Jan. 2006, Vol.21, No.1, pp.1–18 J. Comput. Sci. & Technol.

Recent Advances in Evolutionary Computation

Xin Yao[1,2] (姚 新) and Yong Xu[2] (徐 永)

[1] *Nature Inspired Computation and Applications Laboratory, The University of Science and Technology of China Hefei 230027, P.R. China*

[2] *The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK*

E-mail: x.yao@cs.bham.ac.uk; y.xu@cs.bham.ac.uk

Received August 18, 2005; revised November 4, 2005.

Abstract Evolutionary computation has experienced a tremendous growth in the last decade in both theoretical analyses and industrial applications. Its scope has evolved beyond its original meaning of "biological evolution" toward a wide variety of nature inspired computational algorithms and techniques, including evolutionary, neural, ecological, social and economical computation, etc., in a unified framework. Many research topics in evolutionary computation nowadays are not necessarily "evolutionary". This paper provides an overview of some recent advances in evolutionary computation that have been made in CERCIA at the University of Birmingham, UK. It covers a wide range of topics in optimization, learning and design using evolutionary approaches and techniques, and theoretical results in the computational time complexity of evolutionary algorithms. Some issues related to future development of evolutionary computation are also discussed.

Keywords evolutionary computation, neural network ensemble, prisoner's dilemma, real-world application, computational time complexity

1 Introduction

Evolutionary computation (EC) is the study of computational systems which use ideas and get inspirations from nature evolution and adaptation. It is a fast growing interdisciplinary research field in which a variety of techniques and methods are studied for dealing with large, complex, and dynamic problems. The primary aims of EC are to understand the mechanism of such computational systems and to design highly robust, flexible, and efficient algorithms for solving real-world problems that are generally very difficult for conventional computing methods.

EC was originally divided into four branches[1−6]: evolution strategy (ES), evolutionary programming (EP), genetic algorithm (GA), and genetic programming (GP). However, such classification does not capture the essence of EC. Here we divide it into the following four areas: evolutionary optimization, evolutionary learning, evolutionary design and theoretical foundation, which we will discuss in this paper. Nowadays, all the approaches used in EC employ a population-based search engine with perturbation (e.g., crossover and mutation) and acceptation (selection and reproduction) to find better solutions. Compared to conventional optimization and artificial intelligence methods, the major advantages of EC approaches include[7]: conceptual and computational simplicity, broad applicability, excellent real-world problem solvers, potential to use domain knowledge and hybridize with other methods, parallelism, robust to dynamic environments, capability for self-optimization, able to solve problems with no known solutions, etc. There are also some other advantages with EC approaches, e.g., no need for analytic expression of the problem, no need for derivative, etc.

The advantages of EC approaches make them extremely suitable for the problems with dynamically changing environment and multiobjective optimization requirement. If a problem's environment is constantly changing and makes the current best solution unacceptable, there might be another solution in the population which fits the current environment better. In other words, using an evolutionary algorithm (EA) in a dynamic environment can avoid "having all eggs in one basket". In multiobjective optimization problems, we need to find a set of Pareto-optimal solutions which make compromises among different objectives from which a human expert can make a choice. In this situation, EAs have the advantage of yielding a whole set of potential solutions, which are all optimal in a sense. Therefore, with the successful applications of EAs in more and more areas, particularly to problems which are intractable with traditional methods, EC approaches will attract increasing interest from both academia and industrial society.

The research of EC in the School of Computer Science at The University of Birmingham has a long history. Composed of over 50 people and publishing more than 100 papers in refereed international journals and leading international conferences every year, the Natural Computation Research Group was established within the School and is one of the strongest in the world in evolutionary computation. A research centre, the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) in the School is dedicated to conducting the world-class research in Computational

Survey

This work is partially supported by the National Natural Science Foundation of China (Grant No. 60428202), and the Advantage West Midlands, UK.

Intelligence and helping industry and businesses becoming more globally competitive through innovations. This paper provides an overview of some recent advances in evolutionary computation that have been made in CERCIA. It covers various topics regarding evolutionary learning, design, optimization, and theory. The EC topics covered in this paper include evolutionary programming, neural network ensemble, co-evolution, multiobjective optimization, and game theory. Due to space constraints, many important topics cannot be discussed here. Interested readers are recommended to visit our website on http://www.cercia.ac.uk.

2 Unified Generate-and-Test Framework

Evolutionary algorithms (EAs) can be described under the unified **generate-and-test** framework shown in Fig.1[8]. The advantage of introducing EAs as a type of generate-and-test search algorithms is that EAs and other search algorithms, such as simulated annealing (SA), Monte Carlo (MC) method, tabu search (TS), and hill-climbing, etc., can be classified under the same framework and the relationships among them are clearer and thus easier to be explored and understood. We can see that under this framework, different search algorithms investigated in artificial intelligence, operations research, computer science, and evolutionary computation can be expressed in exactly the same way and this facilitates cross-fertilization among these algorithms.

1. Generate the initial solution at random and denote it as the current solution;
2. **Generate** the next solution from the current one by *perturbation*;
3. **Test** whether the newly generated solution is *acceptable*;
   (a) Accepted it as the current solution if yes;
   (b) Keep the current solution unchanged otherwise.
4. Goto Step 2 if the current solution is not satisfactory, stop otherwise.

Fig.1. Framework of generate-and-test methods[8].

It is obvious that all the four branches of the evolutionary computation can be comprised in this framework. Hill-climbing algorithms can be described in this framework with different deterministic or stochastic perturbation strategies. They all require that the accepted new solution be no worse than the current one. SA and MC methods usually use a stochastic perturbation and do not have such a requirement. They regard a worse solution to be acceptable within a certain probability. The difference among different SAs, e.g., classical SA[9], fast SA[10], very fast SA[11], and a new SA[12], lies mainly in the different ways of perturbations, i.e., methods of generating the next solution. In TS, however, both deterministic or stochastic perturbation strategies are adopted for generating new solutions but a tabu list needs to be introduced in the acceptance of new solution to prevent the algorithm from deterministic cycle among solutions already visited[13]. All the algorithms and techniques described in this paper can be classified under this framework as well.

3 Improving Evolutionary Programming

EP has been applied with success to many function and combinatorial optimization problems[14−16]. Optimization by EP can be summarized into two major steps:

1) mutate the solutions in the current population, and
2) select the individuals for the next generation from the mutated and the current solutions.

These two steps are a population-based version of the above generate-and-test method, where mutation **generates** new solutions (offspring) and selection **tests** which newly generated solutions should survive to the next generation. Our improvement of EP has been done in the following two aspects.

3.1 Fast Evolutionary Programming[17,18,20]

One disadvantage with conventional EP is its slow convergence to an optimal or near-optimal solution. We found one of the reasons is that conventional EP usually uses Gaussian mutation, with which the search step is sometimes not large enough for the individual to jump out of local optimum. Nevertheless, a large search step size may not be beneficial at all if the current search point is already very close to the global optimum. To address this problem, we have made a theoretical analysis for the first time on the relationship between the distance to the global optimum and the search step size, and the relationship between the search step size and the probability of finding a near (global) optimum. Based on such analyses, a fast evolutionary programming (FEP) with Cauchy mutation was proposed and tested with a wide range of benchmark problems[17,18].

To show when and why a large search step size is beneficial, take Gaussian mutation as an example. Gaussian density function f^G with expectation 0 (implying the current search point locating at 0), and variance σ^2 is

$$f^G(x) = \frac{1}{\sigma\sqrt{2\pi}} b^{-\frac{x^2}{2\sigma^2}}, \quad -\infty < x < +\infty. \tag{1}$$

The probability of generating a point in the neighborhood of the global optimum x^* is given by:

$$P^G(|x - x^*| \leqslant \varepsilon) = \int_{x^*-\varepsilon}^{x^*+\varepsilon} f^G(x)\mathrm{d}x, \tag{2}$$

where $\varepsilon > 0$ is the neighborhood size and σ is often regarded as the step size of the Gaussian mutation.

The derivative of this probability $\frac{\partial}{\partial\sigma}P^G(|x-x^*| \leqslant \varepsilon)$ can be used to evaluate the impact of σ on $P^G(|x-x^*| \leqslant \varepsilon)$. To do this, according to the mean value theorem for definite integrals[19], there exists a δ $(0 < \delta < 2\varepsilon)$ such that

$$\int_{x^*-\varepsilon}^{x^*+\varepsilon} f^G(x)\mathrm{d}x = 2\varepsilon f^G(x^* - \varepsilon + \delta). \tag{3}$$

Hence we have,

$$\frac{\partial}{\partial\sigma} P^G(|x - x^*| \leqslant \varepsilon) = \frac{2\varepsilon}{\sigma^2\sqrt{2\pi}} e^{-\frac{(x^*-\varepsilon+\delta)^2}{\sigma^2}} \cdot \left(\frac{(x^* - \varepsilon + \delta)^2}{\sigma^2} - 1\right). \tag{4}$$

It is apparent from the above equation that

$$\frac{\partial}{\partial\sigma} P^G(|x - x^*| \leqslant \varepsilon) > 0 \text{ if } \sigma < |x^* - \varepsilon + \delta|, \tag{5}$$

$$\frac{\partial}{\partial\sigma} P^G(|x - x^*| \leqslant \varepsilon) < 0 \text{ if } \sigma > |x^* - \varepsilon + \delta|. \tag{6}$$

For Cauchy distribution with scale parameter $t > 0$, the density function is

$$f^C(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2}, \quad -\infty < x < +\infty. \tag{7}$$

Similar analysis yields:

$$\frac{\partial}{\partial\sigma} P^C(|x - x^*| \leqslant \varepsilon) > 0 \text{ if } t < |x^* - \varepsilon + \delta|, \tag{8}$$

$$\frac{\partial}{\partial\sigma} P^C(|x - x^*| \leqslant \varepsilon) < 0 \text{ if } t > |x^* - \varepsilon + \delta|. \tag{9}$$

We can see from the above analyses that a large step size is beneficial (i.e., increases the probability of finding a near-optimal solution) only when the distance between the neighborhood of x^* and the current search point (at 0) is larger than the step size, or else a large step size may be detrimental to finding a near-optimal solution. The above analyses also show the rates of probability increase/decrease by deriving the explicit expressions for $\frac{\partial}{\partial\sigma} P^G(|x - x^*| \leqslant \varepsilon)$ and $\frac{\partial}{\partial\sigma} P^C(|x - x^*| \leqslant \varepsilon)$.

Based on the above results, we proposed a simple yet effective evolutionary optimization algorithm—FEP with Cauchy mutation and achieved better results for most of the benchmark problems we tested[17]. The reason why FEP performs better than conventional EP (CEP) is that the initial population is generally far away from the global optimum on average and Cauchy mutation is more likely to generate larger jumps than Gaussian mutation. FEP would be less effective than CEP when the search point is near the small neighborhood of the global optimum. Based on this observation, we analyzed the impact of search step size on search efficiency and proposed an improved FEP (IFEP) in which two individuals were generated for each parent, one by Cauchy mutation and the other by Gaussian. The better one is then chosen as the offspring[17]. Simulations showed that IFEP is robust and performs at least as well as the better one of FEP and CEP for most benchmark problems.

In [20], we generalized this idea by proposing a mutation operator based on Lévy probability distribution since Cauchy distribution is only a special case of Lévy distribution. One of the characteristics with Lévy distribution is its power law in the tail region. The power law implies that there is no characteristic length scale and is the milestone of fractal structure. Moreover, by adjusting this distribution's parameters, the shape of the probability density can be changed, which in turn yields continuously adjustable variation in the mutation. This gives us an ideal opportunity for designing adaptive mutations and, therefore, is also more general and flexible than Cauchy mutation. Experimental results showed that the adaptive Lévy EP (LEP) performed well on the benchmark functions tested.

Inspired by the game theory, an alternative approach to designing a mixed mutation strategy in EP was presented in [21]. The central topic of game theory is the interactions and strategies among a group of players[22,23]. So in this strategy, individuals in EP are regarded as players in a game. Each individual chooses a mutation strategy from its strategy set, consisted of either Gaussian, Cauchy, Lévy or other probability distributions, based on a selection probability and then generates an offspring by the selected strategy. They play a two-player game in pairs iteratively until some stopping criterion is satisfied.

It is obvious that this mixture strategy will solve problems more efficiently than conventional EP, which usually uses a single mutation strategy, since none of the single mutation operators can solve all problems efficiently no matter how powerful it is. This paper has confirmed this point. Experimental results on the seven test functions demonstrated that the mixed mutation had achieved the same or nearly same performance as the best of Cauchy and Gaussian mutations over all test functions, and even better in some cases. If only a single mutation strategy is applied, neither Gaussian nor Cauchy mutation can solve all these test functions efficiently.

3.2 Parallel Evolutionary Programming[26,27]

When EPs are applied to very hard and big problems, they usually need a long time to find adequate solutions. For these situations, apart from making efforts to design a faster EP, dividing a large task into a few smaller and easier ones and running a number of EPs simultaneously in multiple processors to solve each of them is also a good idea. This divide-and-conquer approach has been applied to GAs and EPs in many different ways with successful parallel implementations[24−27]. For GAs, there are mainly three types of parallel implementation scenarios: 1) global single-population master-slave GAs, 2) single-population fine-grained GAs, and 3) multiple-population coarse-grained GAs[25]. We have proposed a number of parallel EP implementation scenarios for training artificial neural networks (ANNs)[26,27].

ANNs have provided an important classification tool for knowledge discovery in databases (KDD). Feedforward neural networks using a back-propagation training algorithm for weight adjustment are common but usually very time-consuming, particularly when large datasets are involved. Furthermore, determining the best ANN structure for a particular task is also a difficult art, with no hard and fast rules. Therefore, we proposed an EP approach, the EPNet, to train ANNs, which dynamically modified not only weights but also ANN structures to obtain a best ANN model for the tasks at hand[28]. Since EPNet is only a serial algorithm, later, we developed a few parallel versions for it[26,27].

In [26], we presented two parallel structures for the EPNet algorithm: population parallelism and individual parallelism. Population parallelism takes advantage of the independence of each individual in the ANN population to train different individuals concurrently. Individual parallelism takes advantage of the independence of the nodes within an ANN. The connection weights between different node pairs can be altered simultaneously. Two parallel models, the farmer/worker and farmer/worker/helper architectures, were implemented. In these models, a *farmer* is responsible for both generating a new ANN population in each run and dividing the population amongst the workers. The farmer must also ensure that the global population remains consistent. Each *worker* is assigned a sub-population of ANNs, and performs the EPNet algorithm on that population. *Helpers* add an extra level of individual parallelism by assisting workers performing specific tasks.

In the farmer/worker model, the farmer generates a global population, which is then divided into several subpopulations. Each worker is assigned a subpopulation to perform the EPNet algorithm and ensures that Selection, Mutation, Replacement, and Completion Testing are performed on its sub-population. On completion of a run, the workers transfer their subpopulations to the farmer.

The farmer/worker model only utilizes population parallelism but not individual parallelism. So we proposed a farmer/worker/helper model in the same paper to take advantage of both population and individual parallelisms. In this model, a farmer and worker perform similar tasks but workers are assigned a group of helpers. Helpers conduct initial training of the entire population and any modified random search training performed on any individual. The latter is performed at individual parallelism level.

Simulation results indicated that our parallel models are feasible and efficient in training ANNs. The helpers are particularly useful in supporting modified random search, the most time-consuming component of EPNet. Therefore, this architecture is a better option in terms of accuracy and time performance.

In [27], a different model was proposed in which all nodes start with different initial subpopulations. Thus, they will search different regions of the search space. Moreover, this model only transfers a small portion of individuals among processing nodes, not the entire population. Simulation results showed that the parallel version is not only faster than the serial version, but also more reliably in finding optimal solutions. We also showed that if the workload increases, the parallel performance in terms of the speedup factor will also improve. For the evolution of neural networks, we found that parallel EP outperforms the serial version in terms of both the reliability in finding the solutions and the execution time. Furthermore, synchronous migration also improves the rate of finding the optimal solutions.

4 Evolving Neural Network Ensembles

Many real-world problems are too large and too complex for a single monolithic system to solve alone. It has been shown from both natural and artificial examples that an integrated system consisting of several subsystems can reduce its total complexity while solving a difficult problem satisfactorily. The success of neural network ensembles (NNEs) in improving a system's generalization ability is a typical one[29].

NNEs adopt the divide-and-conquer strategy. Instead of using a single network, an NNE combines a set of NNs that learns to subdivide a task and thereby tackle it more efficiently and elegantly. An NNE offers several advantages over a monolithic NN. First, it can deal with more complex tasks than any of its components (i.e., individual NNs in the ensemble). Second, it can make an overall system easier to be understood and modified. Finally, it is more robust than a monolithic NN and can show graceful performance degradation in situations where only a subset of NNs in the ensemble is performing correctly.

Given the advantages of NNEs and the complexity of the problems to be investigated, it is clear that NNE method is and will be an important and pervasive problem-solving technique. However, designing an NNE is generally a very difficult task. In order to improve its design, we have made a number of significant progresses in recently years.

4.1 Negative Correlation Ensemble Learning[36−39]

The idea of ensemble learning can be traced back to as early as 1958[30]. Since early 1990's, algorithms based on similar ideas have been developed in many different but related forms, such as NNEs[31,32], mixtures of experts[33,34], and various boosting and bagging methods[35]. However, all of these algorithms rely on some manual design, such as predefining the number of individual NNs and/or dividing the training data according to human experience and prior knowledge. While manual design and a fixed ensemble architecture may be appropriate when experienced human experts have sufficient prior knowledge of the problem to be solved, it is certainly not the best method for very complex real-world problems where we cannot gain such prior knowledge. In these cases, tedious trial-and-error processes were often involved.

In order to automatically design NNEs without relying on human experts, we developed an evolutionary ensemble with negative correlation learning (EENCL)[36] based on negative correlation learning[37–39] and evolutionary learning[28,29,40]. The main advantage of negative correlation learning is that it encourages different individual NNs to learn different aspects of the training data so that the ensemble can learn the whole dataset better. It does not require any manual division of the training data to produce different training sets for different individual NNs in an ensemble. Specifically, EENCL differs from previous approaches in three major aspects.

Firstly, EENCL emphasizes specialization and cooperation among individual NNs in the ensemble where negative correlation learning[37–39] is used to encourage the formation of different species. But most previous approaches did not acknowledge or exploit the correlation information among individual NNs as a driving force to control the problem-solving process. In those methods, individual NNs were often trained independently or sequentially.

Secondly, EENCL uses unique error functions to learn and combine these NNs as a synthesized system, which provides an opportunity for different NNs to interact with each other and to specialize. The strength parameter in EENCL provides a convenient way to balance the bias-variance-covariance trade-offs. On the contrary, most previous approaches separated the design process into two stages: generating individual NNs and then combining them. They did not exploit the possible interactions among individual NNs until the combination stage. There was no feedback from the combination stage to the NN design stage. In this way, some of the independently designed individual NNs would contribute very little to the whole system.

Thirdly, in EENCL, the number of NNs in the ensemble is not predefined but determined by the number of species, which are formed automatically through evolution. Nevertheless, in most previous approaches, the number of NNs in the ensemble was often predefined and fixed, usually manually through a trial-and-error process. Since the number of NNs in the ensemble was not optimized in those approaches, it may result in too many or too few NNs in the ensemble.

There are two levels of adaptation in EENCL: negative correlation at individual level and evolutionary learning based on EP at population level. EP is used to search for a population with diverse individual NNs that work together to solve a problem. Fitness sharing[41] and negative correlation learning are used to encourage the formation of diversified species. A negative correlation penalty term among individual NNs is introduced to the error function of each NN to ensure that all the networks are trained simultaneously and interactively on the same data set. Each NN in the ensemble can be trained by a specific learning algorithm.

The effectiveness of EENCL was verified by comparing it with other 23 algorithms used in Michie *et al.*[42]. Experiments were conducted on both the Australian credit card assessment problem and the diabetes problem obtained from the UCI machine learning benchmark repository[43]. Simulation results showed that EENCL is able to achieve the generalization performance comparable to or better than the best of 23 algorithms tested in [29] for both problems.

4.2 Constructive Learning of NNEs[44,45]

Although EENCL is very effective in generating diversified individual NNs to form an NNE, the structure of individual NNs in the ensemble, e.g., the number of hidden nodes, is pre-designated manually and fixed during the training process, which sometimes does not suit complex learning task. In order to overcome this drawback, we proposed a constructive NN ensemble (CNNE) learning algorithm for training NNEs cooperatively[44]. CNNE combines ensemble architecture design with cooperative training of individual NNs in an ensemble. It automatically determines not only the number of NNs in an ensemble, but also the number of hidden nodes in individual NNs via an incremental training algorithm based on negative correlation learning[37–39].

CNNE starts with a minimal ensemble architecture consisting of two individual NNs with only one hidden node in each NN. All individual NNs in the ensemble are first partially trained on the training dataset using negative correlation learning. If the ensemble error E on the validation dataset is acceptable (e.g., smaller than a pre-designated number), stop the training process. Otherwise, check if the criteria for halting network construction and node addition for each NN are satisfied or not. If they are satisfied, it is assumed that the labeled NNs are not sufficiently trained, so further train those NNs. Otherwise, add one new NN to the ensemble or one new hidden node to each NN that satisfies the criterion for node addition and continue training. The idea behind CNNE is that it tries to minimize the ensemble error first by training, then by adding an appropriate hidden node to the existing NNs, and lastly by adding a new NN to the ensemble. With this method, we can obtain an optimal ensemble architecture with each individual NN in the ensemble having the minimal number of hidden nodes.

In comparison with other ensemble training algorithms, the major advantages of CNNE include: 1) automatic design of ensemble architectures; 2) maintaining of both diversity and accuracy among individual NNs in an ensemble; 3) a good generalization ability of constructively learned ensembles; and 4) minimizing the training time since we do not over-train any NN in the ensemble, nor do we add any superfluous NN to the ensemble.

CNNE has been tested extensively on many benchmark problems, including Australian credit card assessment, breast cancer, diabetes, glass, heart disease, letter recognition, soybean, and Mackey-Glass chaotic time series prediction problems. In terms of generalization ability, experimental results showed clearly that CNNE is better than other ensemble and nonensemble learning algorithms.

We have also proposed an improved CNNE (ICNNE) which improves the original CNNE in the following aspects[45]. First, we used a single NN but not two minimal individual NNs to construct an initial ensemble since the best structure is often problem-dependent and sometimes, a single NN can solve the problem very well so we do not need two NNs to form an ensemble. Second, we modified the stopping criteria to ensure that each NN in the ensemble can be trained sufficiently. Third, we adopted the cross-entropy error function since it is better for our medical diagnosis problems. Simulations on three medical diagnosis problems showed that ICNNE algorithm achieves similar or better results than CNNE.

4.3 Ensemble Learning via Multiobjective Approaches[48−50,53]

In order for an NNE to generalize properly, two factors are considered vital. One is the diversity and the other is the accuracy of the networks that comprise the ensemble. According to Brown *et al.*[46], if two neural networks make different errors on the same data points, they are said to be diverse while accuracy could be defined as the degree of a network (ensemble member) performing better than random guessing on a new input. Since a population is bound to have at least as much information as any single individual[47], we need ensembles for better generalization. But given the fact that similar members would preclude the need for ensembles, members have to be diverse. The more diverse the members are, the more well spread will their outputs be around the target value, which results in the expected (mean) value of the member outputs being closer to the target value. On the other hand, in order for an ensemble to generalize well, the member networks, apart from being diverse, should also be accurate. But these two objectives often conflict with each other. Hence, a multiobjective optimization based approach is suitable for making trade-offs between these two objectives.

Based on the above observation, we proposed the DIVerse and ACcurate Ensemble (DIVACE) learning algorithm to construct an ensemble that searches for the optimum point on the diversity-accuracy curve[48−50]. DIVACE takes in ideas from the memetic Pareto artificial neural network (MPANN)[51] and the negative correlation learning (NCL) algorithms[37−39]. MPANN is used for the evolutionary process and the negative correlation penalty function of NCL is used as one of the objectives for the multiobjective problem for keeping diversity. The two objectives on which to optimize the performance of the ensemble are accuracy and diversity.

Objective 1—Accuracy. Given a training set T with N patterns. For each network k in the ensemble,

$$\text{Minimize: } Accuracy_k = \frac{1}{N}\sum_{i=1}^{N}(f_k^i - o^i)^2, \tag{10}$$

where o^i is the desired output and f_k^i the posterior probability of the class (classification task) or the observed output (regression task) for training sample i.

Objective 2—Diversity. From NCL, the negative correlation penalty function was used as the second objective to optimize the ensemble performance.

Let N be the number of training patterns and let there be M members in the ensemble, for each member k, the following term gives an indication of how different it is from other members.

$$\text{Minimize:} Diversity_k = \sum_{i=1}^{N}(f_k^i - f^i)\left[\sum_{j\neq k, j=1}^{M}(f_j^i - f^i)\right], \tag{11}$$

where f^i is the ensemble output for training sample i. From the information theoretic point of view, mutual information is a measure of the correlation between two random variables. A link between the diversity term, (11), and mutual information was shown in [52]. Minimization of mutual information between variables extracted by two neural networks can be regarded as a condition to ensure that they are different. It has been shown that NCL, due to the use of the penalty function, can minimize mutual information amongst ensemble members[48,52]. Hence it is used as the diversity term in DIVACE. But DIVACE is in no way limited to the use of any particular diversity and accuracy measures. The idea is to address the diversity-accuracy trade-off in a multiobjective evolutionary setup. In fact, we also proposed another diversity measure—pairwise failure crediting (PFC) in [50].

Experiments on both the Australian credit card assessment problem and the diabetes problem obtained from the UCI machine learning benchmark repository[43] were compared with the MPANN[51] and EENCL[36] based on evolutionary ensemble and 21 other algorithms based on statistics, decision trees, rule methods, and neural networks described in [42]. Simulation results showed that both variants of DIVACE, using NCL and PFC as penalty function, are able to achieve competitive and mostly better generalization performance than the best of the other algorithms tested for both problems.

Latter, we tried to incorporate as much information about the diversity enforcement in ensembles as possible into this algorithm so as to develop a framework which may be used as an ensemble learning algorithm generating engine. As a result, an evolutionary framework that uses a myriad of diversity enforcement ideas rolled into one multi-level ensemble learning strategy where individual predictors are generated automatically by successively competing and co-operating with each other. In order to prove the effectiveness/validity of this framework, a new algorithm, called DIVACE-II, was proposed to model this framework. Details about this algorithm can be found in [53]. Detailed experiments with DIVACE-II showed that the framework is indeed valid. It is able to outperform most of the algorithms it is compared with. This indicates that the idea of enforcing diversity at multiple levels (which is modeled by this framework) is a good one. The framework can be used to generate diverse hybrid ensembles that generalize well.

5 Evolutionary Multiobjective Optimizations

Multiobjective optimization (MOO) is no doubt a very important topic for both scientists and engineers because of the multiobjective nature of most real-world problems. The purpose of MOO is to find a set of the so-called Pareto optimal solutions from which the (human) decision maker can choose ideal ones. EAs seem particularly desirable to MOO problems because they deal simultaneously with a set of possible solutions (the population) which allows to find an entire set of Pareto optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, EAs are less susceptible to the shape or continuity of the Pareto front, whereas these two issues are a real concern for mathematical programming techniques[54]. In this section we will review some of the work we have done in improving MOO algorithms.

5.1 Scalability Issues in Multiobjective Optimizations[55]

Most of the previous evolutionary MOO (EMOO) algorithms are mainly applied to two to three objectives. In order to establish their superiority over classical methods and demonstrate their abilities for convergence and maintenance of diversity, these algorithms need to be tested on higher number of objectives. Therefore, in [55], we extended some of the recent-proposed algorithms, i.e., NSGA-II (Non-dominated Sorting Genetic Algorithm-II[56]), SPEA2 (Strength Pareto Evolutionary Algorithm 2[57]), PESA (Pareto Enveloped-based Selection Algorithm[58]), to the test problems with up to 8 objectives. These algorithms were compared on the basis of 1) their ability to converge to Pareto front, 2) diversity of obtained non-dominated solutions, and 3) their running time. Four scalable test problems (DTLZ1, 2, 3 and 6[59]) are used for this comparative study.

Experimental results clearly showed that conclusions drawn from 2 or 3 objectives cannot be generalized to a higher number of objectives. More work need to be done to fully understand the behavior of EMOO problems on different numbers of objectives. For the three performance metrics used, we found that if an algorithm is good in terms of converging to the Pareto front, it would lack good diversity maintenance. We also found that different algorithms have different scalabilities in terms of the performance metrics chosen. The running time increases with the increase of the number of objectives but the increasing speed is different for different algorithms. Sometimes, one algorithm is running faster than another for small number of objectives but it would be slower than the later if the number of objectives increased. Therefore, in terms of running times, different algorithms would suit different numbers of objectives.

5.2 ε-Elimination Diversity Algorithm for Multiobjective Optimizations[60,61]

In EMOO algorithms, keeping the diversity of a population is very important in finding the Pareto front. So in [60] we proposed a new diversity-preserving algorithm, the ε-elimination diversity algorithm to enhance the performance of NSGA-II[56] in terms of diversity of population and Pareto fronts. In this algorithm, we used the ε-elimination diversity approach to replace the crowding distance assignment approach in NSGAII. All the clones and/or ε-similar individuals based on Euclidean norm of two vectors are recognized and simply eliminated from the current population. Based on a predefined value of ε as the elimination threshold ($\varepsilon = 0.001$ was used), all the individuals in a front within this limit of a particular individual are eliminated. Such ε-similarity exists in both the objectives space and the space of the associated design variables, which ensures that very different individuals in the space of design variables having ε-similarity in the space of objectives will not be eliminated from the population. The clones or ε-similar individuals are replaced from the population with the same number of new randomly generated individuals. Meanwhile, this will additionally help to explore the search space of the given multiobjective problem more efficiently.

The proposed ε-elimination diversity algorithm was used to the optimization of turbojet engines. Four conflicting thermodynamic optimization objectives were considered in the process: specific thrust (ST), thrust-specific fuel consumption (TSFC), propulsive efficiency (η_p), and thermal efficiency (η_t). Different pairs of these objective functions, i.e., η_t and η_p, TSFC and ST, were selected for two-objective optimization processes and these objectives are also considered for a four-objective optimization problem. An optimal set of design variables in turbojet engines, namely, the input flight Mach number M_o, the pressure ratio of the compressor π_c, and the turbine inlet temperature T_{4t} were obtained. We found that the results of four-objective optimization can include those of two-objective optimization and, therefore, provide more choices for optimal design of thermodynamic cycle of ideal turbojet engines. Some interesting and important relationships among optimal objective functions and decision variables involved in the thermodynamic cycle of turbojet engines were also discovered consequently. Comparison to NSGA-II demonstrated the superiority of the new algorithm in preserving the diversity of non-dominated individuals and the quality of Pareto fronts in both two-objective and 4-objective optimization problems.

The best Pareto front obtained from the 4-objective optimization problems is a data table representing data pairs of non-dominated vectors of design variables, which are Mach number and pressure ratio, and the corresponding objective functions. So in another work EAs and singular value decomposition were employed simultaneously for optimal design of both connectivity configuration and the values of coefficients, respectively, involved in group method of data handling (GMDH)-type neural networks which were used for the inverse modelling of the input-output data table obtained as the best Pareto front[61]. Two different polynomial relations among the four thermo-mechanical objectives and both Mach number and pressure ratio were searched using that Pareto front. The results were very promising and showed that such important relationships may exist and could be discovered using both multiobjective EAs and evolutionarily designed GMDH-type neural networks.

6 Evolutionary Iterated Prisoner's Dilemma

In a world torn by conflict, there is a keen interest in understanding how mutually beneficial cooperation can emerge spontaneously, with no higher authority than the rule of the jungle. Iterated Prisoner's Dilemma (IPD) is the classic example of a "non-zero sum" game to investigate such phenomenon in economics, political science, evolutionary biology, game theory, and artificial intelligence.

In its basic form, the IPD is a two-player game where each player has two choices, cooperate (C) or defect (D) with the following payoff matrix[62]:

	C	D
C	$R\backslash R$	$S\backslash T$
D	$T\backslash S$	$P\backslash P$

where $T > R$ and $P > S$ (Defection always pays more), $R > P$ (Mutual cooperation beats mutual defection), and $R > (S+T)/2$ (Alternating does not pay).

Robert Axelrod first imposed a co-evolutionary dynamic based on a population of trial strategies playing against each other to gain insight into why mutual cooperation does or does not emerge in IPD[63]. In this approach, a computer maintains a population of trial strategies. The strategy's fitness is evaluated by its peers in the same population and an evaluation function judges the quality of each trial strategy in the population. The fitness is its average payoff from playing all the other strategies in the same evolving population. As the population improves, its members become more challenging and the evaluation function more discerning. The aim is to set up an escalating arms race of innovation. The results showed that mutual cooperation can come to dominate, even without any central authority to enforce the cooperation.

In this Section we will present some of our work in 2-player and N-player IPD.

6.1 2-Player Iterated Prisoner's Dilemma[64,67]

We considered the intermediate choices between full cooperation and full defection and examined why intermediate choices make full cooperation less likely in [64]. This question is relevant to the National Missile Defense, which (its opponents claim) would allow intermediate choices between full peace and all-out war, and thus make a partial nuclear war more likely.

Earlier research found that intermediate choices tended to prevent full cooperation[65,66]. But they did not explain why. We considered an 8×8 payoff matrix in which each player has eight evenly discretized choices of cooperative level as intermediate payoffs. The IPD strategy was represented by a three-layer feed-forward neural network with 10 hidden nodes.

Experiments on the two-choice game showed that if the initial population consisted of only defect strategy, it took some time for mutation to produce some suitably cooperative invaders although it would eventually lead the whole population toward cooperation. Therefore, once behavioral diversity disappears, there is little variation for natural selection to increase cooperation. Increasing cooperation is rare, if the whole population is cooperating at a lower level. Mutation is a slow substitute for behavioral diversity.

When intermediate choice is used, full mutual cooperation is less likely. The question is, if a population can usually increase its degree of cooperation from the bottom level, why does not it keep increasing? Why not ratchet all the way up early in the run, and stay up?

Our experiments showed that mutation did eventually produce an invader strategy that can cooperate at the next highest level. But behavioral diversity disappeared very quickly so that the 8-choice game took quite a long time to move up to the next level. The reason is that although payoff rises to a certain level by making use of the initial diversity, once that diversity dyes out it would be stuck at that level for a long period, until mutation eventually produces a suitable invader.

We also found that more phenotypic diversity (by raising the mutation rate) makes mutual cooperation more likely in the 8-choice IPD. But for a pseudo-continuous IPD, where each player has 1024 choices, even with higher mutation rate, mutual defection is the most likely outcome because a small population simply cannot sample so many possible outcomes.

In short, introducing intermediate choices has the effect of adding obstacles to any change, either for better or worse. Applying this observation to the National Missile Defense system suggests that introducing a missile defense during happy times may act as an obstacle to any deterioration in the diplomatic climate. However, introducing one during a period of tension may act as one more block to improving relations. Therefore, the addition of intermediate choices is not, by itself, going to make conflict more likely.

However, real world dilemmas rarely involve perfect interactions without mistakes (noise). So in [67] we investigated a more realistic situation by considering the IPD game with both intermediate choices and noise. Rather than using a tournament study of fixed strategies, we employed a population-based co-evolutionary neural network system to play the game. The evolutionary behavior of the co-evolving neural networks in learning to play the extended IPD game was discussed in detail. We found that noise has a negative impact on the evolution of cooperation, but could improve, surprisingly, the evolutionary stability. Our study showed that although the extended IPD game is more complicated, observations made from previous studies that extended IPD with intermediate choices and noise separately still hold.

Experiments showed that when noise was added to an IPD game with 4 choices, the evolution towards cooperation was less likely compared to noiseless case. However, with respect to the evolutionary behavior of co-evolving neural networks' population, different noise levels has different effects. At low noise levels, the present of noise actually caused instability to the evolution, with violent fluctuations between full cooperation and full defection. For higher noise levels, evolutionary stability was observed although the evolution tends towards defection play. For a population evolving at a certain cooperation level, it is resistant to invading strategies playing at lower cooperation levels. The population, however, could be invaded by strategies with higher cooperating play which in turn are vulnerable to defecting strategies. Our observations suggested that although generosity is the better response, it is difficult for individuals to discover and to maintain it at face value.

6.2 *N*-Player Iterated Prisoner's Dilemma[71−74]

The N-player iterated prisoner's dilemma (NIPD) game can be defined by the following three properties[68]: 1) each player faces two choices between cooperation (C) and defection (D); 2) the D option is dominant for each player, i.e., each player is better off choosing D than C no matter how many of the other players choose C; 3) the dominant D strategies intersect in a deficient equilibrium. In particular, the outcome if all players choose their non-dominant C strategies is preferable from every player's point of view to the one in which everyone chooses D, but no one is motivated to deviate unilaterally from D.

The NIPD has much richer behavior than the 2IPD. There are a lot of ways to study various issues in the NIPD. Glance and Huberman have used tools from statistical physics to analyze the NIPD and presented an excellent discussion about the sudden emergence of cooperation in a group where mistakes are allowed[69]. They also studied the relationship between the group size and the evolution of cooperation. Albin has used cellular automata to model each player in the NIPD[70]. Each player directly interacts only with his/her neighbors who are a subset of the n players. We also have investigated various aspects of this problem.

In [71] we showed two theoretical results about the evolutionarily stable strategy (ESS) in the NIPD. Assuming that no individual makes mistakes in the NIPD, we showed that no finite mixture of pure strategies can be evolutionarily stable if the probability of further interactions is sufficiently high.

We mathematically showed that every finite history of interactions among n players occurs with positive probability in any evolutionarily stable mixture of pure strategies in the infinite NIPD where the probability of further interaction is sufficiently high. As a result, no finite mixture of pure strategies in the infinite NIPD can be evolutionarily stable.

We also investigated a more realistic model of the NIPD where mistakes were allowed. An individual who intended to cooperate might defect and vice versa. Assuming that there is a positive probability of both types of mistakes (D for C or C for D) on every turn regardless of the node in the game tree and the probability is independent of players, we showed that there exists an ESS which can resist invasion of any finite mixture of other distinct strategies provided these invading strategies are sufficiently rare. ALLD (always defect) was shown to be such an ESS.

In another work, we investigated the localization issue in the NIPD game and the impact of local interaction on genetically evolved strategies for the NIPD game[72,73]. Localization in the NIPD game restricts the opponents of a player to be selected only from his neighborhood. This restriction will understandably influence how the player interacts with others and what fitness he is going to get. Such restriction also influences evolutionary learning of game-playing strategies because fit strategies will reproduce and propagate in the population. The fittest strategies without localization may not be fit at all with localization. Although strategies learned with localization would be less robust than those without it since players have less exposure to a wide range of different opponents, the NIPD game with localization is more realistic in the real world. It would be interesting to know what difference between the two models is.

Our studies showed that localization and payoff function have a dramatic impact on evolved strategies and their generalization ability. Localization encourages cooperative coalition. The more localized the interaction is, the easier it is to evolve cooperation. But a larger difference between the payoffs for defection and cooperation will make non-cooperative move more attractive and hence make a cooperative coalition smaller. We found that history length (the number of the most recent rules stored in a lookup table, see [74]) plays an important role if neighborhood size is sufficiently large. When history length is short, such cooperation could be rather short and unstable. A longer history length makes cooperation more stable but the cooperation takes longer time to emerge.

We also found that the number of players in the NIPD game is not an important parameter when localization is introduced and the neighborhood size is sufficiently small. Localization has a negative impact on the generalization ability of evolved strategies because they are not exposed to a diverse and hostile environment. They perform well against players of their kind, i.e., cooperators, but poorly against defectors. As a result, strategies lose their competitiveness against a wide range of different opponent strategies.

7 Real-World Applications

One of the most important features of EC is its capability of solving real-world problems. Compared to traditional approaches, there are many unique advantages in using EC approaches to real-world problems as was mentioned in the first section of this paper. In this section we will present some of the work done in this centre in applying EC approaches to real-world problems.

7.1 Constraint Handling[75,77]

An issue that is immediately encountered in applying EC approaches to real-world problems is how to handle constraints to the problem. A common way is to apply a penalty function to bias the search toward a feasible solution. However, it is very difficult to strike the right balance between objective and penalty functions. Therefore, we introduced a novel approach to balancing objective and penalty functions stochastically, i.e., stochastic ranking, for constraint handling[75]. This was based on the observation that the penalty function method, in essence, changes the fitness landscape, which in turn influences the selection outcomes since this is the only place when fitness values are used for most EAs. Hence, by changing the selection scheme, we could achieve the same effect as modifying the fitness function. This is exactly what stochastic ranking does.

This technique does not introduce any *a priori* knowledge about the problem since it does not use any penalty coefficient in a penalty function. In this algorithm we introduced a probability P_f which only uses the objective function for comparisons in ranking in the infeasible regions of the search space. That is, given any pair of two adjacent individuals, the probability of comparing them (in order to determine which one is fitter) according to the objective function is 1 if both individuals are feasible; otherwise, it is P_f. The balance between the objective and penalty functions is achieved through a ranking procedure based on the stochastic bubble-sort algorithm[75]. The procedure provides a convenient way of balancing the dominance in a ranked set. In the bubble-sort-like procedure, λ individuals are ranked by comparing adjacent individuals in at least λ swaps. The procedure is halted when no change in the rank ordering occurs within a complete swap.

Stochastic ranking has been tested on a set of 13 benchmark problems[75]. Experimental results suggested that a value of $0.4 < P_f < 0.5$ would be appropriate for many constrained optimization problems. This indicates that a minor bias toward the dominance of the penalty function encourages the evolution of feasible solutions while still maintaining infeasible regions as potential "bridges" to move among feasible regions in the whole search space. Compared with the dynamic penalty method[76], we found that this new approach performed better in most cases.

In another work we treated the constraint handling as a multiobjective optimization problem. We analyzed and explained in depth why and when the multiobjective approach to constraint handling is expected to work or fail[77]. Two multiobjective approaches were presented in this work: one is solely based on constraint violations in determining the Pareto fronts but the other also includes the objective function. We found that the former is more likely to locate feasible solutions than the latter. However, in general, finding feasible solutions using the multiobjective technique is difficult since most of the time is spent on searching infeasible regions. The use of a nondominated rank removes the need for setting a search bias. But it does not eliminate the need for having a bias in order to locate feasible solutions.

Extensive experimental studies have been carried out. We found that the assumption of using the penalty function to bias the search toward the feasible region is a good idea for the 13 test functions but a bad idea for our artificial test function. These results give us some insights into when the penalty function can be expected to work in practice. Our results revealed that the unbiased multiobjective approach to constraint handling may not be as effective as one may have assumed.

7.2 Optimization of Telecommunication Networks[78,79,85–90]

The rapid advances in telecommunication networks and technologies, as well as their applications, have given rise to a massive number of design and optimization problems, difficult to be solved by traditional methods. EC approaches are appealing alternatives, as they have provided very good results in tackling these problems. We have done a lot of work in the optimal design of various telecommunication networks using EC approaches.

We have investigated the terminal assignment (TA) problem in a telecommunication network with a novel hybrid Hopfield network (HNN) GA approach and a tabu search approach[78,79]. TA is important in increasing the telecommunication networks' capacity and reducing the cost of it. It is an NP-complete combinatorial optimization problem in which terminals having a known requirement of capacity have to be assigned to given concentrators with a given maximum capacity[80−82]. The objective of the TA is to minimize link cost to form a network by connecting a given set of terminals to a given collection of concentrators.

In [78], we focused on TA instances where only the cost function of entire feasible solutions can be calculated and the cost of a single assignment cannot be known in advance. We presented a novel hybrid Hopfield network (HNN) GA in which the problem's constraints were managed by the HNN and the quality of the solution obtained was improved by the GA. The use of the HNN reduces the search space of the GA to only the feasible solutions. The performance of our hybrid algorithm was evaluated in several test TA problems and compared with the GA proposed in [81] with very good results in all test instances considered.

This approach has also been applied to a problem directly related to TA, the task assignment problem[83] and FPGA segmented channel routing problems (FSCRPs)[84], in order to show the effectiveness of this approach to other problems. We showed that our algorithm is able to obtain very good results for these problems, outperforming the others GAs within a reasonable computation time.

In [79] we proposed a two-objective model of the TA problem. The two objectives considered in this research were the cost of each terminal assigned to a concentrator and the balance of terminals distributed among concentrators. We presented a tabu search approach to solving two extremes of this TA problem, i.e., the *cost-first* and the *balance-first* cases, respectively. Simulations showed that the minimal cost and the best balance of this problem cannot be reached simultaneously. When the balance-first approach was applied, the cost was 10 to 75 percent higher than the cost-first approach. If the cost-first approach was used, the balance was about 20 to 50 percent higher than the balance-first approach. But the other objective could always reach its optimum.

Comparing our approach with the existing two approaches which used genetic[80] and greedy[81] algorithms, we found that for small and medium sized problems our approach is able to find good solutions with much less effort; for large sized problems, our approach achieved the best results, although consuming longer time. Hence, the comprehensive performance of our approach is better than the previous two.

We have also done a lot of work in traffic grooming in WDM optical networks[87−90]. We proposed a GA approach to the grooming of traffic in WDM ring networks with static and dynamic traffic requirements in [85, 86]. The technique of traffic splitting in grooming was analyzed in detail and incorporated with GA approach to solving strictly and rearrangeably grooming of traffic in WDM ring networks with dynamic traffic requirements[87,88]. The lower bound on the number of ADMs in WDM rings with nonuniform traffic demands was analyzed in detail in [89]. Finally, a comprehensive, thorough, and up-to-date review of EC approaches to the grooming of both static and dynamic traffic in WDM optical networks was provided in [90].

7.3 Unconventional Circuit Design[91−93]

EC approaches have been used in a large variety of design domains, from aircraft engineering to analogue filters. Many of these approaches use measures to improve the variety of solutions in the population. We have combined clustering and Pareto optimization in a single evolutionary design algorithm[91].

The objective of this was to prevent the system from converging prematurely to a local minimum and to encourage a number of different designs that meet the design criteria. Our approach was demonstrated in the domain of digital filter design. Using a polar coordinate based pole-zero representation, two different low-pass filter design problems were explored. The results have been compared to designs created by a human expert using conventional design process, which demonstrated that the evolutionary process is able to create designs that are competitive to those created by a human expert[91]. We also found that each evolutionary run can produce a number of different designs with similar fitness values, but very different characteristics.

But for many complex design problems, EC approaches are usually very time-consuming so that their use is not practical even with high-speed computers. Divide and conquer based methods sometimes improve the situation, but in most cases the biggest speed improvement can be gained by adding domain knowledge. Combining evolutionary methods with conventional design methods is one way of doing this. In [92], we used a digital filter design problem to show how a conventionally derived design can be further improved by evolutionary calibration. Our experimental results showed that the evolutionary calibration algorithm is able to consistently improve the original designs by a considerable margin.

In [93], we showed how artificial evolution can be used to improve the fault-tolerance of electronic circuits. Evolution is also able to create sets of different circuits that, when combined into an ensemble of circuits, have reduced correlation in their fault pattern, and therefore improved fault tolerance. An important part of the algorithm used to create the circuits was a measure of the correlation between the fault patterns of different circuits. Using this measure in the fitness, the circuits evolved towards different, highly fault-tolerant circuits. The measure also proves very useful for fitness sharing purposes. We have evolved a number of circuits for a simple 2 times 3 multiplier problem, and used these to demonstrate the performance under different simulated fault models.

7.4 Dynamic Salting Route Optimization[95]

On marginal winter nights, highway authorities face a difficult decision as to whether or not to salt the road network. The consequences of making a wrong decision are serious, as an untreated network is a major hazard and unnecessary salting causes extra financial and environmental costs. With limited resources and treatment time constraints and relatively large number of salting routes, it is imperative that salting routes are planned in advance for efficient and effective operation. This has traditionally been a manual task and is heavily reliant on local knowledge and experience. Currently, a "static," often paper based, approach is used to optimize salting routes within the given constraints to enable effective use of resources (i.e., treatment vehicles, personnel and de-icing chemical material). The aim is to maintain safe road conditions, whilst minimizing financial and environmental costs[94].

In [95], a new salting route optimization system was proposed which combines EC with the neXt generation Road Weather Information Systems (XRWIS). XRWIS is a new intuitive route based high resolution forecast system which provides the highway engineer with road surface temperature and condition across the road network over a 24-hour period[96]. ECs are used to optimize a series of salting routes for winter gritting by considering XRWIS temperature data along with treatment vehicle and road network constraints. This synergy realizes daily dynamic routing and it will yield considerable benefits for areas with a marginal ice problem.

The prototype system was constructed and examined on two typical marginal nights. The resultant salting routes showed that ECs have the ability to optimize a series of salting routes for winter gritting and elucidate the effectiveness of the proposed method.

7.5 Modeling of Elliptical Galaxies[97,98]

A reasonably good description of the luminosity profiles of galaxies can serve as a guide towards understanding the process of galaxy formation and evolution. Traditionally, a radial brightness profile model of a galaxy is built by means of fitting parameters for a functional form assumed beforehand. As a result, such a model depends crucially on the assumed functional form. Limitations of traditional approaches include: a) an exact mathematical function form must be given before applying any fitting algorithms; b) the commonly used fitting algorithm like non-linear reduced χ^2 tends to be sensitive to initial values provided and more likely culminates in wrong local minima, resulting in unsatisfactory fits.

In [97] we proposed an EC approach that enables one to build profile models from data directly without assuming a functional form in advance. This approach consists of two major steps that serve two goals. The first applies the GP technique to find a promising functional form, whereas the second takes advantage of the power of GP to fit parameters for functional forms found at the first step. The main novelty of this approach lies in the fact that the whole procedure of modeling profiles is a data-driven process without assuming a functional form beforehand. This bottom up process is particularly useful when one faces a large number of galaxy profiles without any prior knowledge of them. It allows one to find a good functional form first and then to fit parameters for the functions in order to build reasonably good galaxy profile models.

The proposed evolutionary approach has been applied to modeling 18 elliptical galaxies profiles. Two major different mathematical function forms were found at the first step. Through a generalization process, three parameters were introduced into each function. Parameter fittings by EP using two different hit criteria were carried out at its second step. Experimental results demonstrated that a good mathematical form plays a more important part in finding good descriptions of galaxy profiles. On the other hand, a smaller hit criterion is preferable to use in order to guide EP to achieve better models.

In [98], we first gave a review of how GP and EP techniques had been applied to the analysis of elliptical galaxies. Then the effectiveness of a maximum likelihood based fitness function was asserted and applied to the parameter fitting using EP. A maximum likelihood based function was found to show consistent and significant improvement over a hit-based fitness function for modeling the profiles of elliptical galaxies. It was asserted that such a function would potentially improve the quality of the model produced by symbolic regression using GP.

8 Computation Time of Evolutionary Algorithms

In EAs, computation time is used to reveal the number of expected generations needed to reach an optimal solution[99,100], which is an important issue in the theoretical analysis of EAs. However, few results exist on this topic[101−103] and most of the tools used before are somewhat ad hoc. Obviously, it is important to develop a systematic theoretical tool investigating into the computation time and time complexity of EAs so that insights can be gained into them.

This section summarizes some of our recent work on this research issue[104−107]. Two techniques for estimating the computation time of EAs, analytic approach and drift analysis, are discussed here. Analytic techniques, coming from the passage time theory for Markov chains[108,109], are very useful in establishing a generic framework for studying EA's computation time. An initial attempt in analyzing EAs appeared in [110, 111], and further study has been carried out in [105, 106]. Drift analysis, based on martingale theory, draws properties of a stochastic process from its mean drift. In early days, it was used to study the properties of general Markov chain[112,113], estimate the time complexity of simulated annealing algorithms[114] and establish the convergence conditions of non-elitist EAs. Recently we applied this technique to the analysis of EAs[104,107].

8.1 Drift Analysis of Computation Time[104,107]

Almost all analyses of time complexity of EAs have been conducted for (1+1) EAs before. Theoretical results on the average computation time of population-based EAs are few. However, since the vast majority of applications of EAs use a population size greater than one, it is important to understand in depth what the real utility of population is in terms of the time complexity of EAs, when applied to combinatorial optimization problems. We have compared (1+1) EAs and $(N + N)$ EAs theoretically by deriving their first hitting time on the same problems. It was shown that a population can have a dramatic impact on EA's average computation time, changing an exponential time to a polynomial time (in the input size) in some cases. It was also shown that the first hitting probability can be improved by introducing a population. However, our results did not imply that population-based EAs will always be better than (1+1) EAs for all possible problems.

Drift analysis reduces the behavior of EAs in a higher dimensional population space E to a supermartingale on the one-dimensional space. This reduction is implemented by introducing a distance function in the population space. The analysis of the one-dimensional random walk is much easier than that of the original Markov chain. Two key points in drift analysis are: 1) to define a good distance function; and 2) to estimate the mean drift.

The basic idea behind the drift analyses is as follows. The evolution of an EA population with multiple individuals is first modeled as a random sequence, e.g., a Markov chain. Then the drift of this sequence to and from the optimal solution is analyzed (assuming an optimization problem is considered). Various bounds on the first hitting time are derived under different drift conditions. Some drift conditions cause the random sequence to drift away from the optimal solution, while others enable the sequence to drift toward the optimal solution. Some general conditions for deriving the time complexity of EAs, including conditions under which an EA will take no more than polynomial time (in problem size) to solve a problem and conditions under which an EA will take at least exponential time to solve a problem were given in [104]. Some more results, both specific and general, that were derived using the drift analysis were showed in [107].

To demonstrate the effectiveness of our approach, we have applied the general theoretical results to several well-known problems, including a classical combinatorial optimization problem—the subset sum problem. It was shown that a certain family of subset sum problems can be solved by an EA within polynomial time, while other families of such problems will need at least exponential time. Although the EAs used in our study do not include all possible variations of EAs, they do represent a fairly large class of EAs with multiple individuals using both crossover and mutation operators.

8.2 Analytical Estimation of Computation Time[105,106]

There has been some work on the analysis of time complexity of (1+1) EAs for some simple functions[103], e.g., the ONE-MAX function[115−118], the linear function[101], and the unimodal function[119,120]. Few results were obtained using EAs with a population size greater than one[104]. Because (1+1) EAs do not include recombination and population-based selection, the results on (1+1) EAs cannot be generalized to EAs with the population size greater than one. The study of how a population may have an impact on an EA's average computation time is expected to shed some light on the real utility of population-based EAs in combinatorial optimizations[99,100].

In [105], we compared (1+1) and $(N + N)$ EAs theoretically on two families of problems using the analytical approach to the passage time of Markov chains[108]. Unlike drift analysis in [104], which estimates the first hitting time from the drift of a Markov chain, these techniques calculate the first hitting time of a Markov chain directly from the transition matrix. The advantage of such analytical approach is that an exact expression of the first hitting time can be obtained for some EAs. But such exact expressions are difficult, if not impossible, to derive from transition matrices if they are too complex. We derived the first hitting time for (1+1) and $(N + N)$ EAs, respectively. Such results enable us to observe when the time should be polynomial or exponential in input size.

In [106] a general framework was built for analyzing the average hitting times of EAs based on their absorbing Markov chain models. Such a framework can facilitate answering questions at two different levels. Firstly, at the abstract level, it can help answering fundamental questions about EAs, such as, what kind of problems are easy (polynomial time) or hard (exponential time) for EAs. It can facilitate comparing first hitting times among different EAs so that insights can be gained into what makes a problem hard for an EA. Secondly, at the example level, the framework can facilitate the derivation of the average first hitting time or its bounds for a given EA and problem.

This framework was established by first studying a simple (1+1) EA and discussing what kind of problems were difficult for it. Then by taking this simple EA as a starting point, other more complex EAs were discussed, which were regarded as an improved models to the simple EA and should have some advantages over it.

In addition to the proposed general framework based on the absorbing Markov chain model, several new results were obtained in [106]. First, the general approach was applied to analyzing different EAs by comparing their first hitting times, which had never been done previously. Second, hard problems for EAs were identified and grouped into two classes in theory, i.e., the "wide-gap" problem and the "long-path" problem. Third, explicit expressions of first hitting times for several (1+1) EAs were derived. Previously, except for two simple cases given in [120] only bounds had been known[99,103]. Finally, some special cases were studied under this framework: one was to generalize the $O(n \log n)$ result for the linear function to a more general case and provide a new and more concise proof; the other was to show that an EA with crossover can solve a problem in polynomial time while the EA without crossover cannot.

9 Concluding Remarks

Evolutionary computation is an important field with many promising application areas. This paper summarizes only some of the recent advances in this field. There is a lot of work that has been done within CERCIA that has not been included in this paper, such as new EAs and their performance analysis[121−127], the application of EC approaches to dynamic problems[128−130], EC applications in data analysis and data mining[131−134], and EC approaches in other application areas[135−142]. In the future, more work still needs to be done in order to develop more efficient EC approaches, to use more mature and scientific methods to analyzing the results and the approaches themselves, and to let EC be accepted by industry and businesses.

Our experiments show that the incorporation of domain knowledge with EAs can greatly improve the quality of solution. Therefore, designing hybrid algorithm of EC with others, e.g., hill climbing, machine learning, symbolic systems etc., and within EC as well, and working with domain experts are two ways to acquire an efficient and effective EA that suits difficult tasks. In the future, high-performance EAs must be a hybrid one which can make best use of domain knowledge. It is also important to know how to best incorporate these knowledge into an EA. Much work needs to be done in tackling problems that have not been solved by conventional approaches but not just applying EC approaches to existing ones only. The development of online evolutionary systems where adaptation and evolution are performed at the same time is important too. In addition, we need to know limitations of EC approaches to the existing methods, to know when it is better to use EC approaches and when to use conventional ones. We also have to understand why and how a specific EC approach is (not) working well.

The balance between theoretical analysis and practical investigation is also an important issue. We need to strengthen theoretical analysis in order to help us understanding the general behaviors of various EC approaches because current theory is often only applicable to simple problems, but not for complex and large ones. On the other hand, apart from conventional EC approaches, more theoretical work needs to be done in understanding the behaviors of MOEA's, NNE's, Co-EC, and EAs in dynamic environment. Nevertheless, it is equally important to gain insight into the general properties of EC by simulations or practical investigation. In his book[143], after having made an extensive review and a valuable discussion over a large range of issues in EC, Fogel concluded that "although theory is always more informative than anecdotal evidence, it appears that empirical trials still hold the ability to persuade". and "in some cases, empirical trials can lead to a greater understanding and theoretical derivations"[143]. Therefore, we think that theoretical analysis and practical investigation are two essential ways to gain good insight into many important issues in EC.

The ultimate goal of EC is its applicability to solving complex real-world problems. Bear in mind that ES was first introduced because traditional approaches could not solve the problem, the wind tunnel optimization problem[1]. We need put more effort to let EC be well recognized and fully accepted by industrial society. In this regard, it is important to include industrial applications in the test suites, to establish solid prototypes and excellent demonstration and visualization systems to convince the industrial people that EC is a good alternative to their existing approaches, and to organize tutorials and workshops for industrial applications. Work also needs to be done to help people in other communities to understand the importance of EC, to fast define their problems, and to provide reasonable answers to their meaningful problems. We also need to provide students, researchers, and industrial people with more interdisciplinary (including EC) learning and training opportunities, to help small and mediate size enterprises solving their problems with various EC approaches, and to create more career opportunities for them. The extensive use of EC approaches in industry and business will greatly benefit both industrial and research communities.

References

[1] Schwefel H P. Numerical Optimization of Computer Models. Chichester, UK: John Wiley & Sons, 1981.

[2] Fogel L J, Owens A J, Walsh M J. Artificial Intelligence Through Simulated Evolution. New York, USA: John Wiley & Sons, 1966.

[3] Holland J H. Adaptation in Natural and Artificial Systems. Ann Arbor MI, USA: University of Michigan Press, 1975.

[4] De Jong K A. Genetic algorithms: A 10 year perspective. In *Proc. the First International Conference on Genetic Al-*

gorithms, Lawrence Erlbaum Associates, Hillsdale, HJ, 1985, pp.169–177.

[5] Fraser A. Simulation of genetic systems by automatic digital computers: I. introduction. *Australian Journal of Biological Science*, 1957, 10: 484–491.

[6] Koza J R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press, 1992.

[7] Fogel D B. The advantages of evolutionary computation. In *Proc. Biocomputing and Emergent Computation* (*BCEC97*), Sköve, Sweden, Singapore: World Scientific, Sept. 1997, pp.1–11.

[8] Yao X. An overview of evolutionary computation. *Chinese Journal of Advanced Software Research*, New York, NY 10011: Allerton Press, Inc., 1996, 3(1): 12–29.

[9] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing. *Science*, 1983, 220: 671–680.

[10] Szu H H, Hartley R L. Fast simulated annealing. *Physics Letters A*, 1987, 122: 157–162.

[11] Ingber L. Very fast simulated re-annealing. *Mathl. Comput. Modelling*, 1989, 12(8): 967–973.

[12] Yao X. A new simulated annealing algorithm. *Int. J. Computer Math*, 1995, 56: 161–168.

[13] Glover F. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 1986, 5: 533–549.

[14] Fogel D B. System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling. Needham Heights, MA 02194: Ginn Press, 1991.

[15] Fogel D B. Evolving artificial intelligence [Dissertation]. University of California, San Diego, CA, 1992.

[16] Fogel D B. Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and Systems*, 1993, 24: 27–36.

[17] Yao X, Liu Y, Lin G. Evolutionary programming made faster. *IEEE Trans. Evolutionary Computation*, 1999, 3(2): 82–102.

[18] Yao X, Liu Y. Fast evolution strategies. *Control and Cybernetics*, 1997, 26(3): 467–496.

[19] Hunt R A. Calculus with Analytic Geometry. New York, NY 10022-5299: Harper & Row Publ., Inc., 1986, p.322.

[20] Lee C Y, Yao X. Evolutionary programming using mutations based on the Lévy probability distribution. *IEEE Trans. Evolutionary Computation*, 2004, 8(1): 1–13.

[21] He J, Yao X. A game-theoretic approach for designing mixed mutation strategies. In *Proc. the 2005 Int. Conf. Natural Computation* (*ICNC'05*), Changsha, China, Aug. 2005, pp.27–29.

[22] Weibull J W. Evolutionary Game Theory. Cambridge, MA: MIT Press, 1995.

[23] Dutta P K. Strategies and Games. Cambridge, MA: The MIT Press, 1999.

[24] Canhi-Paz E. A survey of parallel genetic algorithms. *Calculateun Porolllles, Reseau el Sjvtem Repom's*, 1998, 10(2): 141–171.

[25] Alba E, Troya J M. A survey of parallel distributed genetic algorithms. *Complexity*, 1999, 4(4): 31–52.

[26] Riesscn G A, Williams G J, Yao X. Pepnet: Parallel evolutionary programming for constructing artificial neural networks. In *Lecture Notes in Computer Science*, Berlin: Springer-Verlag, 1997, 1213: 35–45.

[27] Tongchim S, Yao X. Parallel Evolutionary Programming. In *Proc. the 2004 Congress on Evolutionary Computation* (*CEC'04*), Portland, Oregon, USA, June 19–23, 2004, pp.1362–1367.

[28] Yao X, Liu Y. A new evolutionary system for evolving artificial neural networks. *IEEE Trans. Neural Networks*, 1997, 8(3): 694–713.

[29] Yao X, Liu Y. Making use of population information in evolutionary artificial neural networks. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 1998, 28(3): 417–425.

[30] Selfridge O G. Pandemonium: A paradigm for learning. In *Mechanization of Thought Progresses: Proc. a Symp. Held at the National Physical Lab., HMSO*, London, 1958, pp.513–526.

[31] Hansen L K, Salamon P. Neural network ensembles. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1990, 12(10): 993–1001.

[32] Sharkey A. On combining artificial neural nets. *Connection Science*, 1996, 8: 299–313.

[33] Jacobs R A, Jordan M I, Nowlan S J, Hinton G E. Adaptive mixtures of local experts. *Neural Computation*, 1991, 3: 79–87.

[34] Jacobs R A, Jordan M I, Barto A G. Task decomposition through competition in a modular connectionist architecture: The what and where vision task. *Cognitive Science*, 1991, 15: 219–250.

[35] Drecker H, Cortes C, Jackel L D, LeCun Y, Vapnik V. Boosting and other ensemble methods. *Neural Computation*, 1994, 6: 1289–1301.

[36] Liu Y, Yao X, Higuchi T. Evolutionary ensembles with negative correlation learning. *IEEE Trans. Evolutionary Computation*, 2000, 4(4): 380–387.

[37] Liu Y, Yao X. Negative corrected neural networks can produce best ensembles. *Australian Journal of Intelligent Information Processing Systems*, 1997, 4(3/4): 176–185.

[38] Liu Y, Yao X. Ensemble learning via negative correlation. *Neural Networks*, 1999, 12: 1399–1404.

[39] Liu Y, Yao X. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 1999, 29: 716–725.

[40] Liu Y, Yao X. Toward designing neural network ensembles by evolution. In *Proc. 5th Int. Conf. Parallel Problem Solving from Nature* (*PPSN V*), Berlin, Germany: Springer-Verlag, 1998, Lecture Notes in Computer Science, 1498: 623–632.

[41] Liu Y, Yao X, Darwen P. How to make best use of evolutionary learning. In *Complex Systems: From Local Interactions to Global Phenomena*, Stocker R, Jelinek H, Durnota B (eds.), Amsterdam, Netherlands: IOS Press, 1996, pp.229–242.

[42] Michie D, Spiegelhalter D J, Taylor C C. Machine Learning, Neural and Statistical Classification. London, UK: Ellis Horwood Limited, 1994.

[43] ftp://ics.uci.edu/pub/machine-learning-databases.

[44] Monirul Islam M, Yao X, Murase K. A constructive algorithm for training cooperative neural network ensembles. *IEEE Trans. Neural Networks*, 2003, 14(4): 820–834.

[45] Wang Z, Yao X, Xu Y. An improved constructive neural network ensemble approach to medical diagnoses. *Lecture Notes in Computer Science*, 2004, 3177: 572–577.

[46] Brown G, Wyatt J, Harris R, Yao X. Diversity creation methods: A survey and categorisation. *Journal of Information Fusion*, 2005, 6(1): 5–20.

[47] Yao X, Liu Y. Evolving neural network ensembles by minimization of mutual information. *International Journal of Hybrid Intelligent Systems*, 2004, 1(1): 12–21.

[48] Chandra A, Yao X. Evolutionary framework for the construction of diverse hybrid ensembles. In *Proc. the 13th European Symposium on Artificial Neural Networks* (*ESANN'2005*), Bruges, Belgium, April 27–29, 2005, pp.253–258.

[49] Chandra A, Yao X. DIVACE: Diverse and accurate ensemble learning algorithm. *Lecture Notes in Computer Science,* 2004, 3177: 619–625.

[50] Chandra A, Yao X. Ensemble learning using multiobjective evolutionary algorithms. *Journal of Mathematical Modelling and Algorithms,* May 2005 (to appear).

[51] Abbass H A. A memetic Pareto evolutionary approach to artificial neural networks. In *Proc. the 14th Australian Joint Conference on Artificial Intelligence*, Berlin, 2000, pp.1–12.

[52] Liu Y, Yao X. Learning and evolution by minimization of mutual information. *Lecture Notes in Computer Science*, 2002, 2439: 495–504.

[53] Chandra A, Yao X. Evolving hybrid ensembles of learning machines for better generalisation. Submitted to *Neurocomputing*, Elsevier, July 2005.

[54] Coello Coello C A. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and*

Information Systems: An International Journal, 1999, 1(3): 269–308.

[55] Khare V, Yao X, Deb K. Performance scaling of multiobjective evolutionary algorithms. *Lecture Notes in Computer Science*, 2003, 2632: 376–390.

[56] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 2002, 6(2): 182–197.

[57] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[58] Corne D W, Knowles J D, Oates M J. The Pareto envelope-based selection algorithm for multiobjective optimization. *Lecture Notes in Computer Science*, 2000, 1917: 839–848.

[59] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[60] Atashkari K, Nariman-Zadeh N, Pilechi A, Yao X. Thermodynamic Pareto optimization of turbojet engines using multiobjective genetic algorithms. *International Journal of Thermal Sciences*, 2005 (to appear).

[61] Nariman-Zadeh N, Atashkari K, Jamali A, Pilechi A, Yao X. Inverse modelling of multiobjective thermodynamically optimized turbojet engines using GMDH-type neural networks and evolutionary algorithms. *Engineering Optimization*, 2005, 37(5): 437–462.

[62] Axelrod R M. The Evolution of Cooperation. New York: Basic Books, 1984.

[63] Axelrod R M. The Evolution of Strategies in the Iterated Prisoner's Dilemma. Genetic Algorithms and Simulated Annealing, Chap. 3, Davis L (ed.), Los Altos, California: Morgan Kaufmann, 1987, pp.32–41.

[64] Darwen P, Yao X. Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense. *International Journal of Computational Intelligence and Applications*, 2002, 2(1): 83–107.

[65] Frean M. The evolution of degrees of cooperation. *J. Theoretical. Bio.*, 1996, 182(4): 549–559.

[66] Harrald P G, Fogel D B. Evolving continuous behaviors in the iterated prisoner's dilemma. *Biosystems*, 1996, 37: 135–145.

[67] Chong S Y, Yao X. The impact of noise on iterated prisoner's dilemma with multiple levels of cooperation. In *Proc. the 2004 Congress on Evolutionary Computation* (*CEC'04*), Portland, Oregon, USA, June 19–23, 2004, pp.348–355.

[68] Colman A M. Game Theory and Experimental Games. Oxford, England: Pergamon Press, 1982, p.159.

[69] Glance N S, Huberman B A. The dynamics of social dilemmas. *Sci. Am.*, March 1994, pp.58–63.

[70] Albin P. Approximations of cooperative equilibria in multiperson prisoners' dilemma played by cellular automata. *Math. Sot. Sci.*, 1992, 24: 293–319.

[71] Yao X. Evolutionary stability in the n-person iterated prisoner's dilemma. *BioSystems*, 1996, 37(3): 189–197.

[72] Seo Y G, Cho S B, Yao X. Emergence of Cooperative Coalition in NIPD game with Localization of Interaction and Learning. In *Proc. the 1999 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, USA, July 1999, 2: 877–884.

[73] Seo Y G, Cho S B, Yao X. The impact of payoff function and local interaction on the N-player iterated prisoner's dilemma. *Knowledge and Information Systems: An International Journal*, 2000, 2(4): 461–478.

[74] Yao X, Darwen P. The experimental study of N-player iterated prisoner's dilemma. *Informatica*, 1994, 18: 435–450.

[75] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evolutionary Computation,* 2000, 4(3): 284–294.

[76] Joines J, Houck C. On the use of nonstationary penalty functions to solve nonlinear constrained optimization problems with Gas. In *Proc. IEEE Int. Conf. Evolutionary Computing*, Piscataway, NJ, 1994, pp.579–584.

[77] Runarsson T P, Yao X. Search biases in constrained evolutionary optimization. *IEEE Trans. Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 2005, 35(2): 233–243.

[78] Salcedo-Sanz S, Yao X. A hybrid Hopfield network—Genetic algorithm approach for the terminal assignment problem. *IEEE Trans. System, Man and Cybernetics B*, 2004, 34(6): 2343–2353.

[79] Xu Y, Salcedo-Sanz S, Yao X. Non-standard cost terminal assignment problems using tabu search approach. In *Proc. the 2004 Congress on Evolutionary Computation* (*CEC'04*), IEEE Press, Portland, Oregon, USA, June 19–23, 2004, pp.2335–2340.

[80] Abuali F N, Schoenefeld D A, Wainwright R L. Terminal assignment in a communications network using genetic algorithms. In *Proc. 22nd Annu. ACM Computer Science Conf.*, 1994, pp.74–81.

[81] Khuri S, Chiu T. Heuristic algorithms for the terminal assignment problem. In *Proc. 1997 ACM Symp. Applied Computing*, 1997, pp.247–251.

[82] Brudaru I. Grouping-based hybrid genetic algorithm for the terminal assignment problem. In *Proc. 7th Int. Research/Expert Conf. Trends Development of Machinery Associated Techniques*, Barcelona, Spain, 2003.

[83] Salcedo-Sanz S, Xu Y, Yao X. A hybrid genetic algorithm-Hopfield network for task assignment in distributed computer networks. *Computers and Operations Research*, 2005 (to appear).

[84] Salcedo-Sanz S, Xu Y, Yao X. Meta-heuristic algorithms for FPGA segmented channel routing problems with non-standard cost functions. *Genetic Programming and Evolvable Machines*, Vol.6, 2005 (to appear).

[85] Xu Y, Xu S C, Wu B X. Traffic grooming in unidirectional WDM ring networks using genetic algorithms. *Computer Communications*, 2002, 25(13): 1185–1194.

[86] Xu Y, Xu S C, Wu B X. Strictly nonblocking grooming of dynamic traffic in unidirectional SONET/WDM rings using genetic algorithm. *Computer Network*, 2003, 41(2): 227–245.

[87] Liu K H, Xu Y. A new approach to improving the grooming performance with dynamic traffic in SONET rings. *Computer Networks*, 2004, 46(2): 181–195.

[88] Liu K H, Xu Y. Combining GA with splitting methods to the rearrangeably nonblocking grooming of dynamic traffic in SONET ring networks. *Photonic Network Communications*, 2005, 10(2): 179–192.

[89] Xu Y, Yao X. Lower bound on number of ADMs in WDM rings with nonuniform traffic demands. *Electronics Letters*, 2004, 40(13): 824–825.

[90] Xu Y, Salcedo-Sanz S, Yao X. Metaheuristic approaches to traffic grooming in WDM optical networks. *International Journal of Computational Intelligence and Applications*, 2005 (to appear).

[91] Schnier T, Yao X, Liu P. Digital filter design using multiple Pareto fronts, soft computing—A fusion of foundations. *Methodologies and Applications*, 2004, 8(5): 332–343.

[92] Schnier T, Yao X. Evolutionary design calibration. *Lecture Notes in Computer Science*, 2001, 2210: 26–37.

[93] Schnier T, Yao X. Using negative correlation to evolve fault-tolerant circuits. *Lecture Notes in Computer Science*, 2003, 2606: 35–46.

[94] Thornes J E. The cost-benefit of winter road maintenance in the UK. In *Proc. 8th International Road Weather Conference*, Birmingham, UK, 1996, pp.1–10.

[95] Handa H, Chapman L, Yao X. Dynamic salting route optimisation using evolutionary computation. In *Proc. 2005 Congress on Evolutionary Computation* (*CEC'05*), IEEE Press, Edinburgh, UK, Sept. 2–5, 2005, (to appear).

[96] Chapman L, Thornes J E. Real-time sky-view factor calculation and approximation. *Journal of Atmospheric and Oceanic Technology*, 2004, 21: 730–741.

[97] Li J, Yao X, Frayn C, Khosroshahi H G, Raychaudhury S. An evolutionary approach to modeling radial brightness distributions in elliptical galaxies. *Lecture Notes in Computer Science*, 2004, 3242: 591–601.

[98] Washbrook B, Li J. The Application of evolutionary computation to the analysis of the profiles of elliptical galaxies: A maximum likelihood approach. In *Proc. 2005 IEEE Congress on Evolutionary Computation* (*CEC'05*), Edinburgh, UK, September 2–5, 2005 (to appear).

[99] Rudolph G. Finite Markov chain results in evolutionary computation: A tour d'Horizon. *Fundamenta Informaticae*, 1998, 35(1-4): 67–89.

[100] Eiben A E, Rudolph G. Theory of evolutionary algorithms: A bird's eye view. *Theoretical Computer Science*, 1999, 229(1-2): 3–9.

[101] Droste S, Jansen T, Wegener I. A rigorous complexity analysis of the (1+1) evolutionary algorithm for linear functions with Boolean inputs. *Evolutionary Computation*, 1998, 6(2): 185–196.

[102] Beyer H G, Schwefel H P, Wegener I. How to analyze evolutionary algorithms. *Theoretical Computer Science*, 2002, 287(1): 101–130.

[103] Droste S, Jansen T, Wegener I. On the analysis of the (1+1) evolutionary algorithms. *Theoretical Computer Science*, 2002, 276(1-2): 51–81.

[104] He J, Yao X. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 2001, 127(1): 57–85.

[105] He J, Yao X. From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, 2002, 6(5): 495–511.

[106] He J, Yao X. Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, 2003, 145(1-2): 59–97.

[107] He J, Yao X. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 2004, 3(1): 21–35.

[108] Syski R. Passage Times for Markov Chains. Amsterdam: IOS Press, 1992.

[109] Iosifescu M. Finite Markov Processes and Their Applications. Chichester: John Wiley & Sons, 1980.

[110] Bäck T, Fogel D B, Michalewicz Z (eds.). Handbook of Evolutionary Computation. Oxford: Oxford University Press, 1997.

[111] Spears W M. Evolutionary Algorithms: The Role of Mutation and Recombination. Berlin: Springer, 2000.

[112] Hajek B. Hitting time and occupation time bounds implied by drift analysis with applications. *Advances in Applied Probability*, 1982, 14(3): 502–525.

[113] Meyn S P, Tweedie R L. Markov Chains and Stochastic Stability. New York: Springer-Verlag, 1993.

[114] Sasaki G H, Hajek B. The time complexity of maximum matching by simulated annealing. *J. ACM*, 1988, 35(2): 387–403.

[115] Bäck T. The interaction of mutation rate, selection and self-adaptation within a genetic algorithm. In *Proc. Parallel Problem Solving from Nature* (*PPSN*) II, Männer R, Manderick B (eds.), Amsterdam, The Netherlands, North Holland, 1992, pp.85–94.

[116] Mühlenbein H. How genetic algorithms really works I: Mutation and hill-climbing. In *Proc. Parallel Problem Solving from Nature* (*PPSN*) II, Männer R, Manderick B (eds.), Amsterdam, North Holland, 1992, pp.15–25.

[117] Rudolph G. Convergence Properties of Evolutionary Algorithms. Hamburg, Germany: Springer-Verlag, 1997.

[118] Garnier J, Kallel L, Schoenauer M. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 1999, 7(2): 167–203.

[119] Rudolph G. How mutation and selection solve long path problems in polynomial expected time. *Evolutionary Computation*, 1996, 4(2): 207–211.

[120] Garnier J, Kallel L. Statistical distribution of the convergence time of evolutionary algorithms for longpath problems. *IEEE Trans. Evolutionary Computation*, 2000, 4: 16–30.

[121] Lu Q, Yao X. Clustering and Learning Gaussian Distribution for Continuous Optimization. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 2005, 35(2): 195–204.

[122] Tang E K, Suganthan P N, Yao X. Generalized LDA using relevance weighting and evolution strategy. In *Proc. 2004 Congress on Evolutionary Computation* (*CEC'04*), Portland, Oregon, USA, June 19–23, 2004, pp.2230–2234.

[123] Zou P, Zhou Z, Chen G L, Yao X. A novel memetic algorithm with random multi-local-search: A case study of TSP. In *Proc. 2004 Congress on Evolutionary Computation* (*CEC'04*), Portland, Oregon, USA, June 19–23, 2004, pp.2335–2340.

[124] Matsumura Y, Yao X, Wyatt J L, Ohkura K, Ueda K. Robust evolution strategies use adaptive search strategies to design continuous-time recurrent neural networks. In *Proc. 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems* (*CIRAS 2003*), Singapore, December 15–18, 2003.

[125] Runarsson T P, Yao X. Evolutionary search and constraint violations. In *Proc. 2003 Congress on Evolutionary Computation* (*CEC'03*), IEEE Press, Piscataway, NJ, USA, December 8–12, 2003, pp.1414–1419.

[126] He J, Yao X. An analysis of evolutionary algorithms for finding approximation solutions to hard optimisation problems. In *Proc. 2003 Congress on Evolutionary Computation* (*CEC'03*), IEEE Press, Piscataway, NJ, USA, December 8–12, 2003, pp.2004–2009.

[127] Tianfield H, Tian J, Yao X. On the architectures of complex multi-agent systems. In *Proc. Workshop on Knowledge Grid and Grid Intelligence*, held at the *2003 IEEE/WIC International Conference on Web Intelligence/Intelligent Agent Technology,* Halifax, Canada, October 13–16, 2003, pp.195–206.

[128] Yang S, Yao X. Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft Computing*, August 2004 (to appear).

[129] Matsumura Y, Fujimoto N, Yao X, Wyatt J, Hagihara K. Near-optimal dynamic grid task scheduling of evolution strategies. In *Proc. The 4th International Conference on Advanced Mechatronics*, Japan, 2004, pp.175–180.

[130] Yang S, Yao X. Dual population-based incremental learning for problem optimization in dynamic environments. In *Proc. The 7th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, Japan, 2003, pp.49–56.

[131] Ma P C H, Chan K C C, Yao X, Chiu D K Y. An evolutionary clustering algorithm for gene expression microarray data analysis. *IEEE Trans. Evolutionary Computation*, 2005 (to appear).

[132] Au W H, Chan K C C, Yao X. Data mining by evolutionary learning for robust churn prediction in the telecommunications industry. *IEEE Trans. Evolutionary Computation*, 2003, 7(6): 532–545.

[133] Salim M, Yao X. Evolving SQL Queries for Data Mining. *Lecture Notes in Computer Science*, 2002, 2412: 62–67.

[134] Zhang C, Yao X, Yang J. An evolutionary approach to materialized views selection in a data warehouse environment. *IEEE Trans. Systems, Man and Cybernetics, Part C*, 2001, 31(3): 282–294.

[135] Tang E K, Suganthan P N, Yao X, Qin A K. Linear dimensionality reduction using relevance weighted LDA. *Pattern Recognition*, April 2005, 38(4): 485–493.

[136] Sarker R, Yao X. Simulated annealing for solving a manufacturing batch-sizing problem. *International Journal of Operations and Quantitative Management*, 2003, 9(1): 65–80.

[137] Yu J X, Yao X, Choi C H, Gou G. Materialized view selection as constrained evolutionary optimization. *IEEE Trans. Systems, Man and Cybernetics, Part C*, 2003, 33(4): 458–467.

[138] Lin J, Cheong B H, Yao X. Universal multiobjective function for optimising superplastic-damage constitutive equa-

tions. *Journal of Materials Processing Technology*, 2002, 125-126: 199–205.

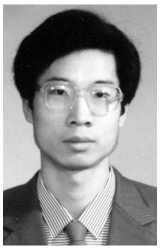
[139] Li B, Lin J, Yao X. A novel evolutionary algorithm for determining unified creep damage constitutive equations. *International Journal of Mechanical Sciences*, 2002, 44(5): 987–1002.

[140] Tang E K, Suganthan P N, Yao X. Nonlinear Feature Extraction Using Evolutionary Algorithm. *Lecture Notes in Computer Science*, 2004, 3316: 1014–1019.

[141] Matsumura Y, Fujimoto N, Yao X. Wyatt J, Hagihara K. Near-optimal dynamic grid task scheduling of evolution strategies. In *Proc. The 4th International Conference on Advanced Mechatronics*, 2004, pp.175–180.

[142] Matsumura Y, Fujimoto N, Yao X, Wyatt J, Hagihara K. Evolution strategies on grid computing. In *Proc. the China-Japan Joint Conference on Mechatronics*, CJ-H2, Suzhou, China, 2004, pp.105–106.

[143] Fogel D B. Evolutionary Computation—Toward a New Philosophy of Machine Intelligence. 2nd Edition, Piscataway, NJ: IEEE Press, 2000.



Xin Yao obtained his B.Sc. degree in 1982 from the University of Science and Technology of China (USTC) in Hefei, M.Sc. degree in 1985 from the North China Institute of Computing Technologies (NCI) in Beijing and Ph.D. degree in 1990 from USTC, all in computer science. He joined the University of Birmingham from Australia as a professor of computer science in 1999. He is a fellow of IEEE, the editor-in-chief of *IEEE Transactions on Evolutionary Computation*, an associate editor of several other international journals, and the editor of the book series on "Advances in Natural Computation" from World Scientific Publishing Co. He has been an invited keynote or plenary speaker of more than 35 international conferences in 11 different countries and a chair/co-chair of 27 international conferences. He is an IEEE Computational Intelligence Society Distinguished Lecturer. He won the prestigious IEEE Donald G. Fink Prize Paper Award (2001). He is currently the Director of The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) at the University of Birmingham, UK. He is also a Distinguished Visiting Professor and Cheung Kong Scholar at USTC and a visiting professor at three other universities. He has more than 200 research publications, including 80 refereed journal papers. His major research interests include evolutionary computation and neural network ensembles.

Yong Xu received the M.Sc. degree in applied optics and lasers from Fujian Normal University in 1987 and the Ph.D. degree in physics specializing in nature-inspired approaches to optical network design from Xiamen University in 2002. He is a member of IEEE and now a professor with Fujian Normal University and a research fellow with the University of Birmingham. From 2001 to 2002 he was a research assistant with the City University of Hong Kong. He is a reviewer for a number of leading international journals and international conferences and serves as a program committee member for many international conferences. He is an organizer of a workshop in PPSN04 and a guest editor of a special issue in *International Journal of Computational Intelligence and Applications*. He has published more than 50 papers and undertaken many research projects. His research interests include evolutionary computation, e.g., neural networks, genetic algorithms and tabu search etc., global optimization, and the optimal design of telecommunication networks.