial intelligence technologies for solving a wider class of complex problems compared to the problem domains of the more traditional management science models. The next section presents a knowledge-based view of the organization and its effects on the various types of information systems. The artificial intelligence technologies can only add value by interacting in a cooperative setting with the other information systems and users in the organization. A cooperative subsystem framework is presented that includes the artificial intelligence systems within an intelligent subsystem. Following this, we present a discussion of the expanding business applications of artificial intelligence technologies, such as, genetic algorithms, which suggests that implementation of the correct artificial intelligence tool in an application is a critical necessity to stay competitive. The expanding use of genetic algorithms suggests that for competitive necessity the organizations should plan for the development of intelligent systems. The conclusion summarizes the usefulness of genetic algorithms in solving a wide variety of problems in a complex environment.

EVOLUTION, EVOLUTION THEORY AND EVOLUTIONARY COMPUTING

Evolution and Evolution Theory

Search procedures based on the well-developed concepts and models from evolutionary biology are finding increasing applications in search, optimization and machine-learning problems across a wide variety of complex problem domains. These evolutionary algorithms use mechanisms or processes such as selection, crossover and mutations, similar to those found in the Darwinian natural selection biological model. By mimicking the natural selection process found in living organisms, computer scientists attempt to capture and adapt the successful controls and drivers of the natural evolutionary processes.

This type of reasoning by analogy suggests several unresolved questions. Do algorithms that implement strategies that are similar to evolutionary biology assist in problem domains where, for example, organizations are not species in the same sense that organisms are species? Does emulating evolution provide us with the creativity to generate new and complex solutions to difficult problems, not accessible by existing tools? Also, is there more than analogy in this strategy; that is, are biological evolution and evolutionary algorithms governed by the same or similar fundamental principles?

In the context of computer science the above questions become: does simulating evolution with evolutionary algorithms provide a tool for solving problems and assisting decision-making? And, does this method build solutions by generating complexity through combinations of tools (Banzhaf *et al.*, 1998, p. 92)? We propose that the proper context for a basic unifying theory of evolution for the emerging debate on the similarities and differences between biotic evolution and evolutionary algorithms is systems science.

The framework offered by systems science presents an intellectual basis for an investigation into these questions and provides for similarities among the concepts, models and laws from various fields, especially for our focus evolutionary biology and computer science. Systems science is the proper context in which to investigate the interplay between the disciplines: mining, molding, adapting and refining any overlapping principles; thus minimizing the duplication of effort across different fields, as in our case, using natural selection as the basis for a programming algorithm.

In systems science we find the paradigms of evolutionary principles that operate at all hierarchical levels of evolution. Fundamental laws observed in chemical, molecular and cellular non-equilibrium systems may, for example, also govern organizations, information systems and evolutionary algorithms. The theory of wholeness (Klir, 1991, p. 28; Jackson, 1995) and evolutionary principles suggests that we need not reduce the discussion of genetic algorithms to those of biological evolution, but rather that both evolutionary algorithms and biological evolution are governed by general evolutionary principles operating at different levels. Research findings in the latter may suggest the way for advances in the former.

An evolutionary algorithm is a process that uses the principles of evolution as the structure mechanism for its design and implementation. The term *evolutionary* has been associated with algorithms that use only selection and mutation, while the term *genetic* is associated with algorithms that use selection, mutation, recombination and a variety of other mechanisms from nature (Goldberg, 1994). Evolutionary computing, often referred to as evolutionary algorithms, together with concepts from artificial intelligence, such as neural networks, fuzzy logic, etc., is sometimes referred to as computational intelligence or soft computing.

Based on simple models of organic evolution, evolutionary algorithms incorporate strategies similar to natural evolutionary processes. The algorithms are then applied to various types of problems modeled in the computer. The different types of evolutionary algorithms that have been developed are classified according to the following scheme: populations of solutions, innovation operations, conservation operations, quality differentials and selection (Banzhaf *et al.*, 1998).

In natural selection, species that adapt well to their environment survive. These are described as having a high degree of fitness. Each species can be differentiated by its specific genetic composition which is contained in its chromosomes. The genetic material, which accounts for the continual adaptation of a species to its changing environment, is acquired and preserved from one generation to the next using three operators: reproduction, crossover and mutations.

The evolutionary algorithm process begins by representing solutions as genotypes, genomes or chromosomes (Banzhaf *et al.*, 1998). Each solution candidate is then judged with respect to the type of problem to be solved. The quality of solution may be measured by an evaluation function specified in advance, or by an interactive subjective juror. Operators, such as innovation operators and conservative operators, are defined to generate variants of the rated solutions. The innovation operator, often called mutation, contains three parameters: its strength within a component of a solution, its spread in simultaneous application to components within a solution, and its frequency of application within the entire algorithm. Conservation operators are used to consolidate what is already 'learned' by individuals in the population.

There are many types of evolutionary algorithms, one of which is the genetic algorithm, the subject of this paper. In 1975, John Holland introduced a method of studying natural adaptive systems, and then designed artificial adaptive systems based on Darwinian natural selection and genetics. This method eliminates weak elements by exploring retention of optimal or near-optimal individuals, then recombining features of the fittest individuals to perhaps make better individuals. Genetic algorithms, an important predecessor of genetic programming, have proved useful both as a search method for problem-solving and for modeling evolutionary systems (Banzhaf *et al.*, 1998). Among the important characteristics of genetic algorithms is the ability to interface with existing application code, since the algorithm uses very little problem-specific information, but yet is capable of evolving complex structures. Genetic programming is an emerging alternative that builds on the genetic algorithm search strategies and machine-learning abilities.

Evolutionary strategies are another paradigm in evolutionary computation that uses evolution as a guiding principle. Discrete mutations were the first evolutionary variants to be applied within evolutionary strategies. This was accomplished by using random events to decide the direction of an optimization process strategy. The underlying purpose was to emphasize causality; for example, strong causes would generate strong effects. Translated into evolutionary strategies, large mutations should result in large jumps in fitness, while small mutations should result in small changes in fitness.

Another important evolutionary algorithm, and predecessor of genetic programming, was evolutionary programming. It uses the mutation operator to change finite state machines or finite automaton. Using random creation, mutation and fitness-based reproduction, for example, a better individual gets reproduced.

The advantage of algorithms using evolution-like structures and mechanisms comes from their evolutionary nature, in that genetic algorithms can self-teach, and can adapt well to changes in their environment. This type of algorithm can uncover previously unknown outcomes or solutions. Systems that build upon genetic algorithms are flexible, adaptable and capable of learning from their environment.

The idea of using evolution to solve problems and model natural phenomena is not a new concept. Simulations of genetically controlled, self-producing systems have been studied, for example, in von Neumann's self-reproducing automaton. Many later simulation models of evolution were studied in an attempt to determine the adaptation or optimization processes that were implemented in formal or artificial systems (Pattee, 1986). In all but one of the models, the process of natural selection is accomplished by criteria which are explicit and pre-established by the programmer.

The sources of algorithmic structure suggested by the evolutionary model may offer insights and opportunities that, as yet, are unknown. The unification of concepts of highly interconnected complex systems may explain developmental and evolutionary behaviors of systems (Klir, 1991, p. 135). To the extent that evolutionary algorithms and biotic evolution are similar processes, we have the opportunity to explore well-developed concepts and models from evolutionary biology. We can also investigate how these models translate, with modification, into evolutionary algorithms. If exactly analogous, one may be used to study the other. In this context evolutionary algorithms are of interest because of the important dissimilarities. Cellular evolutionary biology and genetics argue for a dynamic interplay between organisms and their environments, while, in computing, self-adaptation processes are implemented by algorithmic variation of parameters. It may not be feasible to seek direct genetic analogies since such a model may not exist. However, the overlapping concepts may offer a model for exploring complexities in other domains, such as organizational theory.

The interesting analogy between biological evolution and evolutionary algorithms, with the increasing diversity in each discipline, may be governed by the same or similar fundamental laws. The growing portfolio of computational experiences and applications with evolutionary algorithms, and genetic algorithms in particular, may shed light on the fundamental question of what determines the particular evolutionary pathway chosen by a particular evolutionary system.

The unsupervised learning capabilities of genetic algorithms proffer new opportunities and challenges. They act as aids to improve problem-solving and decision-making capabilities, especially as organizations shift their focus towards emerging developments in artificial intelligence and intelligent systems.

Genetic Programming as Evolutionary Computation

As mentioned earlier, genetic algorithms, the predecessor of genetic programming, is a newer form of artificial intelligence based on inductive learning technique which was first introduced by Holland (1975). A genetic algorithm can be viewed as having as its goal the development of systems that demonstrate self-organization and adaptation. This is accomplished by being exposed to the environment, similar to biological organisms. The integration of genetic algorithms and other techniques, such as neural networks and fuzzy logic, within the intelligent systems, has been the topic of interest to the researchers in the information science field during the last few years (Feng and Xu, 1996). Within this period, a variety of theoretical and application research papers have been published in the literature. Also, industrial applications of this complexity-based tool are on the rise in the business field. In this section, we provide a brief description of a genetic algorithm and how it works in an optimization problem, and later present a selected list of papers dealing with a variety of business applications, and a brief discussion on our preliminary experience with a number of problems using commercially available, spreadsheet-based genetic algorithm microcomputer software.

A genetic algorithm is a heuristic search procedure which is based on the natural process of evolution as in biological sciences. As this highly adaptive evolutionary process progresses, the population genetics evolves in a given environment according to the natural behavior in which the fittest survive and the weakest is destroyed. Thus, the genes from the adept donor will then propagate to another recipient during each successive generation, hence creating more adept offspring suitable for the defined environment. In optimization terms, the search algorithm improves the solution over generations as it progresses toward the optimum. Genetic algorithms have been successfully applied in solving a variety of optimization problems which are difficult to solve. These problems include the traveling salesperson problem, job-shop scheduling problems and routing problems, among others. For a more thorough coverage of genetic algorithms, the reader is referred to the excellent textbook by Goldberg (1989).

In terms of an optimization problem, the genetic algorithm approach is summarized as follows. At any given point in time, the genetic algorithm generates a population of possible candidate solutions. Initially, the population size is chosen at random. However, this choice typically depends on the characteristics of the problem. Each population component is a string entity of a chromosome — for example, a (0,1) bit string, which represents a possible solution to the problem. The population components are evaluated based on a given fitness function. Highly fit population components are given the chance to reproduce through a crossover process with other highly fit population elements by exchanging pieces of their genetic information. This process produces 'offspring' or new solutions to the optimization problem based upon the high-performance characteristics of the parents. A mutation process prevents premature loss of important information by randomly altering bits within a chromosome. This procedure continues until a satisfactory solution is achieved. A general structure for a simple genetic algorithm is shown in Figure 1.

Genetic algorithms can be integrated with other artificial intelligence technologies, such as fuzzy systems and neural networks. They are also known as hybrid intelligent systems, which support problem-solving in business organizations. The complexities of business problems, coupled with the competitive business environment, require information systems that support the creation and sharing of organizational knowledge and intelligence.

INTELLIGENT COMPUTING

Knowledge-Based View of Business Organizations

A first fundamental principle of systems thinking, as it relates to management, should focus on the interactions of parts and not on the actions of parts taken separately. When a system is taken apart it losses its essential properties and so does each of its parts. Performance of the system depends on how the parts fit, not on how they perform separately. Thus, different types of systems are needed at different levels of the organization, and by different functional specialities (Ackoff, 1971; Jackson, 1995).

Secondly, the concept of business problems and decision-making as specific to certain disciplines, such as management science or operations research, is the antithesis of system thinking. Solving each issue taken separately would improve the parts of the system taken separately, but could in fact adversely affect the whole. In a later discussion we build on this concept which leads to the structure of an integrated, cooperating intelligent subsystem, containing the tools of artificial intelligence, within the subsystems of the organization.

Knowledge-based systems are currently recognized by businesses as essential in the current competitive environment. The focus of processing data into information, with the usual delivery of information products and services, is changing. As with data and information, we must now include subsystems that support intelligent decision-making for the organization.

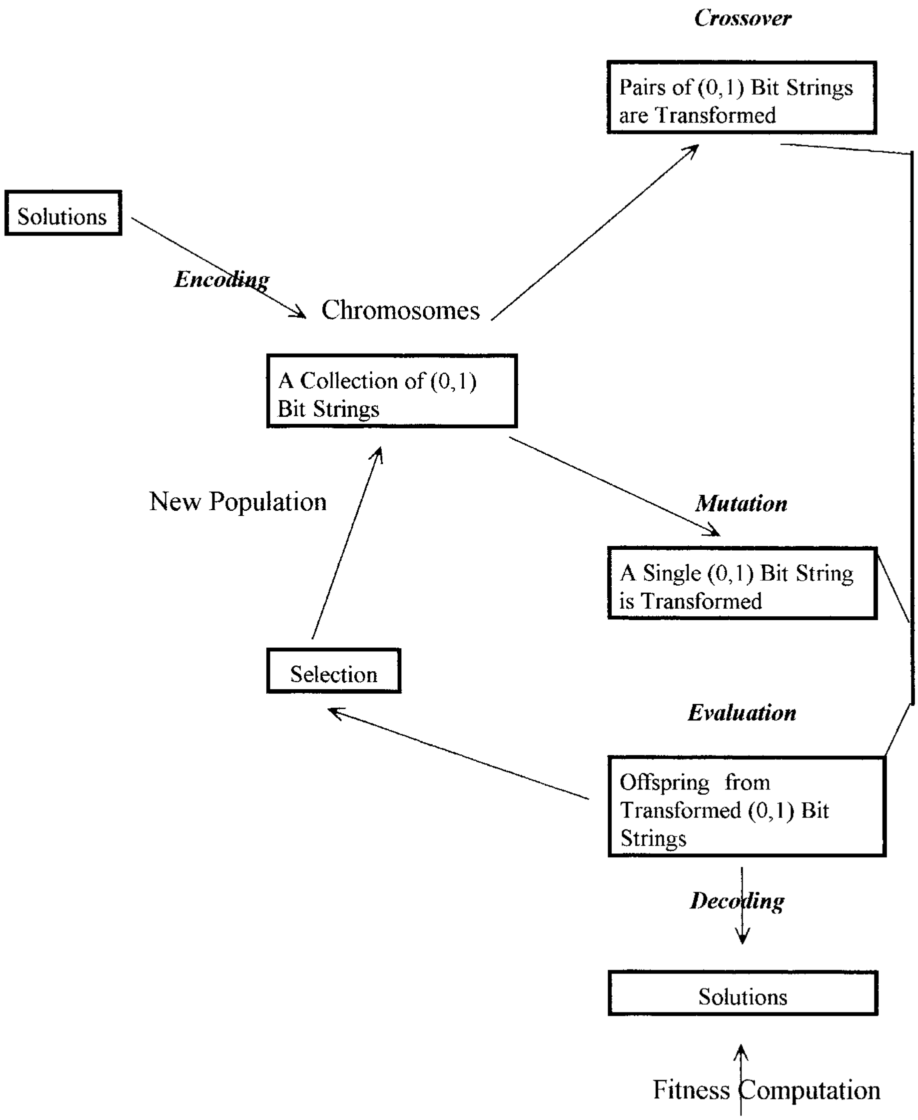

Figure 1. The structure of a genetic algorithm

The usual organizational hierarchy responsible for decision-making is found within the various functional areas of the firm. Information systems are found at all levels of decision-making within the functional areas of the firm. For example, the organization may be viewed as divided into strategic, management, knowledge and operational levels. Using this perspective, an organization may have executive support systems at the strategic level. They may have management information systems and decision support systems at the management level. They may also have knowledge work systems and office automation systems at the knowledge level, and transaction-processing systems at the operational level.

Systems at each level are then specialized to serve the major functional areas such as marketing, manufacturing, accounting, finance and human resources. Adopting this functional perspective of organizational structure reveals that information systems can be represented as given in Figure 2. This view emphasizes the physical system of the firm organized in terms of the traditional tasks performed together with their respective supporting functional information systems. An information system can be defined as a set of interrelated components that collect, process, store and distribute information to support decision-making and control in an organization. However, from a business or management perspective, an information system is more than an input–processing–output system operating in a vacuum.

To understand information systems, one must understand the broader organizational, managerial and information technology dimensions of systems, and their ability to provide solutions (Warfield and Christakis, 1987; Xu, 1991). This view of information systems recognizes the changing nature of the relationship of information systems and organizations, which results from the growing complexity and scope of using contemporary systems to support the full spectrum of institutional activities.

As organizational knowledge becomes a core productive and strategic asset, the success of the firm depends on its ability to gather, produce, maintain and disseminate knowledge. The literature support for this knowledge-based view of the firm includes: knowledge is the central productive and strategic assets of the firm (Arrow, 1973; Badaracco, 1991; Quinn, 1992); all physical capital is an instance of knowledge and

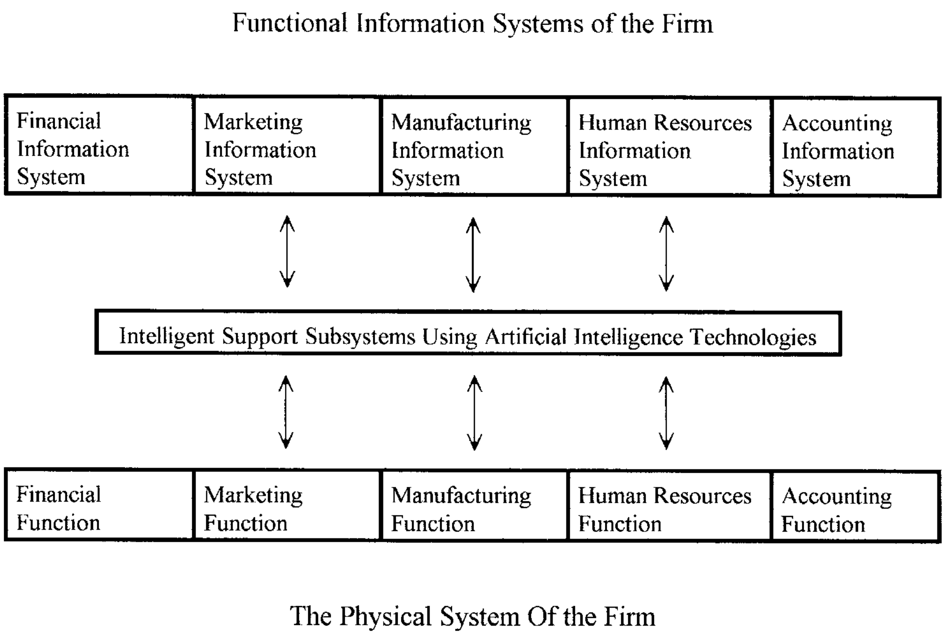

Figure 2. The functional view of the firm and its information systems

can be embedded in machines (Machlup, 1962; Boulding, 1966); the creation of value by the organization requires the application of different types of specialized knowledge (Quinn, 1992); and knowledge is the core productive and strategic assets of the firm and the success of the firm is linked to its ability to manage its knowledge (Laudon and Laudon, 1998).

The major types of information systems discussed above facilitate the flow of information and have organizational knowledge embedded within them. However, there are information systems that are specifically designed to help organizations create, capture and distribute knowledge. Among these are systems such as office automation systems which support distributing the organizational knowledge; knowledge work systems which support the activities of highly skilled knowledge workers as they create new knowledge; group collaboration systems which support the creation and sharing of knowledge; and artificial intelligence systems which provide organizations and managers with codified knowledge that can be used by others in the organization.

As information flows through the organization, it is now possible to develop very large and complex knowledge bases that require advanced specialized training to use and maintain (Holsapple and Whinston, 1996; Turban and Aronson, 1998, p. 824). Firms collect large amounts of knowledge related to problem-solving, servicing customers and product development, among others. This knowledge, if properly captured, codified, organized and stored, can be shared for the benefit of the firm. The process of systematically and actively managing and leveraging this store of knowledge in an organization is called knowledge management (Laudon and Laudon, 1998, p. 553). The technologies that support knowledge management are intranets, the Internet, data warehousing, groupware and artificial intelligence systems. This leads further to the integration of intelligent systems for the information systems of the organization as shown in Figure 2. An example illustrating the role of information and the use of intelligent agents in manufacturing systems is discussed by Chen (1995).

The array of systems above, including artificial intelligence systems, helps support the knowledge management function in an organization by (a) capturing and codifying knowledge, and using artificial intelligence technologies; (b) creating knowledge, using knowledge work systems; (c) sharing knowledge, e.g. group collaboration systems; and (d) distributing knowledge, e.g. using intranets.

Intelligent Systems

Systems that use artificial intelligence techniques are sometimes referred to as knowledge-based systems, implying an intelligent decision support subsystem (Dhar and Stein, 1997, p. 2; Feng and Xu, 1996). Although artificial intelligence applications are still maturing, the potential value for business applications is increasing. For example, FMC Corporation made a major commitment to artificial intelligence systems creating a 90-person artificial intelligence center. Recently the center expanded their applications to include neural computing and other cutting-edge artificial intelligence technologies (Turban and Aronson, 1998, p. 817). Systems based on artificial intelligence technology are able to learn natural languages, accomplish coordinated physical tasks (robotics), act as visual and oral perception systems and emulate decision-making (expert systems).

The traditional management science techniques may yield optimal solutions in some cases, but are restricted to structured problems. Decision support systems and expert systems do not produce optimal or near-optimal solutions. Neural computing can be used in some cases to derive optimal solutions, but only genetic algorithms, and their hybrid forms, provide robust methods for optimizing or nearly optimizing complex unstructured problems. Genetic algorithms refer to a variety of problem-solving techniques that are conceptually based on the process of evolution, providing solutions to certain types of problems. Commercial applications of genetic algorithms are emerging (see section on Applications below).

The early definitions of a decision support system identify it as a subsystem of the organization's information system which is intended to support managerial decision-makers in semi-structured decision environments (Xu, 1988; Dhar and Stein, 1997, p. 2). Since these early beginnings, decision support systems evolved along a path similar to the organization's information systems. With the growing complexity and scope of the applications, decision support subsystems began to use models from disciplines such as statistics, management science, finance and artificial intelligence.

The more traditional types of decision support systems, which use spreadsheets, mathematical optimization techniques and simulation, will continue to use these as the core tools. However, the growth of knowledge workers in the organization (Laudon and Laudon, 1998, p. 11) and the complexity of types of problems will require systems that incorporate intelligent systems. The techniques included in the traditional model-management subsystems may yield optimal solutions but they are typically confined to structured or semi-structured problems. Several recent decision support algorithms, including, for example, genetic algorithms and fuzzy logic, alone or in combination, provide powerful methods for solving very difficult problems that are not solvable with the traditional tools, either alone or in combination with other intelligent systems, such as neural networks.

The business case for managing organizational knowledge is becoming widely accepted and brings with it management issues such as a knowledge management strategy, organizational architecture and capability, planning for a technical architecture, and the ability to develop and leverage critical organizational knowledge to improve performance.

Although information technology applications cannot substitute for human skills and expertise, developments in artificial intelligence technologies are proving valuable in industrial and commercial applications (Hayes-Roth and Jacobstein, 1994, p. 31). Applications of artificial intelligence-based technologies are returning benefits such as large increases in speed of accomplishment of complex tasks, reduced errors, reduced costs, improved quality of decisions and improved customer service.

Once a knowledge base is constructed, most artificial intelligence programs using symbolic representation and manipulation function by using a search and pattern-matching technique. Given some initial start-up information, the artificial intelligence technology builds on two systems: the increasingly large and sophisticated rule-based expert systems, working inferentially from vast information bases; and learning systems, using techniques such as neural networks, genetic algorithms and fuzzy logic (Wright, 1996).

This paper will extend the framework for a decision support system presented in Dhar and Stein (1997). The framework will integrate an intelligent subsystem as a component of a data-model-driven subsystem of a knowledge management system. The concept of a genetic algorithm is introduced and integrated as a core tool in the intelligent subsystem. It is shown that a genetic algorithm, an important useful tool among the array of artificial intelligence technologies, enhances the organization's knowledge base, and assists the knowledge management function, by suggesting solutions to specific problems that are too massive and complex to be analyzed by existing algorithms. Each central building block, i.e. neural nets, fuzzy logic and genetic algorithms, can be used alone or in hybrid form. For example, the genetic learning method could be used for rule discovery in a large database, then fed into the conventional expert system, thus using the systems in series or in parallel. Examples and documented business applications using genetic algorithms, either alone, or as a hybrid system, are discussed to illustrate how managers match costs, benefits and capabilities of this technology with the knowledge management issues of the organization.

Integrating Genetic Algorithms into Intelligent Systems

Using Dhar and Stein's taxonomy of management information systems, a decision support system is divided into two types (Dhar and Stein, 1997). The 'model-driven' type evolves from an earlier concept of a decision support system as a system that uses models which accept data from the user performing 'what-if' analysis. For example, a portfolio manager could use an econometric model to determine how risk varies with certain market parameters. The second type of decision support subsystems are 'data-driven'. The functionality is to support condensing large amounts of data into a form that is useful to management (Dhar and Stein, 1997, p. 5)

The two-component model focuses on data and/or information processing and may become deficient as decision scenarios become more complex. Knowledge is now recognized as an important organization resource, and knowledge-level systems, especially in the form of workstations and office systems, are the fastest-growing applications in business (Laudon and Laudon, 1998, p. 14). An intelligent system will become an important component in the next wave of management support systems, helping the organization to capture, codify, share, create and distribute knowledge.

Although expert systems are the most common technology that uses a knowledge base, several other methods offer support. Technologies which use knowledge, organized in a knowledge base, are considered applications of artificial intelligence. The basic techniques used by artificial intelligence software are searching and pattern matching. Given some initial start-up information, the algorithm searches for specific conditions or patterns until it finds the best match in the knowledge base. A genetic algorithm is a relatively new example of this type of algorithm. It uses an intelligent search method that creates potential solutions by simulating (biological) evolution.

The structure of the intelligent subsystem that contains the core tools, including a genetic algorithm, is given in Figure 3. The subsystem, which builds on the structure of an intelligent decision support system (Sprague and Watson, 1996, p. 13; Turban and Aronson, 1998, p. 784), consists of the following components: a data management subsystem that includes the database and database management software; a model management subsystem that includes financial models, statistical models, management science models, etc. For example, the standard statistical models would include the traditional tools of regression analysis, factor analysis, etc.; an intelligent supervisor that provides intelligence enhances the analysis provided by the other subsystems and that of the decision-maker. The intelligent supervisor consists of a central control, which analyzes the incoming data and coordinates the full array of artificial intelligence technologies. It provides an intelligent interface, intelligent model management and intelligent data management between the inference engine and the artificial intelligence technologies; and a user-friendly, interactive interface to the subsystem.

The intelligent supervisor consists of one or more intelligent subsystems, using artificial intelligence technology such as neural nets, genetic algorithms and fuzzy logic. This structure incorporates an expert system as one of the tools for problem-solving, alone or in combination with the other artificial intelligence technologies. An architecture that proposes that the expert system be placed between the data and the models weakens the potential synergism among the artificial intelligence technologies. In this structure the intelligent subsystem can use, for example, neural computing alone or in combination. An architecture that integrates an expert system and a decision support system, and expands this, for example, to include a model management system, would inherit the restriction to a structured problem domain.

These are some of the recent methods applied to complex problems, each handling uncertainty, structure and knowledge differently. The framework above suggests that each basic component of artificial intelligence technology can be used in series or in parallel. For example, a genetic algorithm can be used to identify classes of membership that are used in the fuzzy logic algorithms. Additionally, a genetic algorithm learning method could be used for discovery of rules in large databases. This is then fed into an expert system.

The intelligent supervisor plays the role of a human expert, much like an expert system, but in this framework the expertise applies to a wider

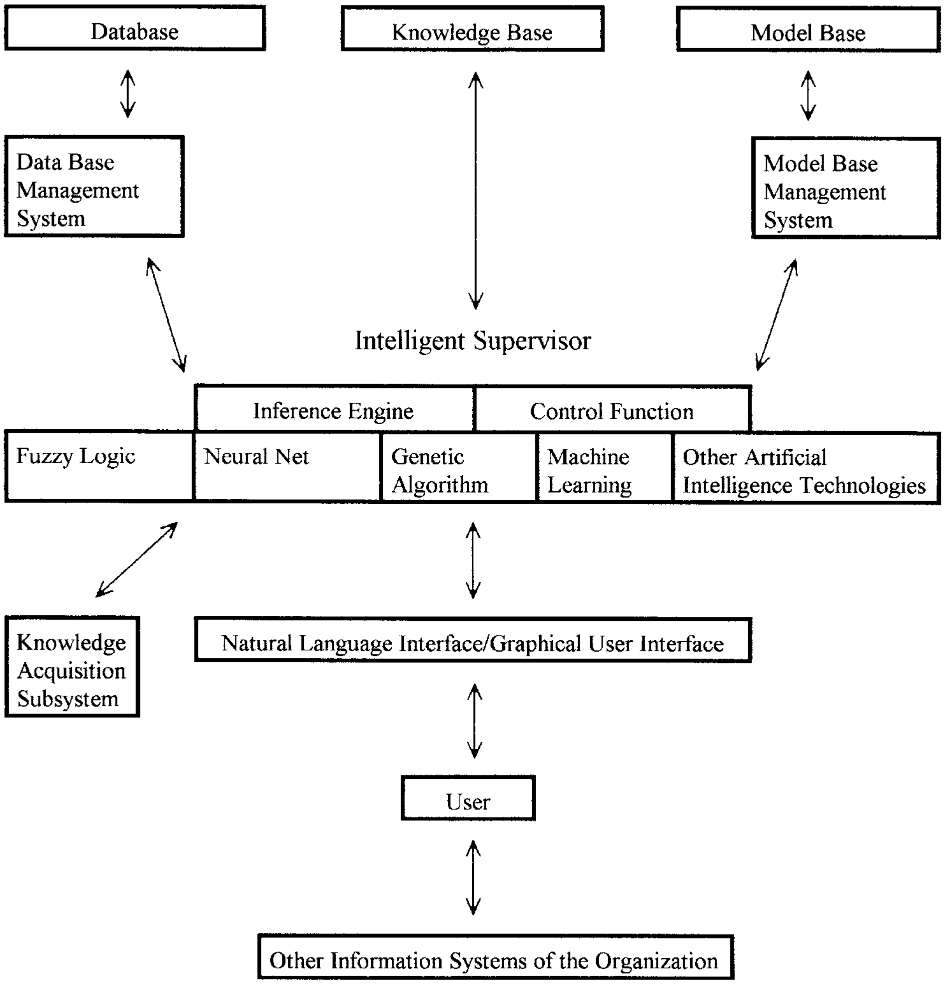

Figure 3. Integrating artificial intelligence with an intelligent subsystem

class of very complex problems, as compared to the usual specialized, narrow problem-solving domains or tasks. This system, which builds on the full array of artificial intelligence tools, expands the potential applications to a wider problem space as well as offering support to the manager confronted with the current knowledge-based view of the firm.

Applications

The application of the evolutionary-based technique of genetic algorithms is gaining momentum in industrial applications. This is mainly because genetic algorithms can solve hard problems quickly and reliably, they are easy to interface to existing simulations and models, are extensible and are easy to hybridize (Goldberg, 1994). For example, genetic algorithms are being used in an intelligent system to assess a company's bankruptcy risk (Goonatilake, 1996); in a parallel database query toolkit of an intelligent business system (Ubois and Vaughan, 1994); in pattern recognition applications with user interaction systems (Johnson, 1998); in generating a production schedule at John Deere & Company's seed planter factory; in aiding the design of jet engines for the Boeing 777 by General Electric Corporation; in the delivery schedule at a Mexican cement company;

in painting trucks at a General Motors production facility (Rayner, 1996); in optimizing the sequence of advertisements within a commercial break at a British television station (Al-Attar, 1994); and in automatic scheduling system for a steel-making plant (Hamada *et al.*, 1995), to name a few. Engineers at General Electric used a genetic algorithm to design jet turbine aircraft, a problem involving 100 variables and 50 constraints. Other applications include the Coors Brewing Company, which used genetic algorithms for scheduling filling and shipment orders (Burtka, 1993). Another technique, dynamic hill climbing, which combines genetic algorithms with traditional methods, created a new tool used in a medical imaging system that helped brain surgeons to plan and perform operations (Turban and Aronson, 1998, p. 705).

In addition, there are numerous academic application papers which have appeared in the literature. Some of the selective few applications include negotiations (Matwin *et al.*, 1991; Oliver, 1996/1997); facility layout (Delmair *et al.*, 1997); project scheduling (Tseng and Mori, 1997); financial forecasting (Mahfoud and Mani, 1996); bankruptcy predictions (Back *et al.*, 1996); electric power demand forecasting (Lee *et al.*, 1997); production scheduling and vehicle routing (Tajima *et al.*, 1996); queries in a distributed database system (Rho and March, 1997); airline crew scheduling (Levine, 1996); statistical applications (Chatterjee and Laudatto, 1997); goal programming (Gen *et al.*, 1997); trading in foreign exchange markets (Neely *et al.*, 1997). Also, a recent textbook covers the central aspects of genetic algorithms and their applications to optimization problems which are inherent in the industrial engineering and manufacturing systems design (Gen and Cheng, 1997). Furthermore, this reference provides an excellent summary of the various events associated with genetic algorithms, including conferences, workshops, books, special journal issues and public Internet web sites.

There are a number of commercially available software packages based on genetic algorithms. In our computational experience, we used three packages: *Evolver* (1997), *GeneHunter* (1995) and *Generator* (1995). These packages were chosen since they provide a simple, direct interface with a spreadsheet software, which a typical manager has readily available for use, such as Microsoft's *Excel*. A major advantage of this approach is that it does not require a manager to earn a new programming language, such as Visual Basic and/or C^{++}, for using genetic algorithms to solve the business problem in hand. Thus, this reduces the set-up time to a minimum for the manager who is more concerned about the day-to-day issues of the business rather than learning a new programming language. For more details about the software used and the problems tested, see Chaudhry (1998). In all, 12 different problems were solved using the three genetic algorithm software packages. These problems included the traveling salesperson problem; the assignment problem, in which the objective was to maximize total utility of the workforce assignment; the advertising problem, in which the objective was to maximize profit subject to budget constraints; and to locate a minimum of facilities not only to cover all the demand points but also to provide inter-facility coverage, among others. Each problem was solved 10 times and with two different initial conditions, which gave a total of 240 problem instances. For consistency purposes, we initialized the parameters to the same levels, so that a direct comparison would be possible. Two of the three genetic algorithm software packages performed about the same in terms of obtaining a solution and the amount of computational burden in achieving the solution. They were *GeneHunter* and *Generator*, whereas *Evolver* was almost always third in performance with respect to the two criteria.

As seen above, the genetic algorithm software links easily with the Microsoft Excel package. This experience exhibits that the emerging genetic algorithm commercial software of this type can be easily integrated with the existing information system platforms within the organization using the framework presented in this paper.

CONCLUSIONS

In this paper, we build upon the biotic evolution and evolutionary strategies which are mappings of the concepts and principles of evolution found in systems science. We presented a framework which integrated the artificial intelligence technologies, specifically genetic algorithms, in a cooperating environment with the users and organization's information systems.

Many believe that intelligent systems will become valuable components in the next generation of management support systems. In these systems, each central building block technology can be used in series or in parallel; for example, the genetic algorithms could be used for rule discovery in large databases, then these rules fed into an expert system. Also, the emergence of hybrid forms of the new algorithms give promise of even more powerful intelligent systems.

We have demonstrated through previous research and our own computational experience with commercial software that the genetic algorithms can solve a wide class of business problems as compared to other solution procedures which are constrained to a more limited problem domain. The increasing business applications suggest that in order to be competitive in the global business environment organizations must now consider and plan for the appropriate artificial intelligence tools within the existing information systems.

ACKNOWLEDGEMENTS

The first author would like to thank the Presidents of Palisade Corporation of Newfield, NY, and New Light Industries, Inc. of Spokane, WA, for their gracious support in providing the software.

REFERENCES

Ackoff, R. L. (1971). Towards a system of systems concepts. *Management Science* **17**(11), 661–671.

Al-Attar, A. (1994). A hybrid GA-heuristic search strategy. *AI Expert* **9**(9), 34–37.

Arrow, K. J. (1973). *Information and Economic Behavior*, Federation of Swedish Industries, Stockholm.

Back, B., Laitnen, T., and Sere, K. (1996). Neural networks and genetic algorithms for bankruptcy predictions. *Expert Systems with Applications* **11**(4), 407–413.

Badaracco, J. (1991). *The Knowledge Link: How Firms Compete Through Strategic Alliances*, Harvard Business School Press, Boston, MA.

Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998). *Genetic Programming — An Introduction: Automatic Evolution of Computer Programs and its Applications*, Morgan Kaufmann, San Francisco, CA.

Boulding, K. (1966). The economics of knowledge and the knowledge of economics. *American Economic Review* **56**(2), 1–13.

Burtka, M. (1993). Genetic algorithms. *Stern Information Systems Review* **1**(1), 7–27.

Chatterjee, S., and Laudatto, M. (1997). Genetic algorithms in statistics: procedures and application. *Communicational Statistics* **26**(4), 1617–1630.

Chaudhry, S. S. (1998). An evaluation of genetic algorithm software packages. Presented at the CORS/INFORMS meeting, Montreal.

Chen, S. (1995). Role of the information infrastructure and intelligent agents in manufacturing enterprises. *Journal of Organizational Computing* **5**(1), 53–67.

Dhar, V., and Stein, R. (1997). *Intelligent Decision Support Methods*, Prentice-Hall, Upper Saddle River, NJ.

Delmaire, H., Langevin, A., and Riopel, D. (1997). Skeleton-based facility layout design using genetic algorithms. *Annals of Operations Research* **69**, 85–104.

Evolver: Advanced Genetic Optimization for Spreadsheets (1997). Paalisade Corporation, Newfield, NY.

Feng, S., and Xu, L. (1996). A hybrid knowledge-based system for urban development. *Expert Systems with Applications* **10**(1), 157–163.

Gen, M., and Cheng, R. (1997). *Genetic Algorithms and Engineering Design*, John Wiley and Sons, New York.

Gen, M., Ida, K., Lee, J., and Kim, J. (1997). Fuzzy nonlinear goal programming using genetic algorithm. *Computers and Industrial Engineering* **33**(1–2), 39–42.

GeneHunter (1995). Ward Systems Group, Frederick, MD.

Generator (1995). New Light Industries, Spokane, WA.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

Goldberg, D. E. (1994). Genetic and evolutionary algorithms come of age. *Communications of ACM* **37**(3), 113–119.

Goonatilake, S. (1996). Risk assessment using intelligent systems. *Insurance Systems Bulletin* **11**(10), 2–3.

Hamada, K., Baba, T., Sato, K., and Yufu, M. (1995). Hybridizing a genetic algorithm with rule-based reasoning for production planning. *IEEE Expert* **10**(5), 60–67.

Hayes-Roth, F., and Jacobstein, N. (1994). The state of knowledge-based systems. *Communications of ACM* **37**(3), 27–39.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Holsapple, C., and Whinston, A. B. (1996). *Decision Support Systems*, West Publishing, Minneapolis, MN.
Jackson, M. C. (1995). Beyond the fads: systems thinking for managers. *Systems Research* **12**(1), 25–42.
Johnson, R. C. (1998). Surfing the web gets new smarts. *Electronic Engineering Times* **987**, 37–38.
Klir, G. J. (1991). *Facets of Systems Science*, Plenum Press, New York.
Laudon, K. C., and Laudon, J. P. (1998). *Management Information Systems: New Approaches to Organization and Technology*, Prentice-Hall, Upper Saddle River, NJ.
Lee, D. G., Lee, B. W., and Chang, S. H. (1997). Genetic programming model for long-term forecasting of electric power demand. *Electric Power Systems Research* **40**, 17–22.
Levine, D. (1996). Application of a hybrid genetic algorithm to airline crew scheduling. *Computers and Operations Research* **23**(6), 547–558.
Machlup, F. (1962). *The Production and Distribution of Knowledge in the United States*, Princeton University Press, Princeton, NJ.
Mahfoud, S., and Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence* **10**, 543–565.
Matwin, S., Szapiro, T., and Haigh, K. (1991). Genetic algorithms approach to a negotiation support system. *IEEE Transactions on Systems, Man and Cybernetics* **21**(1), 102–114.
Neely, C., Weller, P., and Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of Financial and Quantitative Analysis* **32**(4), 405–426.
Oliver, J. R. (Winter 1996/1997). A machine-learning approach to automated negotiation and prospects for electronic commerce. *Journal of Management Information Systems* **13**(3), 83–112.
Pattee, H. H. (1986). Universal principles of measurement and language functions in evolving systems. In Klir, G. J. (ed.), *Facets of Systems Science*, Plenum Press, New York.
Quinn, J. B. (1992). *Intelligent Enterprise: A Knowledge and Service Based Paradigm for Industry*, Free Press, New York.
Rayner, B. (1996). Life's getting complex. *Electronic Business Today* **22**(12), 49–51.
Rho, S., and March, S. T. (1997). Optimizing distributed join queries: a genetic algorithm approach. *Annals of Operations Research* **71**, 199–228.
Sprague, R. H., and Watson, H. J. (1996). *Decision Support for Management*, Prentice-Hall, Upper Saddle River, NJ.
Tajima, K., Adachi, N., Sasagawa, F., Sato, M., and Yoshida, Y. (1996). Genetic algorithms and their practical applications. *Fujitsu Science Technology Journal* **32**(2), 271–286.
Tseng, C. C., and Mori, M. (1997). A genetic algorithm for multi-mode resource constrained project scheduling problem. *European Journal of Operational Research* **100**, 134–141.
Turban, E., and Aronson, J. E. (1998). *Decision Support Systems and Intelligent Systems*, Prentice-Hall, Upper Saddle River, NJ.
Ubois, J., and Vaughan, J. (1994). Parallelizing DBs come to town. *Software Magazine* **14**(6), 24–25.
Warfield, J. N., and Christakis, A. N. (1987). Dimensionality. *Systems Research* **4**, 127–137.
Wright, D. (1996). The accountant in 2005. *Accountancy* **117**(1232), 73.
Xu, L. (1988). A fuzzy multiobjective programming algorithm in decision support systems. *Annals of Operations Research* **12**, 315–320.
Xu, L. (1991). Systems characteristics in information systems design. *Systems Research* **9**, 67–78.