A New Evolution Strategy and Its Application to Solving Optimal Control Problems*

M. M. A. HASHEM**, Keigo WATANABE***
and Kiyotaka IZUMI****

Evolution strategies (ESs) are search algorithms which imitate the principles of natural evolution as a method to solve parameter optimization problems numerically. The effectiveness and simplicity of ES algorithms have lead many people to believe that they are the methods of choice for difficult, real-life problems superseding traditional search techniques. However, the inherent strength of the ES algorithms largely depends upon the choice of a suitable crossover and mutation technique in their application domains. This paper discusses a new ES in which both a subpopulation-based arithmetical crossover (SBAC) and a time-variant mutation (TVM) operator are used. The SBAC operator explores promising areas in the search space with different directivity while the TVM operator exploits fast (but not premature) convergence with high precision results. Thus, a balance between exploration and exploitation is achieved in the evolutionary process with these combined efforts. The TVM also acts as a fine local tuner at the converging stages for high precision solutions. Its action depends upon the age of the populations, and its performance is quite different from the Uniform Mutation (UM) operation. The efficacy of the proposed methods is illustrated by solving discrete-time optimal control models which are frequently used in the applications.

Key Words: Evolution Strategy, Evolutionary Computation, Linear-Quadratic Control, Push-Cart Control, Discrete-Time Optimal Control, Mutation, Arithmetical Crossover, Intelligent Systems

1. Introduction

Recently, a wide variety of evolutionary algorithms (EAs) have been proposed to solve different kinds of optimization problems and for machine learning^{(1)~(7)}. Among these, the evolution strategies (ESs)^{(1),(2),(4),(7)} proved able to efficiently solve difficult optimization problems numerically. The ESs generally represent an individual as real-valued vectors.

The main objective behind such implementation is to move the algorithm very close to the problem space. Such a move forces, but also allows, the genetic operators to be more problem specific by utilizing some specific characteristics of real space⁽⁴⁾. The effectiveness and simplicity of ES algorithms have lead many people to believe that they are the methods of choice for difficult, real-life problems superseding traditional search techniques^{(2),(4),(7)}. However, they are not without their limitations. In particular, the choice of a good problem-specific genetic operator can make a considerable difference in the exploration, exploitation, and often even the feasibility of the evolutionary search^{(2),(4),(8),(9)}.

Schwefel⁽²⁾ suggested, but neither implemented nor tested, that the form of the selection/reproduction profile should be parameterized (perhaps on the level of competing sub-populations). The outcome would at least be a variable selection pressure, and this is very important for the balance between exploration

* Received 2nd June, 1997

** Graduate School of Science and Engineering, Faculty of Engineering Systems and Technology, Saga University, 1-Honjomachi, Saga 840-8502, Japan

*** Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University, 1-Honjomachi, Saga 840-8502, Japan

**** Department of Mechanical Engineering, Faculty of Science and Engineering, Saga University, 1-Honjomachi, Saga 840-8502, Japan

and exploitation of the evolutionary search. However, the theory of ES influenced to the discovery of the so called “Evolution Window”⁽⁷⁾. The meaning of this term is that changes (mutational jumps) lead to evolutionary progress only when they lie within a narrowly confined and calculable step-width band. Schwefel⁽²⁾ also reported that optimum mutation rates are inversely proportional to the number of decision variables involved and proportional to the distance from the optimum. The authors have already found this juncture in their study⁽¹²⁾. Rechenberg⁽⁷⁾ indicated that mutation steps and recombination (crossover) steps which fall outside the evolution window are ineffective. It is a very difficult task to set up an optimum mutation step for the problem at hand. Thus, it is apparent that an optimal compromise must exist between the frequency of success and the degree of progress in the mutations. Therefore, the choice of a suitable mutation step, especially in the presence of many variables, becomes a decisive factor for the convergence of the ESs^{(4),(7)}. The inherent strength of the ES algorithms depends largely upon the choice of suitable crossover and mutation techniques in their application domains. The crossover operator is very important in exploring promising areas in the search space, and the mutation operator helps in exploiting fast convergence.

On the other hand, generally it is a very difficult task to design and implement algorithms for the solution of optimal control problems, especially for nonlinear or large-scale systems. Some algorithms exist in the literature for this class of problems. However, these algorithms are not effective for problems of moderate size and complexity, suffering from what is called “the curse of dimensionality”⁽⁴⁾. Optimal control problems are quite difficult to deal with numerically even though the system is linear.

The main objective of this paper is to find a balance between exploration and exploitation with the help of a new crossover and mutation technique and to show that the proposed ES is quite capable of solving discrete-time optimal control problems. Therefore, we proposed a subpopulation-based arithmetical crossover (SBAC) and a time-variant mutation (TVM) operator in the existing ES algorithm. The SBAC explores promising areas in the search space with different directivity while the TVM exploits fast (but not premature) convergence with high precision results. With a view to demonstrating effectiveness, discrete-time optimal control problems^{(4),(10),(11)}, frequently used in the applications of optimal control are solved with different control steps. The results obtained indicate that an optimal compromise has been found between exploration and exploi-

tation with the introduction of proposed operators in the ES algorithm.

2. The Evolution Strategy (ES)

The ES is a probabilistic search algorithm which maintains a population of individuals, $\Pi(t) = \{\phi_1^t, \dots, \phi_n^t\}$ for generation t . Each individual represents a potential solution to the problem at hand, and, in any evolution program, is implemented as some (possibly complex) data structure ξ (chromosome representation). Each solution ϕ_i^t is evaluated to produce some measure of its “fitness”. Then a new population (generation $t+1$) is formed by selection (reproduction), crossover (recombination) and mutation operations. The members of the new population undergo transformations by means of “genetic” operators to form new solutions. There are a higher order transformation, ξ_i (crossover type), which creates new individuals ($\xi_i: \xi \times \xi \rightarrow \xi$) and a unary transformation, ξ'_i (mutation type), which creates new individuals by a small change in a single individual ($\xi'_i: \xi \rightarrow \xi$) with zero-mean Gaussian noise^{(2),(4)}. After some number of generations the program converges; it is hoped that the best individual represents a near optimum (reasonable) solution. The structure of an evolution strategy is shown in Fig. 1, and it is described below.

2.1 Initial population

The initial population is generated using uniform random numbers (URN). If we generate a variable u_1 within the range $-100 \leq u_1 \leq 100$, then we must use the URN $[-100, 100]$. Thus, for the other variable u_2 with an alternative range, we use other URN. Thus the first

Fig. 1 The structure of an evolution strategy

individual in the population $\phi_1 = [u_1, u_2, \dots, u_n]^T$ is generated. Similarly, the remaining individuals for the population $\phi_2, \dots, \phi_\mu$ are generated using the same method, where μ denotes the number of individuals in the population.

2.2 Arithmetical crossover

To generate a child, an arithmetical crossover or intermediate crossover is used. This generates a new variable by a linear combination of the corresponding object variables and can generate any point on the lines connecting the corresponding vertices and in the volume of the hypercube defined by the variable sequences⁽⁶⁾. Thus two offspring $\{\xi_1, \xi_2\}$ are produced by a linear combination of their parents $\{\phi_1, \phi_2\}$, i.e.,

$$\xi_1 = \alpha\phi_1 + (1-\alpha)\phi_2 \quad (1)$$

$$\xi_2 = (1-\alpha)\phi_1 + \alpha\phi_2 \quad (2)$$

where the parents $\{\phi_1, \phi_2\}$ are selected randomly from the population μ , and α is selected from the URN[0,1].

Thus μ number of new individuals (children) are created uniformly in the current generation. In this paper, we called this crossover method the conventional method.

2.3 Mutation

Mutation plays a significant role in the progress of the search (exploitation) and fine tuning for the ESs. Smaller changes occur more often than larger ones in biological evolution. According to this observation, ES uses zero-mean Gaussian noise to perturb all object variables of a child^{(2),(4)}. The mutation for a child i is made as

$$\xi'_i = \xi_i + N(0, \sigma^2) \quad (3)$$

where $N(\cdot)$ is a zero-mean Gaussian random number vector and σ denotes the standard deviation, $\sigma = [\sigma_1, \dots, \sigma_n]^T$.

When σ is fixed throughout the evolution process, we call this type of mutation a uniform mutation (UM).

2.4 Evaluation

After mutation, each child is evaluated as to its cost function (fitness) for a possible solution in each generation. These evaluations are preserved for creating a new generation.

2.5 Alternation of generation

In the alternation of generation, $(\mu + \mu)$ -ES is used. Among μ parents, which were evaluated at the former generation, and μ children, which are evaluated in the current generation, the $\mu + \mu$ individuals are ordered in proportion to the cost function, and the best μ individuals are selected for the next generation.

3. Subpopulation-Based Arithmetical Crossover (SBAC) Operator

We proposed a subpopulation-based arithmetical crossover (SBAC) operator in which the competing

Fig. 2 Max-mean arithmetical crossover with subpopulation's elite

subpopulation's (subgroup's) elite and the mean strength of that subpopulation except the elite are used in the crossover technique^{(10)~(12)}. This technique has a very strong directivity to the elite as shown in Fig. 2. The SBAC explores promising areas in the search space with different directivity towards the optimum point. There is less possibility of being trapped in local minima while attempting to attain the optimum. The method is described below.

We divide the μ populations into l number of competing subpopulations. We define $\phi_{j,\max}$ as an elite individual that maximizes a cost function within the j -th subpopulation and $\bar{\phi}_j$ as a mean strength of the j -th subpopulation excluding the $\phi_{j,\max}$. We define the crossover for the competing subpopulation j as follows:

$$\xi_1 = \alpha\phi_{j,\max} + (1-\alpha)\bar{\phi}_j \quad (4)$$

$$\xi_2 = (1-\alpha)\phi_{j,\max} + \alpha\bar{\phi}_j \quad (5)$$

where α is selected from URN[0,1]. Note that the children are generated for each subpopulation by using $(\mu/l + \mu/l)$ -ES.

4. Time-Variant Mutation (TVM) Operator

The inherent strength of the ES algorithm, towards convergence and high precision results, lies in the choice of the mutation steps (standard deviations). A special dynamic time-variant mutation (TVM) operator is proposed, aiming at both improving the fine local tuning and reducing the disadvantage of uniform mutation in the ES algorithm. Moreover, it can exploit fast (but not premature) convergence. The TVM is defined for a child i as

$$\xi'_i = \xi_i + N(0, \sigma^2(t)) \quad (6)$$

where $\sigma(t) = [\sigma_1(t), \dots, \sigma_n(t)]^T$ is the time-variant standard deviation vector at the generation, t , and $\sigma(t)$ is defined as

$$\sigma(t) = [1 - r^{(1-\frac{t}{T})}] \quad (7)$$

where r is selected from the URN[0,1], T is the maximal generation number, and γ is a real-valued parameter determining the degree of dependency on the progress and success of the evolution. Normally,

Fig. 3 Characteristics of $\sigma(t)$ with $\gamma=6.0$ and $T=200$

the value of γ should be selected as greater than unity. However, using a value of γ greater than unity can produce high precision final results in the presence of many variables. With a value of γ less than unity, it may not be able to converge to the final result in the presence of many variables, and this may decrease the success and progress of the evolution process.

The function $\sigma(t)$ returns a value in the range $[0, 1]$, which falls within the evolution window such that the probability of $\sigma(t)$ being close to 0 increases as the age of the population t increases. This property of $\sigma(t)$ causes searching of the problem space uniformly initially (when t is small) and very locally at larger stages of t . Thus, it increases the probability of generating the new mutation step very close to its successor mutation step rather than making a uniform mutation choice for the whole evolution process. The generation of a typical time-variant standard deviation $\sigma(t)$ with $\gamma=6.0$ is shown in Fig. 3.

5. Optimal Control Problems

Two simple discrete-time optimal control models frequently used in applications have been selected as the test problems for the proposed methods: the linear-quadratic control problem (for minimization) and the push-cart control problem (for maximization). We discuss them simultaneously in the following subsections.

5.1 The Linear-Quadratic Control (LQC) problem

The first test problem is the dynamic one-dimensional linear-quadratic control problem^{(4),(11)}. The problem is to minimize the following cost function:

$$J_1 = qx^2(N) + \sum_{k=0}^{N-1} [sx^2(k) + ru^2(k)] \quad (8)$$

subject to

$$x(k+1) = ax(k) + bu(k), \quad k=0, 1, \dots, (N-1) \quad (9)$$

where $x(0)$ is a given initial state, a, b, q, s, r are

given constants, $x(k) \in R$ is the state at time k , and $u(k) \in R$ is the control input of the system. N is the total number of control steps involved in the system. When the cost (fitness) function (8) subject to (9) is minimized, the desired control inputs, $u = [u(0), \dots, u(N-1)]^T$ are found.

The optimal cost of (8) subject to (9) can be analytically expressed as

$$J_1^* = K(0)x^2(0) \quad (10)$$

where $K(k)$ is the solution of the Riccati equation:

$$K(k) = s + \frac{ra^2K(k+1)}{[r + b^2K(k+1)]}, \quad K(N) = q \quad (11)$$

5.2 The Push-Cart Control (PCC) problem

The second test problem is the discretized push-cart problem^{(4),(10)}. The problem is to maximize the total distance $x_1(N)$ traveled in a given time (a unit), minus the total effort with small control inputs. The system is second order. The discrete model of the cart is given by

$$x_1(k+1) = x_2(k) \quad (12)$$

$$x_2(k+1) = 2x_2(k) - x_1(k) + \frac{1}{N^2}u(k) \quad (13)$$

where $x_1(k) \in R$ is the cart migration distance, $x_2(k) \in R$ is the cart speed, and $u(k) \in R$ is the control input of the system. At unit time, the performance index to be maximized is:

$$J_2 = x_1(N) - \frac{1}{2N} \sum_{k=0}^{N-1} u^2(k) \quad (14)$$

The first term on the right-hand side in (14) is evaluated as the total distance traveled by the cart, and the second term is evaluated as the total contribution of the control inputs. When the cost (fitness) function (14) subject to (12) and (13) is maximized, the desired control inputs u are found.

The optimal cost of (14) subject to (12) and (13) can be analytically expressed as

$$J_2^* = \frac{1}{3} - \frac{3N-1}{6N^2} - \frac{1}{2N^3} \sum_{k=0}^{N-1} k^2 \quad (15)$$

6. Simulation Examples

6.1 Implementation details

The proposed ES is implemented using a population of 60 individuals. The competing subpopulation number l is selected as 10 and 6 for the LQC and PCC problems respectively. An individual in the population is represented by

$$\phi = [u(0), u(1), \dots, u(N-1)] \quad (16)$$

The chromosomes, the control inputs u , are represented by floating-point vectors. Each element of the chromosome is initially generated randomly but within a desired domain, and the operators are carefully designed to preserve this domain during evolution.

The fixed domain for each chromosome $u(i)$ is

assumed to be $[-100, 100]$ and $[-10, 10]$ for the LQC and PCC problems, respectively, because actual solutions fall within these ranges for the class of tests performed. The initial state $x(0)$ is assumed to be 100.0 and the constants, a, b, q, s and r , are selected as 1.0, 0.01, 1.0, 1.0, and 1.0, respectively, for the LQC problem. The initial cart migration distance $x_1(0)$ and initial cart speed $x_2(0)$ are assumed to be 0.0 for the PCC problem.

6.2 Simulation example I: The SBAC method with TVM and UM operators

In this example, we applied the proposed SBAC method and TVM along with the UM in the ES for solving both the problems. The intuitive reason behind this was to demonstrate the effectiveness of TVM over UM with SBAC and to find a suitable γ value for this class of problems. Mutation steps (standard deviations) for UM are chosen as 1.0, 0.1 and 0.01. To observe the effects of the degree of dependency on evolution, the parameter γ for TVM is selected as 4.0, 6.0 and 8.0. For the LQC problem, the maximal generation number T is selected as 1 000 and 1 500 for the control steps, for which $N=15$ and 20 respectively. For the PCC problem, the maximal

generation number T is selected as 175 and 350 for the control steps, for which $N=10$ and 20 respectively.

6.3 Simulation example II: The conventional and SBAC methods with TVM

In this example, we applied the proposed TVM operator in the ES consisting of the conventional crossover method as well as in the ES consisting of SBAC for solving both problems. The intuitive reason behind this was to demonstrate the effectiveness of combined efforts of the proposed SBAC and TVM operators against the conventional method.

The parameter γ for TVM is kept fixed as 8.0 for both the problems. For the LQC problem, the maximal generation number T is selected as 300 and 1 500 for the control steps, for which $N=5$ and 20 respectively. For the PCC problem, the maximal generation number T is selected as 250 and 350 for the control steps, for which $N=15$ and 20 respectively.

7. Results

The results presented are the outcome of averaging results of ten independent runs with different sample paths. The ordinates in the evolution histories displayed in logarithmic scale (Figs. 4 - 11) indicate

Fig. 4 The evolution histories of the LQC problem with control steps, $N=15$

Fig. 6 The evolution histories of the PCC problem with control steps, $N=10$

Fig. 5 The evolution histories of the LQC problem with control steps, $N=20$

Fig. 7 The evolution histories of the PCC problem with control steps, $N=20$

the “Best Fitness Error”, which is defined for the LQC problem as

$$\text{Best Fitness Error} = \text{BestFitness}_i - \text{ExactSolution} \quad (17)$$

and that for the PCC problem is defined as

$$\text{Best Fitness Error} = \text{ExactSolution} - \text{BestFitness}_i \quad (18)$$

where BestFitness_i denotes the best fitness (cost) value found at generation i .

For most optimization problems, the time necessary for an algorithm to converge to the optimum depends upon the number of decision variables. However, the proposed methods cause convergence reasonably faster with increasing control steps for both problems (Figs. 4 - 11). The best results for exploration and exploitation were found from the new ES when the SBAC and TVM operators are used together. The detailed results are discussed in the following subsections.

7.1 Results for example I

The results in Figs. 4 - 7 indicate that when $\sigma=1.0$ and 0.01 the process cannot converge within the specified generations, but when $\sigma=0.1$, it causes convergence with poor results, which is undesirable for

problems demanding high precision. From these results, it is apparent that, although the SBAC method explores the search space with different directivity, the UM operator cannot exploit fast convergence. In contrast, the TVM operator causes convergence successfully with very accurate results within specified generations. The dilemma of selecting the proper σ value is resolved by the introduction of the TVM operator in the ES.

With increasing control steps (decision variables), the behavior of the TVM operator is almost the same and it causes very global searching at the initial stages and very local searching at the final stages. The evolution histories (Figs. 4 - 7) indicate that the effects on the selection of the value of γ are obvious. The degree of dependence on evolution is reduced (i.e., fast exploitation is achieved) as the value of γ increases in the TVM operator. The solution domain of the LQC problem is large (e.g., $J_1^* = 158771.813820$ for $N=15$) and that for the PCC problem is small (e.g., $J_2^* = 0.154375$ for $N=20$). The results indicate that the selection of the γ value does not depend on the solution domain or on the optimization mode (maximization / minimization), and the

Fig. 8 The evolution histories of the LQC problem with control steps, $N=5$ and $\gamma=8.0$

Fig. 10 The evolution histories of the PCC problem with control steps, $N=15$ and $\gamma=8.0$

Fig. 9 The evolution histories of the LQC problem with control steps, $N=20$ and $\gamma=8.0$

Fig. 11 The evolution histories of the PCC problem with control steps, $N=20$ and $\gamma=8.0$

suitable γ value is 8.0 for this class of problems.

7.2 Results for example II

The results in Figs. 8 - 11 show that the SBAC method along with the TVM operator causes convergence faster for both test problems. If we analyzed the results obtained (Figs. 8 - 11), it can be inferred that the subgrouping effects of the SBAC method are clearly pronounced over the conventional crossover method. An interesting behavior was found (Fig. 8) in solving the LQC problem with the control steps $N=5$ in that the conventional method failed to converge to the optimal point. Thus, capacity of the conventional crossover method to explore the search space is poor due to the stochastic error in sampling. It is evident that only the TVM operator is not successful and makes no progress unless the SBAC is used. On the other hand, the SBAC method explores the search space in various directions depending on the subgroup number to the optimum point, and the TVM helps exploit fast (but not premature) convergence. The best results are found in exploring and exploiting the evolutionary process when the SBAC and TVM operators are used together.

8. Concluding Remarks

In this paper, a new ES consisting of the subpopulation-based arithmetical crossover (SBAC) and time-variant mutation (TVM) operators is investigated with different optimization modes. To evaluate its performance, discrete-time optimal control problems are solved with different control steps. However, it ought to be stated that the proposed operators in the ES algorithm do not demonstrate that the proposed ES is better than any other evolutionary algorithm, except for the specific problems investigated, because we did not make any comparison with existing evolutionary algorithms. The simulation results indicate that the proposed operators in the ES can outperform the conventional ESs with respect to both convergence and accuracy of the solution. An optimal compromise was found between exploration and exploitation in the evolutionary process by the introduction of the proposed operators in the ES. Thus, the inherent strength of the ES algorithm is increased substantially.

References

- (1) Schwefel, H.-P., Rudolph, G. and Bäck, T., Contemporary Evolution Strategies, In *Advances in Artificial Life*, Third International Conference on Artificial Life, Vol. 929 of Lecture Notes in Artificial Intelligence, (1995), p. 893-907, Springer-Verlag, Berlin.
- (2) Schwefel, H.-P., On the Evolution of Evolutionary Computation, In *Computational Intelligence: Imitating Life*, (1994), p. 116-124, IEEE Press, New York.
- (3) Poli, R. and Logan, B., Evolutionary Computation Cookbook: Recipes for Designing New Algorithms, Procs. of the 2nd Online Workshop on Evolutionary Computation (WEC 2), Nagoya, Japan, (1996), p. 33-36.
- (4) Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, 3rd. rev., and extended edition (1996), Springer-Verlag, Berlin.
- (5) Goldberg, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, (1989), Addison-Wesley, Reading, Mass., U. S. A.
- (6) Falco, I.D., Balio, R.D., Cioppa, A.D. and Tarantino, E., A Comparative Analysis of Evolutionary Algorithms for Function Optimisation, Procs. of the 2nd Online Workshop on Evolutionary Computation (WEC 2), Nagoya, Japan, (1996), p. 29-32.
- (7) Rechenberg, I., Evolution Strategy, In *Computational Intelligence: Imitating Life*, (1994), p. 147-159, IEEE Press, New York.
- (8) Linkens, D.A. and Nyongesa, H.O., Genetic Algorithms for Fuzzy Control, part 1: Offline System Development and Application, IEE Proc. - Control Theory Appl., Vol. 142, No. 3 (1995), p. 161-176.
- (9) Linkens, D.A. and Nyongesa, H.O., Genetic Algorithms for Fuzzy Control, part 2: Online System Development and Application, IEE Proc. - Control Theory Appl., Vol. 142, No. 3 (1995), p. 177-185.
- (10) Izumi, K., Hashem, M.M.A. and Watanabe, K., An Evolution Strategy with Competing Subpopulations, Procs. of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97), Monterey, CA, U. S. A., (1997), p. 306-311.
- (11) Hashem, M.M.A., Watanabe, K. and Izumi, K., Evolution Strategy: A New Time-Variant Mutation for Fine Local Tuning, Procs. of the 1997 SICE International Sessions, Tokushima, Japan, (1997), p. 1099-1103.
- (12) Hashem, M.M.A., Watanabe, K. and Izumi, K., A New Evolution Strategy with the Elite and Mean of Individuals in a Population, Saga University Faculty Report, Vol. 26, No. 1 (1997), p. 21-31.