

# 琉球大学学術リポジトリ

## A Hybrid Optimization Method Using a Real-coded Multi-parent GA, Simplex & Simulated Annealing with Applications in Resolution of Overlapped Signals

メタデータ	言語: 出版者: 琉球大学工学部 公開日: 2007-09-16 キーワード (Ja): キーワード (En): Genetic Algorithms, Simplex, Simulated Annealing, Hybrid Algorithms, Overlapping Signals 作成者: Mendoza, Noel, Chen, Yen-Wei, Nakao, Zensho, Adachi, Tatsuhiro, Masuda, Yoshihisa, 陳, 延偉, 仲尾, 善勝 メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/20.500.12000/1958">http://hdl.handle.net/20.500.12000/1958</a>

# A Hybrid Optimization Method Using a Real-coded Multi-parent GA, Simplex & Simulated Annealing with Applications in Resolution of Overlapped Signals<sup>+</sup>

Noel Mendoza\*, Yen Wei Chen\*, Zensho Nakao\*, Tatsuhiko Adachi\*\*, & Yoshihisa Masuda\*\*.

Email: {nmendoza, chen, nakao}@augusta.eee.u-ryukyu.ac.jp

## Abstract

This paper presents a GA algorithm for function optimization that combines the features of Simplex crossover and Simulated Annealing. Hybridization with a local search tool like Simplex directs the GA to more promising search spaces while preventing premature local minima convergence. The use of Simulated Annealing(SA) as a mutation operator further enhances the GA's overall capability. This elitist real-coded multi-parent model is able to address the issue of slow convergence rate very well and is robust in optimizing epistatic multimodal problems. In addition to test functions, our model was applied to the resolution of overlapping signals.

**Keywords:** Genetic Algorithms, Simplex, Simulated Annealing, Hybrid Algorithms, Overlapping Signals

## 1 Introduction

Genetic algorithms (GA) occupy a growing niche in the area of search and problem optimization, having wide applications in many fields specifically engineering and science<sup>1</sup>. Since Holland's first GA<sup>2</sup>, GA's of numerous forms and classes have been proposed and research has continued with unending fervor.

One of the principal obstacles in GA research has to do with GA's slow convergence rate. Enormous initial population, complex selection, crossover and mutation operator requirements do drive computation costs above acceptable limits. GA's convergence rate is slower than most direct methods but have the advantage of being more "globally oriented". Direct search methods like Gradient Search<sup>30,31</sup>, Simplex<sup>3</sup> and Simulated Annealing<sup>4,32,33</sup>, though efficient fine tuners, concentrate on local information, thereby increasing the risk of premature local minima convergence.

Toward amending GA's convergence problem, there have been several efforts at combining GA with local search algorithms<sup>1,5,6</sup> in the hope that resulting the hybrid system can explore a better tradeoff between computational cost and the global optimality of the

solution found. Another way of addressing the convergence dilemma is to work on improving the operators or representation<sup>7,8</sup>.

It has been reported that GAs utilizing only binary and gray representations can not obtain accurate solutions compared to other techniques. Although fairly slower computation-wise, real-coded vectors offer better performance and have, over the years been studied and shown to outperform traditional bit string operations<sup>7,9,10,11,12,13,28</sup>.

Furthermore, the No Free Lunch Theorem, dismissed any hope of finding better bit representation, by establishing that all search algorithms possess the same performance over the set of all functions<sup>18</sup>.

Of real-coded GAs, Eschelmann's real-coded BLX- $\alpha$ <sup>12</sup> has been noted to get relatively good results. However, it behaves poorly on epistatic or nonseparable functions<sup>19</sup>. This is attributed to the fact that with real-coded GA's, offspring may not get to inherit parental characteristics properly. The design of algorithms whereby children suitably inherit useful parental characteristics is of paramount importance<sup>20</sup>.

Ono's Unimodal Normal Distribution(UND) crossover operator which employs  $N>2$  parents, preserves parental characteristics better and moves free of the coordinate system. In this case, offspring are generated using a normal distribution defined by 3 parents. UND has been shown to outperform the BLX- $\alpha$ <sup>14</sup>. Other multi-parent operators like Center of Mass Crossover (CMX)<sup>15</sup>, Multi-parent Feature-wise Crossover(MFX), Seed Crossover(SX)<sup>16</sup>, and Simplex-based Crossover(SPX)<sup>17</sup> have been developed and shown to do better. In this paper, we will also present a real multi-parent crossover

受理：1999年12月6日

\*本論文は平成11年度電気学会・電子情報通信学会合同講演会にて発表済み

\*工学部電気電子工学科

(Department of Electrical and Electronics Engineering, Faculty of Engineering, University of the Ryukyus)

\*\*郵政省通信総合研究所

(Communications Research Laboratory, Ministry of Posts & Telecommunications)

called **Multi-parent Uniform Arithmetic Crossover(MAC)**, its operations will be discussed in the next section.

Multi-parent algorithms as above, are not new and several studies depicting their relative advantages over traditional 2 parent recombination have been done<sup>21</sup>. In the case of Eiben and Bäch<sup>22</sup>, it was observed that though the performance of algorithm depended on the particular combination of operators and objective functions, there was a substantial gain as the number of parents increased.

Toward the objective of improving the convergence rate and efficiency of GA, we introduce a hybrid approach that combines a real-coded multi-parent GA with a concurrent simplified variant of Yen’s probabilistic simplex<sup>1</sup>(PS). Simulated Annealing is likewise used in place of traditional mutation operators.

Although our basic architecture bears close resemblance to that proposed by Yen et al, it differs in the type of operators used(Table1). As experiment results show, the choice of operator contribute significantly to the hybrid’s overall performance. We tested our GA-simplex hybrid using SPX and SA, alongside the following, with functions commonly found in literature, and compared the results.

- Simulated Annealing<sup>4</sup> alone
- SPX<sup>17</sup> alone
- Yen’s GA-simplex hybrid<sup>1</sup>
- GA-simplex hybrid using SPX
- GA-simplex hybrid using MAC

The proposed model was also utilized in resolving overlapping signals. Please refer to succeeding sections for additional information regarding this.

Table 1 Operator Differences in Test Models

Model /Operator	SA	SPX	Yen’s GA-Simplex Hybrid	MAC-Simplex Hybrid	SPX – Simplex Hybrid	SPX-Simplex +SA
Crossover	NA	SPX	2P AC	MAC	SPX	SPX
Mutation	NA	RM	RM	RM	RM	SA
Hybrid	NA	NA	Concurrent PS	Concurrent SS	Concurrent SS	Concurrent SS
Selection	NA	Elitist	Elitist	Elitist	Elitist	Elitist
Others	None	BEM	None	BEM	BEM	BEM

The next section provides a brief description of key processes. An explanation of the employed hybrid architecture is given in Section 3. Experiment using standard GA test functions and results are presented in Section 4. Section 5 discusses the successful application

of our model in overlapped signal resolution. The summary and future plans are in Section 6.

## 2 General Concepts

### 2.1 Genetic Algorithms

Genetic Algorithms belong to a class of methods patterned after natural evolution<sup>23,7</sup>, developed to solve a wide range of optimization problems. Simply put, they operate by maintaining a set of candidate solutions<sup>2</sup> where each parameter to be optimized is encoded as a binary string or real vector called a chromosome. Crossover, mutation, and selection operators facilitate exchange of genetic information, generating new and better individuals (i.e. solutions) over time. The convergence property for string-based GA can be mathematically demonstrated using the Schema Theorem<sup>7,11</sup>.

To identify and safeguard good solutions, an objective function is utilized to evaluate each potential candidate and resulting fitness values are assigned. The cycle of evaluation, selection and reproduction continues till an acceptable solution is found or when an acceptable criterion is met.

What makes GA, a promising optimization tool is that it is inherently parallel. GA provides for several search space points to be explored in simultaneously. Furthermore, the presence of mutation operators, to a certain extent, forestalls local minima entrapment (a waterloo for most linear optimization algorithms).

As cited previously, multi-parent real GA implementations have gained tremendous following in recent years. Since the discussion of the merits of real over traditional string implementation and multiple parent over dual parent is beyond the scope of this paper, it is encouraged that the reader refer to other sources for this purpose.

### 2.2 Elitist Selection Scheme

Selection schemes influence the GA treatment of potential solutions. An Elitist Selection Scheme<sup>24</sup> is one where the best estimates (chromosomes) are, sans modifications, copied directly to the next generation. This operation guarantees that low cost/high fitness individuals are preserved, (i.e. do not disappear during the course of evolution.)

Non-elite chromosomes are selected via a roulette wheel selection process. Since the scheme simulates a roulette wheel with slot sizes proportionate to fitness values, the best chromosomes obtain more copies, average ones stay even and worst ones die off. The wheel is executed

*popsiz*-*n* times, where *n* is the number of elites.

### 2.3 Crossover Operators

Crossover operators play a critical role in the proper interchange of genetic information. Crossover combines the features of parent chromosome by swapping corresponding parental segments. It allows uniform exchange of different potential solutions.

Relevant to our experiment, we shall be discussing two real multi-parent crossover operators namely a Multi-parent Uniform Arithmetic Crossover (MAC) and Tsutsui's Simplex-based crossover (SPX).

#### 2.3.1 Multi-parent Arithmetic Crossover (MAC)

In the case of simple uniform arithmetic crossover<sup>24</sup>, an offspring is a linear combination of two parents. If *b'* and *c'* are to be crossed, the resulting offspring are  $b^{t+1} = \phi b^t + (1-\phi) c^t$  and  $c^{t+1} = \phi c^t + (1-\phi) b^t$ . Parameter  $\phi$  can have a predefined value, or can obtain its value by means of some random process.

Multi-parent Arithmetic Crossover (MAC), extends this operation to  $N > 2$  parents. Thus a random set of *N* parents is selected from the set of *M* parents selected for crossover. *N'* offspring (linear combinations of *N* parents) are produced. Therefore, given *b* to be collection of *N* parents, and *a*, a random set of coefficients adding up to unity, a child *c<sub>j</sub>* is given as:

$$c_j = \sum_{i=1}^n a_i b_i$$

#### 2.3.2 Simplex-based Crossover (SPX)

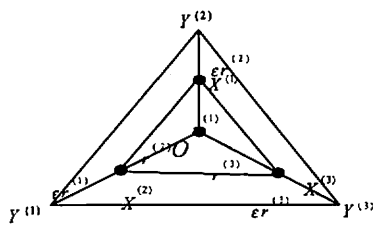


Figure 1. SPX-2-3-ε

The SPX algorithm, proposed by Tsutsui et al, works by uniformly picking vector values from an expanded simplex generated by *N* parents<sup>17</sup>. This expanded simplex is constructed the following manner: Lets consider a 3-parent SPX in a 2D search space as shown in Fig.1, where  $x^{(1)}$ ,  $x^{(2)}$ ,  $x^{(3)}$  are parameter vector of three parents. Then, these vectors form a simplex. We then amplify this simplex for each direction  $(x^{(j)} - O)$  by  $(1 + \epsilon)$  ( $\epsilon \geq 0$ ) times, where *O* is the center of mass of the 3 parents calculated as

$$Y^{(j)} = (1 + \epsilon)(x^{(j)} - O)$$

and

$$O = \frac{1}{3} \sum_{j=1}^3 x^{(j)}$$

$Y^{(j)}$  forms the expanded simplex. Three offspring can thus be produced from three parents by randomly selecting values from  $Y^{(j)}$ . The SPX method is specified as SPX-*n-m-ε*, where *n* is the number of parameters of the search space, *m* is the number of parents and  $\epsilon$  is a control parameter that defines the expanding rate. In the case above, the SPX given as SPX-2-3-ε. *M* can be less than or equal to *N*+1.

SPX is deputed to work well with functions having multimodality and epistasis. Nonetheless, convergence is slow as the MNT (i.e. mean number of function evaluations where the optimum is reached) is noticeably large, generally running to thousands<sup>17</sup>.

### 2.4 Mutation Operators

To induce some degree of variability into the population, two mutation operators namely Random Mutation and Simulated Annealing (SA) were employed. Although SA, is by itself considered as a separate optimization tool, it is presented here as a mutation operator.

#### 2.4.1 Random Mutation (RM)

Random mutation creates a new chromosome by changing the value of its gene component. The new gene's value is computed as:

$$g_j = x_j + \beta(x_h - x_l)$$

where  $x_h$  and  $x_l$  are the lower and upper limits of the parameter *x*;  $\beta$  is can be a random or predefined value between 0 and 1.

#### 2.4.2 Simulated Annealing

Simulated annealing (SA)<sup>32,33,4</sup> is an optimization tool that works well on a wide variety of practical problems. It is usually preferred over GA, especially for problems that may inherently require a large number of state transitions for a near-optimal solution to be found. In such cases, genetic search is rendered infeasible given the high cost of computing a single iteration in the enlarged state-space.

SA's ability to find the optimum depends as much on the temperature schedule and perturbation rate as on the initial sample. Since SA works on a singleton population, we attempted to use SA as a mutation operator.

Lastly if  $p_r$  is not worse than  $p_{n+1}$  but is worse than  $p_n$ ,  $p_{cw}$ , close to the centroid but on the opposite side is generated(Fig. 4).

The new points are obtained as follows. Figure 4 illustrates the position of said points.

$$P_r = p_g + \alpha(p_g - p_{n+1}), \quad \alpha > 2 \quad (1)$$

$$p_c = p_g + \beta(p_g - p_{n+1}), \quad \beta > 2 \quad (2)$$

$$p_{\alpha} = p_g + \gamma(p_g - p_{n+1}), \quad 1 < \gamma < 2 \quad (3)$$

$$p_{cw} = p_g + \delta(p_g - p_{n+1}), \quad 0 < \delta < 1 \quad (4)$$

( $\delta$ ,  $\beta$ ) and  $\gamma$  are expansion and contraction coefficients respectively.

### 2.5.2 Probabilistic Simplex Method (PS)

Yen et al, introduced a probabilistic version of N-M simplex whereby the distance between  $p_g$  and  $p_r$  is determined stochastically<sup>1</sup>. With this, the aforementioned four equations are thereby reduced to:

$$\begin{cases} p_r = p_g + \alpha(p_g - p_{n+1}) - \text{Probabilistic Expansion} \\ p_r = p_g + \beta(p_g - p_{n+1}) - \text{Probabilistic Contraction} \end{cases}$$

$\alpha$ , and  $\beta$  obtain values from intervals [0,2] and [0,1] respectively based on a predefined probability distribution. In this case, a triangular probability density function that peaks at 1 and reaches 0 probability at 0 and 2 and another that peaks at 0.5 were used respectively.

Probabilistic simplex is assumed to be more flexible as it enabled the search space to be explored unhindered.

### 2.5.3 Stochastic Simplex Method (SS)

By constraining  $\alpha$  and  $\beta$  to ranges [0,2] and [0,1] respectively, we simplified PSM even further by allowing  $\alpha$  and  $\beta$ 's values to be defined through uniform random distribution. This approach eases exploration and lets the distance between centroid and  $p_r$  be even more liberally set.

### 2.5.4 Concurrent Simplex

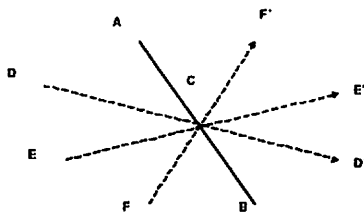


Figure 5. 2-dimensional concurrent simplex

Concurrent simplex<sup>1</sup> is a simplex variant which utilizes  $N + \Omega$  points, where  $\Omega > 1$ . In lieu of a single point,  $p_{n+1}$ ,  $p_{n+2}, \dots, p_{n+\Omega-1}, p_{n+\Omega}$  points are reflected across the centroid (computed from the best N points), to create  $p'_{n+1}, p'_{n+2}, \dots, p'_{n+\Omega-1}, p'_{n+\Omega}$ . All the points are then re-evaluated and a new set of best points ( $p'_1, p'_2, \dots, p'_{n-1}, p'_n$ ) is selected(Fig. 5) The process is then iterated for a predefined number of times.

Although, with each iteration, concurrent simplex requires  $\Omega - 1$  more point evaluations than conventional simplex, it provides for a more rigorous search of new frontiers.

### 2.6 Boundary Extension by Mirroring (BEM)

Boundary Extension by Mirroring (BEM)<sup>15</sup> was introduced by Tsutsui et al, to enable SPX, CMX and similar algorithms to suitably cover optimum situated in the corner of the search space or beyond.

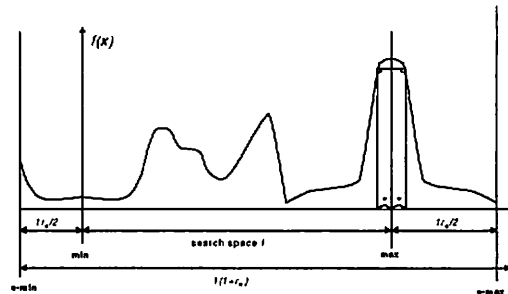


Figure 6. Boundary extension by mirroring (BEM)

Functional values of points outside the boundary are computed as though they belong inside the search space at points symmetrical to the boundary. An extension coefficient,  $r_e$  is introduced to attenuate the boundary by a factor of  $1 + r_e$  in each dimension. In our case, we set  $r_e$  to its maximum (1).

Thus, the functional value of each parameter of vector  $X^{(i)}$  is obtained as:

$$f(X^{(i)}) = f(Y^{(i)})$$

where

$$Y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \dots, y_{n-1}^{(i)}, y_n^{(i)})$$

$$y_j^{(i)} = \begin{cases} 2 \min_j - x_j^{(i)} : x_j < \min \\ 2 \max_j - x_j^{(i)} : x_j > \max \\ x_j^{(i)} : \text{otherwise} \end{cases}$$

Min and max represent the minimum and maximum values allowed for each parameter. Tsutsui et al, went on to show that while BEM improves the algorithms

The simulated annealing algorithm we employed is shown in Fig.2 .  $T$  and  $E$  represent the annealing temperature and entropy of the system at some instance  $I$ ,  $\Delta E$  is the energy gap or corresponding change in entropy resulting from perturbation  $\delta^{25}$ .

For a minimization problem, the SA provides a mechanism for uphill climbing which prevents it from being trapped in local minima. The probability for this climb is determined using the Boltzmann probability function:

$$p = e^{-\Delta E/T}$$

Since  $T$  is always decreased gradually as the annealing proceeds, the system is less likely to accept uphill moves in latter stages. For optimum control of  $T$ , we used the following exponential annealing schedule<sup>32</sup> with  $0.77 \leq \alpha \leq .99$

$$T_{i+1} = \alpha T_i$$

With each annealing cycle, a small random perturbation,  $\delta$  is added to each parameter, the value of which is defined as the product of a random variable  $q \in [-0.5, 0.5]$  and some stochastic value between  $[0, 1]$ . SA mutation is applied only once every generation.

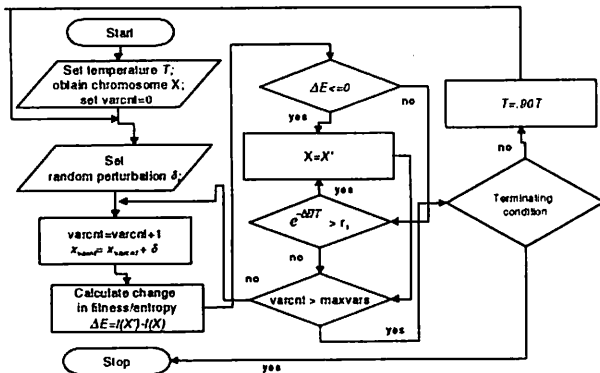


Figure 2. Simulated Annealing Flowchart

### 2.5 Simplex Method

A simplex is defined as a set of  $N+1$  independent points where  $N$  is the number of dimension of a given search space. The simplex method, developed by Spendley et al<sup>27</sup>, is a local search technique that evaluates simplices to determine promising search directions.

In the basic method, we evaluate a simplex of  $N+1$  points( $p_1, p_2, p_3, \dots, p_n, p_{n+1}$ ) to get corresponding fitness estimates  $E(p_1), E(p_2), \dots, E(p_n), E(p_{n+1})$ . We then order the points from best to worst accordingly.

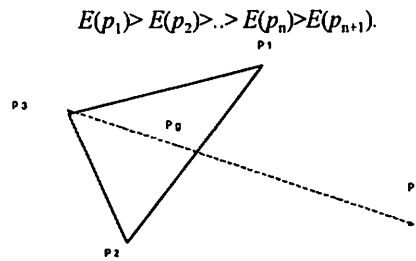


Figure 3. 2-dimensional simplex

We create new simplices by replacing the worst point, denoted by  $p_{n+1}$ , with a new point  $p_r$ , generated by reflecting  $p_{n+1}$ , over the centroid  $p_g$  of the remaining points

$$p_r = p_g + (p_g - p_{n+1})$$

where

$$p_g = \frac{1}{l} \sum_{l=1}^n p_l$$

The new simplex contains all the points of the previous one except the worst one which is replaced by  $p_r$ . The cycle is continued until the step size becomes less than a predefined value or the simplex circles around the optimum<sup>27,1</sup>. Figure 3 demonstrates the operation for 3 points.

#### 2.5.1 Nelder-Mead Simplex

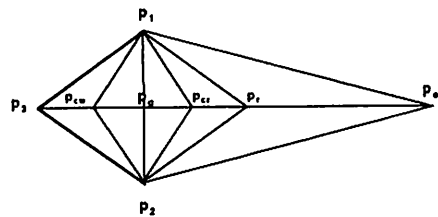


Figure 4. 2-dimensional N-M Simplex

The Nelder and Mead Simplex<sup>3,29</sup> is a modification of the original simplex method that allows the procedure to adjust its search step in accordance with the evaluation result of the new point generated. This is done through expansion, contraction and reflection.

New points  $p_r, p_e, p_{cr}, p_{cw}$  are created to replace  $p_{n+1}$ . If the new point  $p_r$  is better than the best point  $p_1$ , expanding further along the reflection direction creates point  $p_e$ . If  $p_r$  is worse than  $p_{n+1}$ , contraction produces  $p_{cr}$  which is close to the centroid on the same side as  $p_{n+1}$ .

performance when it comes to finding optimum located on the corner or beyond the search space, it does not have any detrimental effects on functions having their optima within and around the search space.

### 3 Hybrid Architecture

Several researchers in the field have endeavored juxtaposing GA and other optimization techniques under the pretext that hybridization increases the reliability and speed of GA convergence<sup>5,34,35,36</sup>.

The type and degree of hybridization are difficult to establish and depend largely on experience and testing. Too much of a local technique causes the system to carry on behavior like of that technique, i.e. becoming too locally adapted. While too less, on the other hand, makes it no better than an ordinary GA.

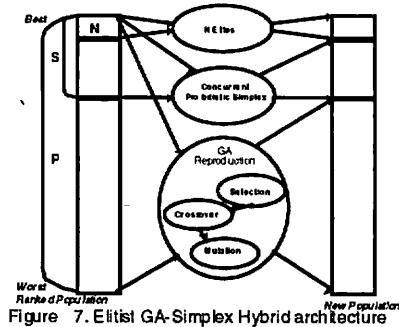


Figure 7. Elite-based GA-Simplex Hybrid architecture

Yen et al's, Elite-based GA-simplex hybrid<sup>1</sup> (Fig.7) is of particular interest to us. Per this architecture, three sets of individuals make up the new population.

Yen's GA Simplex Algorithm	Modified GA- Simplex Algorithm
(Initialize) Generate a random population of size P.	(Initialize) Generate a random population of size P.
Repeat	Repeat
<ul style="list-style-type: none"> <li>(Evaluate and Ranking) Evaluate the fitness of each chromosome. Rank them based on fitness results.</li> <li>(N Elites) Copy N elites to the next generation.</li> <li>(Simplex) Apply probabilistic simplex to the top S-N chromosomes and copy generated chromosomes to the next generation.</li> <li>(Selection) Select P-S chromosomes based on ranking of fitness and copy to the next generation.</li> <li>(Mutation) Apply mutation with the mutation probability to the P-S chromosomes.</li> <li>(Crossover) Apply 2 parent crossover with the crossover probability to the P-S chromosomes.</li> </ul>	<ul style="list-style-type: none"> <li>(Evaluate and Ranking) Evaluate the fitness of each chromosome. Rank them based on fitness results.</li> <li>(N Elites) Copy N elites to the next generation.</li> <li>(Simplex) Apply stochastic simplex to the top S chromosomes and copy the best S-N chromosomes to the next generation.</li> <li>(Selection) Select P chromosomes from old generation via roulette wheel for reproduction.</li> <li>(Crossover) Apply multi-parent crossover with the crossover probability to the P chromosomes.</li> <li>(Mutation) Apply SA mutation with the mutation probability to the P chromosomes</li> <li>(Selection) Select P-S best chromosomes from P' and transfer to next generation.</li> </ul>
Until a termination condition is set.	Until a termination condition is set.

Fig. 8 Algorithmic Differences

The first group consists top-ranking chromosomes(elites) from the previous generation that are translated without

changes to the new generation. The second set is made up of individuals resulting from a special operator applied to top members of the previous generation. Lastly is the set created through conventional GA crossover and mutation.

Yen's GA-simplex hybrid operates by applying a concurrent version of probabilistic simplex operator on top ranking chromosomes. The operator is applied to the top S-N chromosomes of the population to produce S-N children, N being the number of elites. The remaining P-S chromosomes are generated using ordinary GA reproduction.

In our proposed method, we varied the architecture slightly while retaining many of its advantageous features. Similarly, we introduced SPX and SA as GA crossover and mutation operators. The algorithms are shown in Figure 8.

Whereas the former architecture restricted operations to S-N, P-S individuals, we used the S individuals for concurrent simplex and P individuals for GA. The best S-N, and best P-S results are copied to the next generation. Coupled with SA mutation, the new population contains the so-called best results of these various operations. The population is re-evaluated, sorted and the cycle is repeated anew.

### 4 Simulation Using Test Functions

#### 4.1 Test Suite

We have relied on test functions commonly found in GA literature to evaluate our modified GA-hybrid. These functions include the well-known Dejong test suite and other notables like Rastrigin, Schwefel, Griewank, etc., which were introduced over the years<sup>26,17</sup>. These functions cover a variety of characteristics that affect algorithmic performance.

F1, more properly known as the Sphere function, is unimodal with global minimum at (0,0,0). F2 (Rosenbrock's Function), whose minimum is at (1,1), is a nonlinear function that possess strong interaction (epistasis) between parameters. F3 is a discontinuous function with a minimum function value of -30. F4 has minimum at (0,0,...,0,0) but the presence of Gaussian noise makes finding it difficult. F5 is a multi-modal function with several local optima.

F6, called Rastrigin's function, contains several local optima around the global minimum,(0,0,...,0,0). Schwefel's function(F7) is a multi-modal function with global minimum at (420.968746,...,420.968746), very close to the corner of the search space. F8 (Griewank's function) is a scalable nonlinear and

Table 2 GA Test Suite

	Function	Domain	Characteristics
F1	$f(x_{1..3}) = \sum_{i=1}^3 x_i^2$	$x_i \in [-512,511]$	unimodal with global minimum at center
F2	$f(x_{1..2}) = 100(x_1^2 - x_2) + (1 - x_1)^2$	$x_i \in [-2.048,2.047]$	highly epistatic with global minimum at center
F3	$f(x_{1..5}) = \sum_{i=1}^5 \lfloor x_i \rfloor$	$x_i \in [-512,511]$	highly discontinuous w/ global minimum at corner of search space
F4	$f(x_{1..30}) = \left[ \sum_{i=1}^{30} ix_i^4 \right] + Gauss(0,1)$	$x_i \in [-1.28,1.27]$	noisy with global minimum at center
F5	$f(x_{1..1,2}) = 0.002 + \sum_{i=1}^{30} \frac{1}{j + \sum_{n=1}^j (x_n - a_n)^4}$	$x_i \in [-65.536,65.535]$	nonseparable with global minimum bet. center and corner
F6	$f(x_{1..N}) = (N*10) + \left[ \sum_{i=1}^N x_i^2 - 10 \cos(2\pi x_i) \right]$	$x_i \in [-512,511]$	highly multimodal with global minimum at center
F7	$f(x_{1..N}) = \sum_{i=1}^N -x_i \sin(\sqrt{ x_i })$	$x_i \in [-512,511]$	multimodal with global minimum at corner
F8	$f(x_{1..N}) = 1 + \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos(x_i/\sqrt{i})$	$x_i \in [-512,511]$	multimodal with global minimum at corner
F9	$f(x_{1..2}) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$	$x_i \in [-100,100]$	epistatic, nonlinear and multimodal with global minimum at center
F10	$f(x_{1..2}) = (x_1^2 + x_2^2)^{2.2} [\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1.0]$	$x_i \in [-100,100]$	epistatic, nonlinear and multimodal with global minimum at center

nonseparable multi-modal function having minimum at 0. F9 and F10 are known as the Sine Envelope Sine wave and the Stretched V Sine wave functions.

F1, F3, F5, F6, F7 and F4 (in the absence of Gaussian noise) are separable functions. Of nonseparable (epistatic), nonlinear problems F2, F9, F8 and F10, only F8 is scalable. However, F8 is found to exhibit undesirable properties as the dimensionality of the function increased. With increased dimension, the contribution of the product term of F8 becomes smaller and the local optima induced by the cosine term decreases, the function then becomes easier to solve for numeric real valued representation<sup>26</sup>.

### 4.2 Simulation Parameters

Table 3 lists the parameters used for each model in the experiment. We performed a total of 40 runs of 500 generations each for each problem and model. A population size of 150 was used for problems F1, F2, F3, F5 while 300 was used for the rest. Except for Tsutsui's SPX, the number of parents is equal to the number of problem parameters.

With  $O_g$  and  $R$  representing theoretical global optimum and the actual optimum reached after 500 generations respectively, we compute the fitness to be

$$fitness = \frac{1}{1 + R - O_g}$$

If the theoretical global optimum is obtained, fitness equals 1. Given the fitness, we then proceeded to calculate the percent perturbation from the optimum, using:

$$diff = \frac{1 - fitness}{1} * 100\%$$

When fitness is 1, diff equals 0. Exhibit 1.0 shows the number of trials in which diff falls within a specific range. The more trials in the upper ranges, the better the model. The diff ranges are 0, (0,0.05], (0.05,0.5],(0.5,2],(2,5], and above 5.0%.

Table 3 Parameters set in experiment

	SA	Tsutsui's SPX	Yen's GA-Simplex	MAC-Simplex	SPX-Simplex	SPX-Simplex +SA
Simulated Annealing						
Initial temperature	200					200
Iterations	1000					1000
Alpha	Random					Random
Genetic Algorithm						
Population	1	150/300	150/300	150/300	150/300	150/300
Parents		2-4	Max. no. of vars.	Max. no. of vars.	Max. no. of vars.	Max. no. of vars.
Elites		1	0	0	0	0
Crossover Prob.		80%	80%	80%	80%	80%
Mutation Prob.		2%	2%	2%	2%	2%
Concurrent Simplex						
% of Population			50%	50%	50%	50%
Elites			5	5	5	5
Iterations			10	10	10	10
Simplex Type			PS	SS	SS	SS
Concurrent			.5P-5	.5P-5	.5P-5	.5P-5
Points						
Others						
Trials	40	40	40	40	40	40

### 4.3 Results of Simulation

Simulation results are presented in Fig. 9 and Table 4.



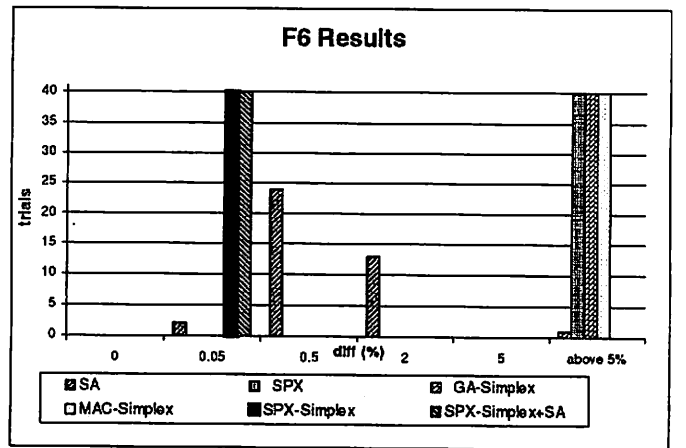
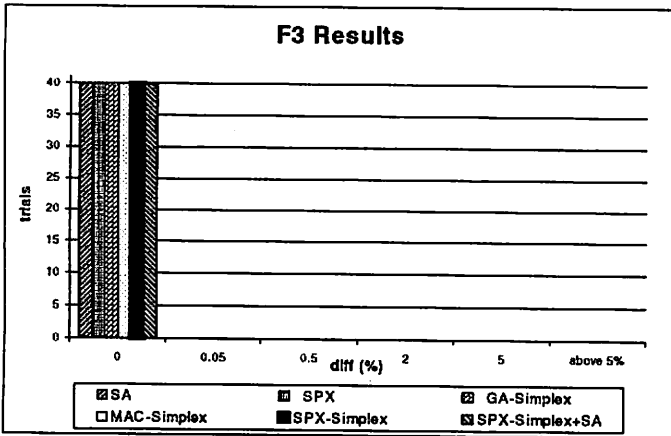
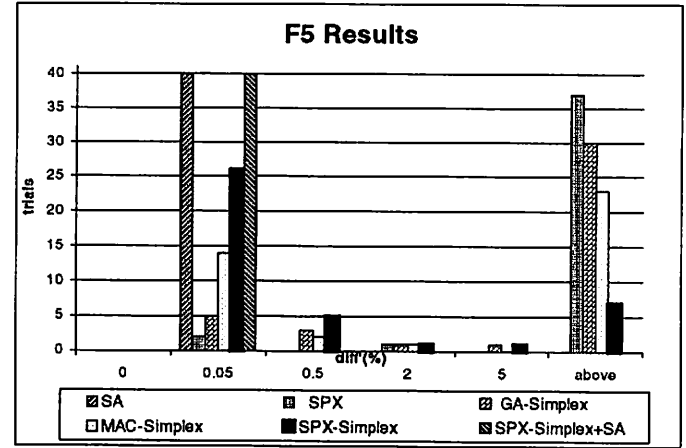
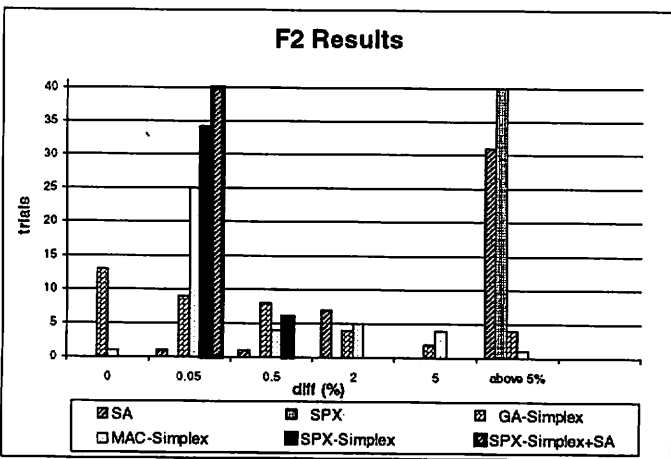
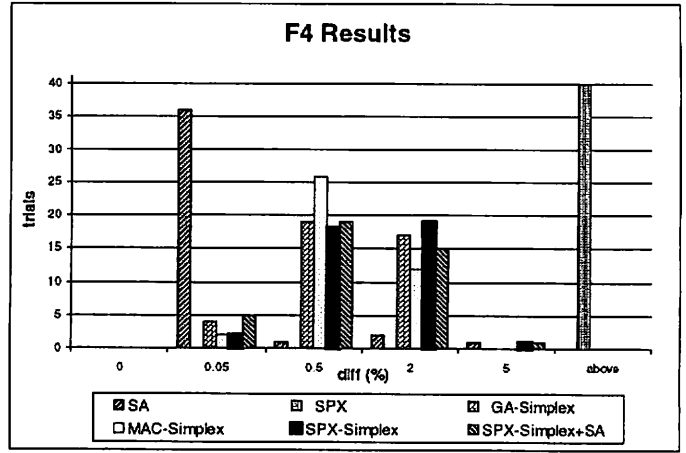
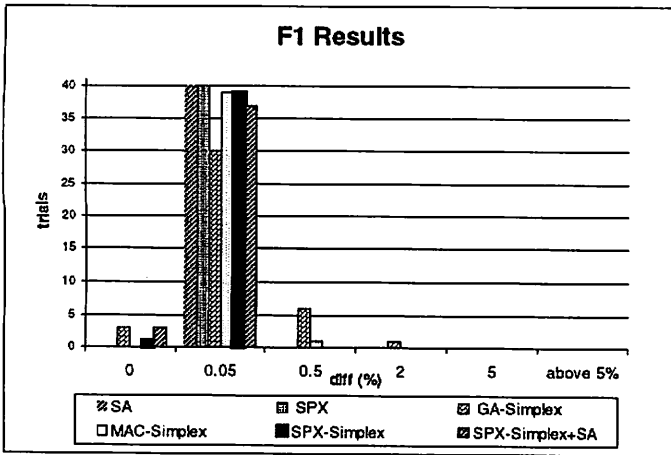


Fig 9. Test Suite Simulation Results

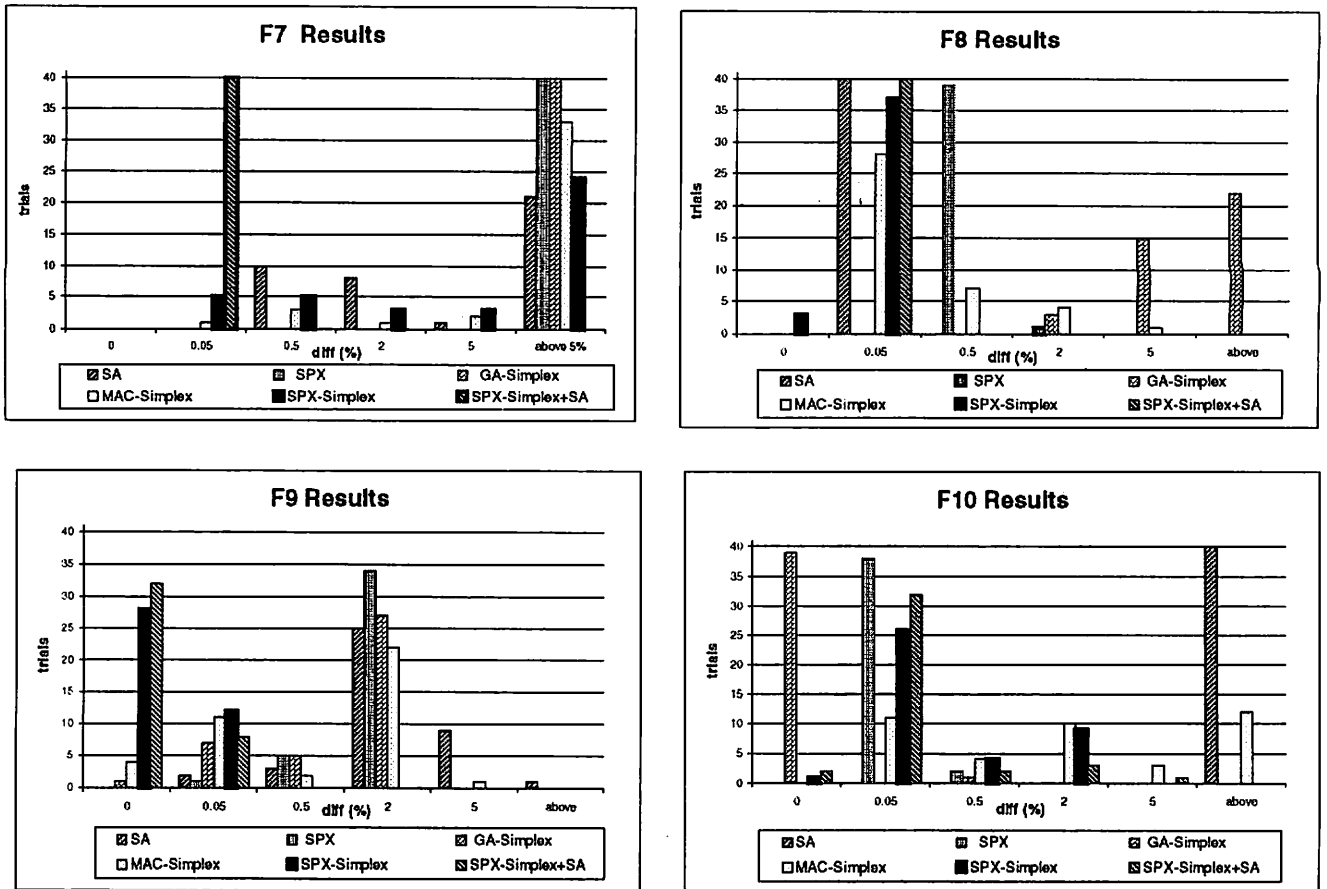


Fig 9. Test Suite Simulation Results (2)

Table 4 Simulation Results for Test Functions

		Models					
		SA	SPX	Yen's GA-Hybrid	MAC-Simplex	SPX-Simplex	SPX-Simplex-SA
F1	*Avg	0.000002	0.000004	0.000480	0.000031	0.000005	0.0
	Std	0.000003	0.000007	0.001707	0.00011	0.000015	0.0
F2	*Avg	0.1785051	0.442331	0.015494	0.00706	0.000219	0.000001
	Std	0.145382	0.014566	0.040532	0.018239	0.000364	0.000001
F3	*Avg	0.0	0.0	0.0	0.0	0.0	0.0
	Std	0.0	0.0	0.0	0.0	0.0	0.0
F4	*Avg	0.001414	0.562473	0.005177	0.00472	0.006956	0.005446
	Std	0.005836	0.001053	0.004379	0.004023	0.00587	0.005745
F5	*Avg	0.0	0.572473	0.456444	0.352104	0.09268	0.0
	Std	0.0	0.258421	0.339771	0.361023	0.215657	0.0
F6	*Avg	0.006737	0.989852	0.81519	0.706878	0.0	0.0
	Std	0.13901	0.001568	0.059026	0.180166	0.0	0.0
F7	*Avg	0.527019	0.999506	0.945361	0.567621	0.371442	0.000002
	Std	0.500243	0.000177	0.084910	0.372935	0.382261	0.000004
F8	*Avg	0.000037	0.001973	0.056086	0.002616	0.0	0.0
	Std	0.000055	0.001177	0.027614	0.007702	0.0	0.0
F9	*Avg	0.015927	0.008457	0.006589	0.006267	0.0	0.0
	Std	0.15204	0.003241	0.004216	0.006716	0.000002	0.0
F10	*Avg	0.211107	0.000065	0.000013	0.039348	0.002041	0.998520
	Std	0.79385	0.00015	0.000081	0.069976	0.004164	0.005363

\*\* avg diff (ln %).

*F1*, with its single central peak, is relatively easy to solve. Though all models came within 0 to 0.05% of the theoretical minimum, Yen's hybrid performed comparably less.

*F2* has strong inter-parameter linkage. Thus, after 500 generations, SPX hadn't yet managed to converge. Meanwhile, the others (in descending order of performance: SPX-Simplex-SA, SPX-Simplex, Yen's GA-Simplex, MAC-Simplex, and SA) did relatively well.

Discontinuous *F3* posed no problem for all the models; all having reached the theoretical minimum accurately.

SA's small perturbation adjustments contribute to its splendid performance in *F4*(a noisy function), so did the multi-parent addition model(MAC-Simplex). Yen's model, our SPX-Simplex-SA model, and SPX-Simplex came next. Finally, SPX came last.

SA and our SPX-Simplex-SA hybrid came within 0.05% of *F5*'s theoretical minimum for all trials. SPX-SA-Simplex, SPX-Simplex, MAC-Simplex, Yen's model, and SPX followed.

Rastrigin's function (*F6*) with its numerous local minima was a severe test for our models. Nonetheless, SPX-Simplex and SPX-Simplex-SA outperformed the rest. SA followed closely behind. SPX, which normally converges in the several thousands<sup>17</sup> for complicated functions didn't come within 5% of theoretical minimum. Yens model and MAC-Simplex likewise performed poorly.

Our model outstood the rest in Schwefel's *F7* (another multimodal variant). SA and SPX-Simplex are on equal footings while the rest remained unconverged.

Yen's 2-parent model did comparably worst for Griwank's problem. Multi-parent models and SA showed better to best results. Special mention goes to SA and our model, both having reached within 0.05% of the global minimum and to SPX which did surprisingly well here.

Dejong's Sine Function(*F9*) and the Stretched V Sine wave function (*F10*) are both epistatic and multimodal. In *F9*, the simplex hybrids did well. They are followed by Tsutsui's SPX and by SA respectively. In *F10* Yen's model nearly reached the minimum for all trials. SPX did well, our SPX-Simplex-SA hybrid, SPX-Simplex, MAC-Simplex and finally SA, succeeded.

### 5 Application in Resolution of Overlapping Signals

In many experiments, the observed signal is an

overlapped of other signals and noise, the resolution thereof into constituent signals, is critical for quantitative component analysis.

Assuming that the function of component signals to be known, signal resolution can be taken as an optimization problem. When the estimated signal is made to come as close as possible to the observed signal using Genetic Algorithms, component signal can be determined. With this in mind, let us assume all signals to be Gaussian. The estimated signal as well as the observed one can be defined using:

$$f(x) = \sum_{i=1}^N a_i \exp\left\{-\frac{(\bar{x} - b_i)^2}{c_i^2}\right\} + dx + e$$

$a_i$ ,  $b_i$ , and  $c_i$  are and parameters of component signals and  $dx+e$  is the background. By optimizing the cost function below, optimal values for parameters  $a_i$ ,  $b_i$ , and  $c_i$  are obtained.  $y(x)$  and  $f(x)$  represent the observed and estimated signal respectively.

$$fitness = \left\| y(x) - f(x) \right\|^2$$

#### 5.1 Simulation and Simulation Results

For this experiment, parameters  $a_1, b_1, c_1, a_2, b_2, c_2, \dots, a_N, b_N, c_N$ , are coded as alleles. Inasmuch as N(no. of signals) is unknown, a large N is opted.

The observed signal and the same signal with noise are shown in Fig. 10. A population size of 500 is used and the total number runs is 1000. SA, SPX and N-M Simplex are also used for comparison.

Costs after 5 runs for each model are shown in the figure below. The best graphs for each model, with the original signal superimposed are likewise shown in Fig 11. Our model was found to not only outperform the others could

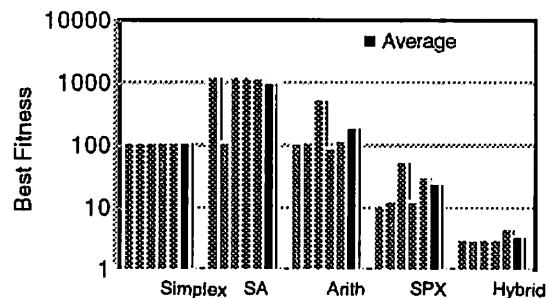


Fig 12. Fitness Values after 5 runs

obtain near perfect results even in the presence of degrading noise.

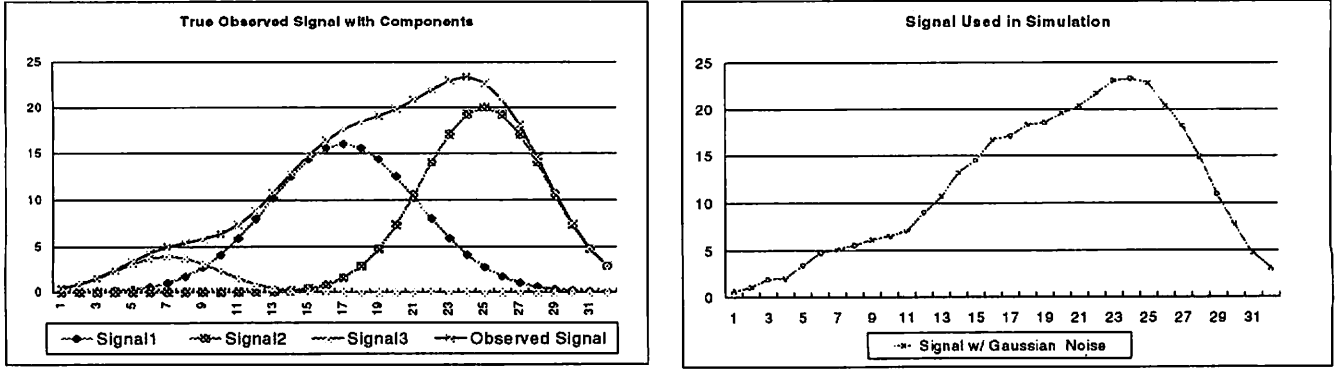


Fig 10. Signals used in Resolution Experiment

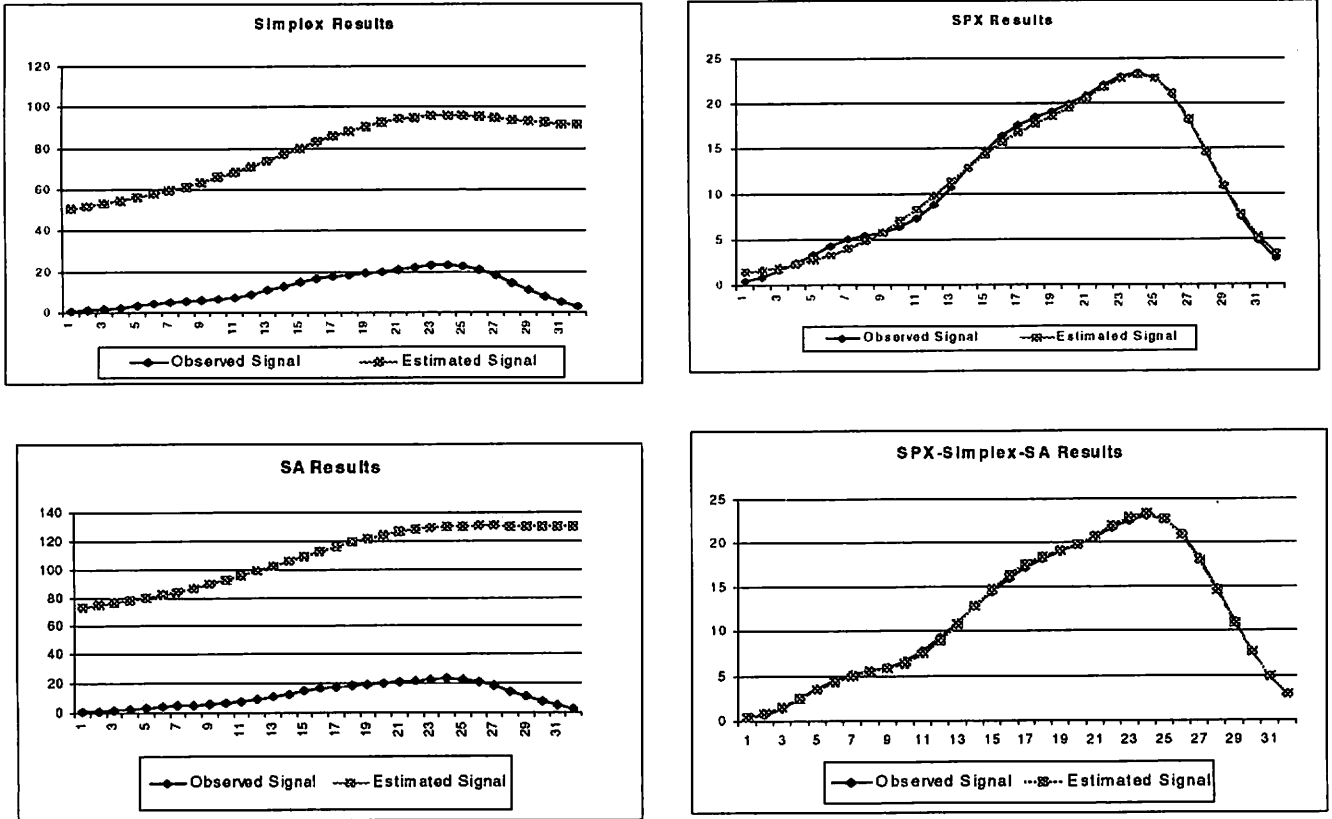


Fig 11. Resolution of Overlapped Signal Results

## 6 Conclusion and Future Work

As far as our experiment with test functions is concerned, although the models behaved diversely subject to the problem set, to some extent, our SPX-Simplex hybrid model with SA mutation did better than the rest in all problems.

Tsutsui's model (utilizing SPX alone) can address multimodality and epistasis, but convergence is slow. Yen's 2-parent model has similar convergence problems.

By substituting Yen's 2-parent addition crossover operator with MAC (yielding MAC-Simplex,) we noticed improvements in  $F1, F2, F4, F5, F7, F8$  and  $F9$ . Except  $F10$ , the use of more than 2 parent, even for restrictive crossover operations like Uniform Arithmetic Crossover, improved overall performance.

Since MAC doesn't do well with epistatic problems, substituting MAC with SPX (yielding SPX-Simplex) showed more suitable results (see  $F1, F2, F5, F6, F7, F8, F9$ , and  $F10$ .) SPX-Simplex, combines the coordinate-free, parent adaptive features of SPX and the speed and direction determining ability of the Simplex hybrid. Also, SPX-Simplex exceeds Tsutsui's SPX in performance.

Simulated Annealing (SA) works well for noisy functions but not as well on epistatic or multimodal functions. When juxtaposed with SPX-Simplex, it extended the model's fine tuning capabilities (see  $F1, F2, F4, F5, F7, F8$ , and  $F9$ ).

In summary, our experiment validated the following hypotheses:

- Multi-parent crossover, in certain respects, is far superior to 2-parent crossover.
- Combining multi-parent SPX and the GA-Simplex hybrid proved mutually beneficial. Also, SA mutation extends the overall fine tuning capability of the GA model.
- The modified hybrid architecture encourages competition and catalyzes convergence as the best p' results from various operations (e.g. SPX crossover, SA mutation and Concurrent SPX) comprise the next generation.

Not discounting the subtle effects of CSS and BEM, all things considered, our SPX-Simplex-SA(S3) model not only converges faster but is effective for nonseparable (epistatic), multi-modal problems as well.

When applied to real problems like signal resolution, our model performed robustly as expected. Even for  $N=4$ , a near perfect resolution was obtained after 1000 generations. As an extension, we are currently exploring

the hybrid's applicability in solving more specific signal separation problems like the determination of wind velocity from mixed Doppler radar data; initial tests of which, showed promising results.

## References

1. J. Yen, J. Liao, B. Lee, and D. Randolph, "A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method," *IEEE Transaction of Systems, Man and Cybernetics* **28**(2), 173-189 (1998).
2. J.H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1995).
3. J.A. Nelder, and R. Mead, "A simplex method for function minimization," *Computer Journal*, **7**, 308-313 (1965).
4. S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science* **220**, 671-680 (1983).
5. J. Renders, and H. Bersini, "Hybridizing genetic algorithms with hill-climbing methods for global optimization: 2 possible ways," *Proceedings of the 1<sup>st</sup> IEEE Conference on Evolutionary Computation*, 312-317 (1994).
6. M. Hagiwara, "Pseudo-Hill Climbing Genetic Algorithm (PHGA) for function optimization," *Proceedings of the 1993 Joint Conference on Neural Networks*, 713-715 (1993).
7. Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs* (Springer-Verlag, Berlin, 1992).
8. D. Whiteley, S. Rana, and R. Heckendorn, "Representation Issues in Neighborhood Search and Evolutionary Algorithms," *Genetic Algorithms in Engineering and Science* (Wiley and Sons, 1997).
9. D.E. Goldberg, "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking," *Complex Systems*, **5**, 139-167, (1991).
10. C.Z. Janikow, and Z. Michalewicz, "An experimental comparison of binary and floating point representations in Genetic Algorithms," *Proceedings of the 4<sup>th</sup> International Conference on Genetic Algorithms*, (1993).
11. L. Davis, *The Handbook of Genetic Algorithms* (Von Nostrand Reinhold, New York, (1991).
12. L.J. Eschelman, and J.D. Schaffer, "Real-coded generic algorithms and interval schemata," *Foundations of Genetic Algorithms 2* 187-202 (1993).
13. A.H. Wright, "Genetic Algorithms for Real Parameter Optimization," *Foundations of Genetic Algorithms*, 205-208, (1991).
14. I Ono, and S. Kobayashi, "A Real-coded Genetic Algorithm for function optimization using unimodal normal distribution crossover," *Proceedings of the*

- 7<sup>th</sup> International Conference on Genetic Algorithm, 246-253 (1997).
15. S. Tsutsui, "Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring," *Proceedings of the 5<sup>th</sup> International Conference on Parallel Problem Solving from Nature*, 428-437 (1998).
  16. S. Tsutsui, and A. Ghosh, "A study on the effect of multi-parent recombination in real coded genetic algorithms," *Proceedings of the 1998 IEEE International Conference of Electric Circuits*, 828-833 (1998).
  17. S. Tsutsui and M. Yamamura, "Multi-parent recombination with simplex crossover in real coded genetic algorithms," *Proceedings of the GECCO '99* (1999).
  18. D.H. Wolpert, and W.G. Macready, "No free lunch theorems for search," *Technical Report SFI-TR-95-02-010*, Sta. Fe. Institute (1995).
  19. I Ono, M. Yamamura, and S. Kobayashi, "A Genetic Algorithm with characteristics preservation for function optimization," *Proceedings of IIZUKA 1996*, 511-514 (1996).
  20. S. Kobayashi, I. Ono, and M. Yamamura, "An efficient genetic algorithm for job shop scheduling problems," *Proceedings of the 6<sup>th</sup> International Conference on Genetic Algorithms*, 506-511(1995).
  21. A.E. Eiben, P.E. Raue, and Z. Ruttkay, "Genetic algorithms with multi-parent recombination," *Proceedings of the 3<sup>rd</sup> International Conference on Parallel Problem Solving from Nature*, 78-87 (1994).
  22. A.E. Eiben, and T. Bäch, "Empirical investigation of multiparent recombination operators in evolution strategies," *Evolutionary Computation*, 5(3), 347-365 (1997).
  23. D.E. Goldberg, *Genetic Algorithms in search, optimization and machine learning* (Addison-Wesley, 1989).
  24. Y.W. Chen, Z. Nakao, K. Arakaki, X. Fang, and S. Tamura, "Restoration of gray images based on a genetic algorithm with Laplacian constraint," *Fuzzy Sets and Systems* 103, 285-293 (1999).
  25. F. F. Ali, Z. Nakao, and Y.W. Chen, "Two neural network approaches to two-dimensional CT image reconstruction," *Fuzzy Sets and Systems* 103, 295-302 (1999).
  26. D. Whitley, S. Rana, J. Dzubera, and K. Mathias, "Evaluating evolutionary algorithms," *Artificial Intelligence* 85, 245-276 (1996).
  27. W. Spendley, G.R.Hext, and F.r. Himsworth, "Sequential application of simplex designs in optimization and evolutionary operation," *Technometrics*, 4, 441-461 (1962).
  28. D.E. Goldberg, "Analysis of the behavior of a class of genetic adaptive systems," Ph.d. dissertation, Dept. Comput. Commun. Sci., Univ. Michigan, (1975).
  29. S. Minami, *Signal Processing for Scientific Instrumentation* (CQ Press, 1994) in Japanese.
  30. F. Zenitani, and S. Minami, *Japanese Journal of Applied Physics* 13(5), 875-881 (1974).
  31. R.D.B. Freshet, and E. Suzuki, "Resolution of overlapping absorption bands by least squares procedures," *Analytical Chemistry*, 38(12), 1770-1773 (1966).
  32. L. Ingber, "Simulated annealing: practice versus theory," *Math. Compt. Model*, 18(11), 29-57 (1993).
  33. L. Ingber, and B. Rosen "Genetic algorithms and very fast simulated annealing," *Math. Compt. Model*, 16(11), 87-100 (1992).
  34. P.S. de Souza, and S.N. Talukdar, "Genetic algorithms in asynchronous teams," *Proceedings of the 4<sup>th</sup> International Conference on Genetic Algorithms*, 392-297, (1991).
  35. J. Renders, and S. Flasse, "Hybrid methods using genetic algorithms for global optimization," *IEEE Transactions on Systems, Man and Cybernetics*, 26, 243-258 (1996).
  36. D.H. Ackley, "Stochastic iterated genetic hillclimbing," Ph.d. dissertation, Carnegie Mellon Univ. (1987).