

# Niche Evolution Strategy for Global Optimization

**B. Porter**

Department of Industrial and Manufacturing Systems  
Engineering  
The University of Hong Kong  
Pokfulam Road, Hong Kong  
bporter@hkucc.hku.hk

**F. Xue**

Department of Industrial and Manufacturing Systems  
Engineering  
The University of Hong Kong  
Pokfulam Road, Hong Kong  
fxue@hkusua.hku.hk

**Abstract**—Many real-world problems can be formulated as global optimization problems. In recent years, evolutionary computation has been successfully applied in many such practical optimization problems which are hard to solve using traditional approaches due to their analytical intractability. However, practitioners are always in need of more effective and robust evolutionary algorithms as real-world problems become increasingly complex. In this paper, a niche evolution strategy (NES) motivated by Wright's shifting balance theory is therefore proposed. This NES employs several niches which evolve simultaneously and independently. During the evolutionary process, niche extinction and regeneration accompanied by gene flow occur at intervals of several generations. Two famous benchmark optimization problems are used to test the effectiveness of the proposed NES. The test results are compared with the corresponding results obtained from traditional single-population evolutionary computation and from distributed genetic algorithms (DGAs). It is shown that the NES is more effective than both of these approaches. It is also shown that the proposed NES can be readily applied in parallel computer architectures.

## I. INTRODUCTION

Many practical problems in engineering, natural science, economics, and business can be formulated as global optimization problems whose objective functions often possess many local optima in the region of the search space of interest. The task of global optimization in these problems is to find the absolute optimum of the objective function. In general, global optimization problems are very difficult to solve exactly. In an optimization problem, it is generally required to find an  $n$ -dimensional vector,  $x \in M \in \mathcal{R}^n$ , characterising the system under consideration

such that a certain objective function  $f: M \rightarrow \mathcal{R}$  is minimized (or maximized), i.e.,

$$f(x) \rightarrow \min(\max). \quad (1)$$

The vector,  $x$ , is usually subjected to various constraints.

In real-world applications, the objective function and the constraints are often not analytically tractable (Bäck et al 1997). It is also usually very difficult to develop formal solvable models to approximate these real application problems. However, evolutionary algorithms have often been successfully applied in these situations. Evolutionary computation mimics aspects of natural evolutionary processes under the influence of Darwinian natural selection. The better an individual performs under specified conditions the greater chance for the individual to live longer and generate offspring. In this way, the population evolves gradually to the global optimum. The majority of current implementations of evolutionary algorithms descend from three strongly related but independently developed approaches, namely, Genetic Algorithms (GAs), Evolutionary Programming (EP), and Evolution Strategies (ESs) (Bäck, 1996).

In recent years, much research has been conducted on the design of powerful evolutionary algorithms. Many different ways have been developed for the implementation of recombination and mutation (Eiben et al, 1999), and adaptation is also an important characteristic of these evolutionary operators (Hinterding et al., 1997). Many approaches have also been developed in order to deal with the constraints in real applications (Michalewicz et al., 1996). In all of these researches, the effectiveness of evolutionary algorithms in their application to real-world problems has been greatly increased. In most cases, evolutionary computations are able to find good solutions in reasonable amounts of time. There is, however, usually a big increase in the time required for evolutionary algorithms to

find adequate solutions as they are applied to larger and harder problems. In such situations, the computational time can be greatly reduced by applying parallel versions of evolutionary algorithms on parallel computers with multiple processor elements. There is an increasing potential for applying such parallel and distributed evolutionary algorithms as parallel processing computers become more accessible (Tomassini, 1998).

There are three main types of parallel evolutionary algorithms (Cantú-Paz, 1997), namely, globally parallel evolutionary algorithms, coarse-grained parallel evolutionary algorithms (*island model*), and fine-grained parallel evolutionary algorithms (*diffusion model*). The globally parallel evolutionary algorithms are similar to the traditional serial panmictic evolutionary algorithms with global selection and mating except that the evaluation of individuals is distributed in the multiple processor elements. The important characteristics of the *island model* are demes (sub-populations), local selections, and migrations from one deme to another. The whole population is divided into several sub-populations which evolve simultaneously. There are migrations among the demes every several generations. In the *diffusion model*, individuals are placed on a large toroidal one- or two-dimensional grid with one individual per grid location (and ideally one individual per processor element). During the evolutionary process, fitness evaluation is performed simultaneously for all individuals, and selection, reproduction, and mating take place locally within a small neighborhood. The information concerning individuals with high fitness can spread in the whole population due to the overlap of the neighborhoods.

Most of these previous studies on parallel evolutionary computation embody parallel versions of genetic algorithms. However, in the development of evolutionary algorithms, evolution strategies have been proposed as alternatives to genetic algorithms (Bäck, 1996). It also has been shown that evolution strategies perform better than genetic algorithms in some real applications (Porter, 1998) (Porter and Xue, 2000). In this paper, a parallel evolutionary algorithm embodying evolution strategies is therefore investigated. Unlike the current parallel evolutionary algorithms (i.e., either the *island model* or the *diffusion model*), the proposed niche evolution strategy (NES) is designed to mimic Wright's shifting balance theory (Wright, 1931). The NES contains several niches (sub-populations) which evolve simultaneously, and provides chances for trials of the search space in the evolutionary process by niche competition and gene flow among niches. By the continuous exploration of the search space, the NES is better able to find the global optima of optimization problems.

The rest of the paper is organized as follows. Section II introduces the methodology of the proposed NES based on Wright's shifting balance theory. In section III, the implementation of the NES is described. Some

computational examples and the corresponding results are given in section IV. The conclusions are presented in section V.

## II. METHODOLOGY

The character of an organism in nature is not the phenotype of a single gene, but results from the interaction of many genes. This is expressed by Wright (1978) as follows:

"I recognized that an organism must never be looked upon as a mere mosaic of 'unit characters', each determined by a single gene, but rather as a vast network of interaction systems."

If a fitness value is attributed to each set of gene combinations, the "surface" of fitness values will usually have many peaks, each separated from its neighbor by saddles. In such a huge space of gene combinations, the populations in nature can occupy only a small part of the space. In this situation, evolution employing only a single pool of population with mass selection is very slow.

The reason for this is that evolution depends too much on the new mutants, which must be favorable to some extent on their first occurrences in order to survive the competition in the big population. In nature, there is little chance that a completely new mutant is favorable; and there is even less chance that such a mutant can be selected for the next generation when competing with other individuals (Wright, 1939). Thus, according to Wright, an effective way for evolution to occur is by the simultaneous evolution of many semi-isolated local populations of a relatively large species with cross breeding. Such a form of evolution provides an effective mechanism for trials of gene combinations that lead continuously to higher peaks of fitness values.

The corresponding evolutionary algorithm applied to practical application problems is in just the same situation. In complex real-world problems, there are often many variables rather than only a single variable. Thus, the search space for such optimization problems is also very large. In comparison with such a huge search space, the population size normally used in evolutionary algorithms is very small. Therefore, it is also very important to provide individuals with the chance to explore the whole region of the search space in complex problems. Evolutionary algorithms designed to solve these complex problems should accordingly take into consideration these properties of real-world problems.

The proposed NES is designed to drive the local population to its adaptive peak of fitness value and also to provide individuals with chances to explore new regions of the search space as well. Several important characteristics of the NES mimic evolution in nature. In the proposed NES,

the whole population consists of several niches (i.e., sub-populations) which evolve simultaneously and independently. Then, at intervals of several generations, the worst niche dies. This extinct niche is replaced by new individuals which are produced by inter-niche combinations. Such a procedure guides the population to explore new regions of the search space, and thus continuously drives the whole population to higher levels of fitness.

### III. IMPLEMENTATION OF NICHE EVOLUTION STRATEGY

There are two main components of the proposed NES. These are the independent evolution of each niche and the gene flow among niches. The latter is considered as a big-step mutation that provides chances to explore new regions of the search space.

#### 3.1. Evolution of Each Niche

The evolution of each niche is like the  $(\mu + \lambda)$  evolution strategy without recombination (Porter, 1998) applied to a single pool of population. The procedure can be described by the following pseudo code:

```

NicheEvolution(i, t)
  Niche'(i, t) = Mutate[Niche(i, t)];
  Evaluate[Niche'(i, t)];
  Niche(i, t + 1) = Select[Niche'(i, t)
    ∪ Niche(i, t)];
End

```

In this algorithm,  $Niche(i, t)$  denotes the population of size  $\mu$  of niche  $i$  at generation  $t$ ;  $Niche'(i, t)$  denotes the offspring population of size  $\lambda$  produced by  $Niche(i, t)$  via mutation;  $Niche(i, t + 1)$  denotes the population of size  $\mu$  at generation  $t + 1$ .

In the evolution within niches, each individual in the niche produces offspring via mutation only. The offspring individuals in the niche are evaluated by calculating the fitness for each of the solutions,  $x$ , represented by individuals in  $Niche'(i, t)$ . This fitness value indicates the survival ability of an individual. Then, the fittest  $\mu$  individuals are selected from both the  $\mu$  parents and the  $\lambda$  offspring as the  $\mu$  parents in the next generation. This selection intensifies the search within each niche to drive the local population to an adaptive peak of fitness value.

#### Mutation operator

The evolutionary algorithm within niches uses only mutation as its operator. The mutation for each individual within a niche is like the mutation for the self-adaptive evolution strategy used by Bäck (1997). The individuals consist of a system variable vector,  $x$ , and a strategy parameter vector,  $\sigma$ . The evolutionary operators are applied to both the system variables and the strategy parameters at the same time. In general, an individual  $v = (x, \sigma)$  consists of the vectors  $x \in \mathbb{R}^n$  and  $\sigma \in \mathbb{R}_+^n$ . The mutation operator works by adding a normally distributed random vector,  $z \in \mathbb{R}^n$ , with  $z_i \sim N(0, \sigma_i^2)$  (i.e., the components of  $z$  are normally distributed with expectation of zero and variance of  $\sigma_i^2$ ). The mutation is defined by the formulas

$$\sigma'_i = \sigma_i \cdot \exp[\tau_1 \cdot N(0, 1) + \tau_2 \cdot N_i(0, 1)] \quad (2a)$$

and

$$x'_i = x_i + \sigma'_i \cdot N_i(0, 1), \quad (2b)$$

where  $\tau_1 = f_1 \cdot (\sqrt{2n})^{-1}$  and  $\tau_2 = f_2 \cdot (\sqrt{2\sqrt{n}})^{-1}$ .

#### Parameter setting

Thus, for each individual, the initial value of the step-size,  $\sigma(t = 0)$ , and the factors,  $f_1$  and  $f_2$ , must be set before the evolutionary process begins. These parameters comprise the individual's environment, which has an important impact on the evolution of the individual. In this algorithm, the initial values of these parameters for each individual are randomly selected from pre-specified intervals. Thus, both the system variables and the corresponding environment are subjected to the evolutionary process. The individuals occupying suitable environments will produce offspring with high fitness, and this suitable environment is therefore kept in the evolutionary process. This approach to parameter setting evidently saves manual adjustments of the operator parameters.

#### 3.2. Gene Flow among Niches

The gene flow among niches is another very important component of the NES. The evolution of each niche drives the local population to its adaptive peak of fitness value. But there may, of course, be some higher peaks of fitness value in the search space. Because the population size is so small compared with the huge search space, the evolutionary process needs to guide the population to some higher peaks of fitness value. The inter-niche breeding in the proposed NES, combined with niche extinction and regeneration, acts in such a role so as to allow the population to explore new regions in the search space.

In the NES, there are extinctions and regenerations at the niche level besides the parallel evolution of the local niches at the individual level. At intervals of several generations, the niche whose average fitness is the lowest among the niches dies. Some new individuals are produced to replace this extinct niche. The regeneration of the new niche is not the same as its initialization in which the system variables and the strategy parameters of each individual are randomly initialized. In niche regeneration, both the system variables of the individuals and their corresponding strategy parameters are produced by certain combination of the parental individuals from different niches.

In this inter-niche breeding, two niches are firstly selected at random. The fittest individuals from each of these niches are then selected as parents. Then, a new individual is produced by a combination of these two parental individuals and the newly generated offspring becomes one of the individuals of the new niche to replace the extinct one. This process is repeated until all of the individuals are produced for the new niche. The inter-niche combination can use any combination method which is appropriate to the given optimization problem and its corresponding representation. The current algorithm applies linear combinations both to the system variables and to the strategy parameters. In this way, each child is produced by linear combination of the parents such that

$$x_c = (1-a)x_{p1} + ax_{p2} \quad (3a)$$

and

$$\sigma_c = (1-a)\sigma_{p1} + a\sigma_{p2}, \quad (3b)$$

where  $x_c$  and  $\sigma_c$  denote the newly produced system variable vector and the strategy parameter vector, respectively. In addition,  $x_{p1}$ ,  $x_{p2}$  and  $\sigma_{p1}$ ,  $\sigma_{p2}$  denote the parental system variable vectors and strategy parameter vectors, respectively; and  $a$  is a weighting coefficient. In the current algorithm,  $a$  is set equal to  $1/2$ .

Such inter-niche combinations, as previously mentioned, serve as big-step mutations to produce individuals that explore new regions in the search space. The parental individuals for breeding are elite selected from different niches. When the parental individuals are very different from each other, their offspring generated by the combination are supposed to constitute new promising trials in the search space. When the parental individuals are similar, their offspring generated by the combination are supposed also to be good solutions. These newly generated individuals do not need to compete with other individuals on their first occurrences. Thus, these individuals provide trials of new search regions. The new niche consisting of these individuals is expected to be able to evolve to a higher adaptive peak of fitness value in subsequent evolution.

In this way, population diversity can be maintained and new regions in the search space can be well explored. The continuing explorations of new adaptive peaks of fitness value drive the whole population to evolve step by step to the global optimum.

### 3.3. Core Algorithm of NES

By combining its two main components, the general procedure of the NES can be described by the following pseudo code:

```

t = 0;
InitializeNiche(i) ∀ i ∈ N;
InitializeParameters();
Evaluate[Niche(i,0)] ∀ i ∈ N;
While(Not End)
    NicheEvolution(i,t) ∀ i ∈ N;
    At Certain Interval
        NicheCompete()
        GeneFlow()
    t = t + 1;
End

```

In this procedure,  $t$  is the generation number and  $N$  denotes the number of niches in the algorithm. The procedure of *NicheEvolution(i,t)* performs the independent evolution of each niche. In *NicheCompete()*, the worst niche dies; and *GeneFlow()* creates a new niche to replace the extinct one.

In the NES, it is also necessary to set the gene flow period (*GFP*). This parameter controls the frequencies of niche regeneration and gene flow. In the proposed algorithm, this parameter is again selected randomly from a pre-specified interval and then varies during the evolutionary process.

## IV. NUMERICAL EXAMPLES

In order to validate the proposed niche evolution strategy (NES), two famous benchmark functions are used as test cases. One is Ackley's function, and the other is Rosenbrock's function: Ackley's function has a lot of local optima, while Rosenbrock's function involves the interplay of all of the variables. Thus, both of these functions are good test cases for the use of evolutionary computation in global optimization problems.

#### 4.1. Ackley's Function

The benchmark Ackley function (Bäck, 1993),  $f$ , is defined by the formula

$$f = -20 \exp \left[ -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left[ \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + 20 + e, \quad (4)$$

where  $n$  is the dimension of the problem. This is the same function as that used by Bäck and Schwefel (1993), and the tests in the current investigation also have the same number of variables,  $n = 30$ , as used by Bäck and Schwefel (1993). The variables are randomly initialized such that  $-30 \leq x_i \leq 30$  ( $i = 1, 2, \dots, n$ ). The theoretical minimum for this function is  $f_{\min} = 0$  with all of the variables equal to 0. In this case, the fitness,  $\Phi$ , of an individual is defined by the equation

$$\Phi = F - f, \quad (5)$$

where  $F \in \mathbb{R}^+$  is an appropriately large number.

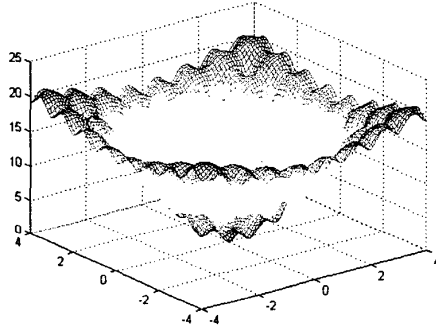


Figure 1: Two-dimensional landscape of Ackley's function

In Figure 1, the two-dimensional version of Ackley's function is plotted. The difficulties of this test case lie in its non-linearity, its involvement of quite a large set of variables, and its numerous multimodals. Thus, this test can be used to study the computational efforts required for evolutionary algorithms to solve optimization problems (Schwefel, 1995).

The NES was applied to this optimization problem with the parameters indicated in Table 1, and 10 independent trials were performed. The best function value and the mean function value obtained in each trial are listed in Table 2.

Niche Amount	10
$(\mu + \lambda)$	(20+20)
$\sigma_i(t=0)$	Random(1,6)
$f_1, f_2$	Random(1,2)
Gene Flow Period (GFP)	Random(2,10)

Table 1: The parameter setting of NES for the test cases

Best Value	Mean Value
2.6581E-09	3.1988E-09
1.1099E-10	1.7943E-10
1.1055E-10	1.6053E-10
1.6901E-11	2.3650E-11
7.8626E-12	1.1034E-11
1.3642E-10	1.9069E-10
1.1017E-11	1.4256E-11
5.1934E-11	7.3289E-11
5.2008E-11	6.9136E-11
3.9592E-11	5.3126E-11

Table 2: Best value and mean value of Ackley's function obtained using NES in each of the 10 trials

#### 4.2. Rosenbrock's Function

The benchmark Rosenbrock function (Whitley et al., 1995),  $f$ , is defined by the formula

$$f = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2], \quad (6)$$

where  $n$  is the dimension of the problem. The current tests have 30 variables, i.e.,  $n = 30$ . The variables are randomly initialized such that  $-30 \leq x_i \leq 30$  ( $i = 1, 2, \dots, n$ ). The theoretical minimum for this function is  $f_{\min} = 0$  with all of the variables equal to 1. In this case, the fitness,  $\Phi$ , of an individual is again defined by the equation

$$\Phi = F - f, \quad (7)$$

where  $F \in \mathbb{R}^+$  is an appropriately large number.

In Figure 2, the two-dimensional version of Rosenbrock's function is plotted. It is much harder to obtain the global optimum of Rosenbrock's function than of Ackley's function, even though the latter has many local optima. In Ackley's function, all of the variables contribute to the function value independently, i.e., the approach of any variable to its ultimate value can contribute to the increase in fitness value of the individual. However, the situation is

different with Rosenbrock's function whose value depends on the interplay of all the variables. In Rosenbrock's function, the approach of any single variable to its ultimate value does not necessarily improve the individual's fitness. This is just like the organisms of nature, in which networks of interacting systems determine the characteristics of the organism.

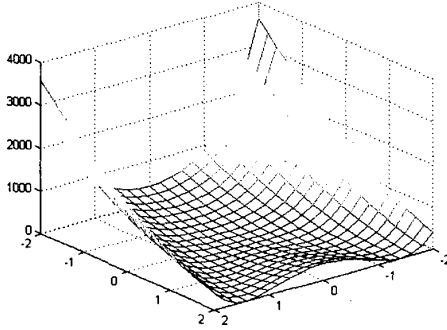


Figure 2: Two-dimensional landscape of Rosenbrock's function

The NES was applied to this optimization problem with the parameters shown in Table 1, except that the number of niches is increased to 20 for this harder problem. As in the previous case, 10 independent trials were performed and the corresponding best function value and the mean function value obtained in each trial are listed in Table 3.

Best Value	Mean Value
2.3495E-04	2.4005E-04
5.9500E-05	6.2308E-05
1.0374E-04	1.0584E-04
1.1350E-05	1.1863E-05
1.1935E-04	1.2254E-04
4.2221E-05	4.3189E-05
1.4270E-04	1.4512E-04
4.0182E-05	4.1213E-05
2.5065E-05	2.5957E-05
1.8329E-05	1.8762E-05

Table 3: Best value and mean value of Rosenbrock's function obtained using NES in each of the 10 trials

#### 4.3. Discussion

The computational results show that the proposed niche evolutionary algorithm can usually obtain the global optima

in these hard numerical benchmark functions with high accuracy.

McDonnell (1995) also used the same Ackley function in a study of recombination in evolutionary search. Indeed, McDonnell conducted an empirical study of the merits of evolution strategies incorporating different modes of recombination, namely, the MM mode (mutation only for both system variables and strategy variables), the DI mode (with, additionally, discrete recombination for system variables and intermediate recombination for strategy variables), the ID mode (with, additionally, intermediate recombination for system variables and discrete recombination for strategy variables), and the II mode (with, additionally, intermediate recombination for both system variables and strategy variables). By incorporating intermediate recombination for both system variables and strategy variables in the standard self-adaptive ES, McDonnell obtained a better approximation to the theoretical results than by using only the standard self-adaptive ES. The average best function values obtained by McDonnell using the evolution strategy with recombination mode II are about 0.3 over 50 runs, which is better than the average best function values (greater than 1.0) obtained using other forms of evolution strategy. The proposed NES is even more effective in that it usually produces very accurate results at a reasonable cost of function evaluations: in fact, the mean best value obtained by the NES shown in Table 2 is 3.20e-10.

In another paper, Yang and Kao (1996) showed that they can produce very accurate results for this Ackley function by using a deliberately devised Combined Evolutionary Algorithm (CEA). This CEA, however, is rather too complex for routine application in that it combines three phases of evolutionary processes.

It is very hard for traditional evolutionary algorithms to obtain the global optimum of Rosenbrock's function. Hiroyasu (1999) points out that even the Distributed Genetic Algorithm (DGA) with local populations cannot find the global optimum of Rosenbrock's function. The best function values that Hiroyasu obtained using the DGA for the same Rosenbrock's function as that used in this paper are usually above 0.5. However, the current test results given in Table 3 show that the mean best value obtained by the NES is 7.97e-5, thus indicating that the proposed NES provides an effective way to solve such hard optimization problems.

## V. CONCLUSIONS

In this paper, a parallel evolutionary algorithm embodying evolution strategies has been investigated. This niche evolution strategy (NES) is based on Wright's shifting balance theory of natural evolution. Instead of using only a single population of large size, the NES contains several

local populations of small size. Each niche evolves independently as in the traditional single-population evolution strategy. Simultaneously, niche competition and gene flow among niches occur every several generations. These two components of the NES guide the population continuously to explore progressively higher peaks of fitness value in the search space until the global optimum is found. The test results of the two illustrative examples show that the proposed NES is very effective in solving difficult global optimization problems.

It should also be noted that this NES could be implemented conveniently in parallel computer architectures at the level of population. Thus, the proposed algorithm can be readily applied in parallel computers with the number of processor elements equal to or greater than the number of niches in the algorithm. The results in the current investigation make it possible for the NES to get *superlinear* speedup (i.e., to obtain more than an  $N$ -fold acceleration with  $N$  processors with respect to the uniprocessor time) when applied in parallel computer architectures.

## Acknowledgements

The partial support of the Hong Kong Research Grants Council (RGC) under grant HKU 7079/98E is gratefully acknowledged.

## References

- Bäck, T. and Schwefel, H.-P., "An overview of evolutionary algorithms for parameter optimization", *Evolutionary Computation*, vol. 1, no. 1, pp1-23, 1993.
- Bäck, T., "Evolutionary Algorithms in Theory and Practice", Oxford University Press, New York, 1996.
- Bäck, T., Hammel, U., and Schwefel, H.-P., "Evolutionary computation: comments on the history and current state", *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp3-17, 1997.
- Cantú-Paz, E., "A survey of parallel genetic algorithms", University of Illinois at Urbana-Champaign Technical Report, No. 97003, 1997.
- Eiben, A.E., Hinterding, R., and Michalewicz, Z., "Parameter control in evolutionary algorithms", *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 2, pp124-141, 1999.
- Hinterding, R., Michalewicz, Z., and Eiben, A.E., "Adaptation in evolutionary computation: a survey", *IEEE International Conference in Evolutionary Computation*, pp65-69, 1997.
- Hiroyasu, T., Miki, M., and Negami, M., "Distributed genetic algorithms with randomized migration rate", *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, pp689-694, 1999.
- McDonnell, J.R. and Waagen, D.E., "An empirical study of recombination in evolutionary search", *Proc. Fourth Annual Conference on Evolutionary Programming*, pp465-478, 1995.
- Michalewicz, Z. and Schoenauer, M., "Evolutionary algorithms for constrained parameter optimization problems", *Evolutionary Computation*, vol. 4, no. 1, pp1-32, 1996.
- Porter, B., "Evolutionary production planning: a cautionary tale", *CIRP J. Manufacturing Systems*, vol. 27, pp115-120, 1998.
- Porter, B. and Xue, F., "Evolutionary computation of geodesics on three-dimensional curved surfaces for CAD/CAM", *Proc. Ninth International Manufacturing Conference in China*, vol. 2, pp61-62, 2000.
- Schwefel, H.-P., "Evolution and Optimum Seeking", Wiley, 1995.
- Tomassini, M., "Parallel and distributed evolutionary algorithms: a review", in "Evolutionary Algorithms in Engineering and Computer Science", Miettinen, K., Neittaanmäki, P., Mäkelä, M.M., and Périaux eds., 1998.
- Whitley, D., Mathias, K., Rana, S. and Dzuber, J., "Building better test functions", *Proc. of the Sixth International Conference on Genetic Algorithms*, pp239-247, 1995.
- Wright, S., "Evolution in mendelian populations", *Genetics*, vol. 16, pp97-159, 1931.
- Wright, S., "The relation of livestock breeding to theories of evolution", *Journal of Animal Science*, vol. 46, pp1192-1200, 1978.
- Wright, S., "Statistical genetics in relation to evolution", *Actualités scientifiques et industrielles, 802: Exposés de Biométrie et de la statistique biologique XIII*, 1939.
- Yang, J.M. and Kao, C.Y., "A combined evolutionary algorithm for real parameters optimization", *Proc. IEEE International Conference on Evolutionary Computation*, pp732-737, 1996.