

Monogamous Pairs Genetic Algorithm (MopGA)

Ting Yee Lim, Ahamad Tajudin Khader

Abstract—As the race in producing better Genetic Algorithms (GAs) to alleviate the notorious premature convergence problems heats on, the danger of overly complex solutions, ignoring the practicality and feasibility of basic algorithms continues in some researches. In this paper, we propose a new variant of GA with decent complexity without losing the search power. Our approach is inspired by the monogamous behavior observed in nature. The efficacy of MopGA is verified on nine benchmark numerical test functions. The results are mostly comparable to standard GA and even achieve better overall average reliability and speed.

Index Terms—Genetic algorithm, monogamy, numerical function optimization

I. INTRODUCTION

Nature sometimes exhibits the formation of enduring relationships between mating partners. In modern human society, some avian models [1], fish [2], rodents [3] and even lizards [4], pair bonds are integral aspects of their social behaviour. These species usually share the same mating partners throughout their lifetime – social monogamy. On the other hand, computational evolution such as the Genetic Algorithm (GA) that derives its behaviour from a metaphor of processes in natural evolution is however never too close to nature in certain aspects, for instance, the negligence of monogamous behaviour. Canonical or standard GA (SGA) comprises only panmictic population with the simple idea of crossing over among and occasional mutation within individuals (i.e. solution strings) that are each subjected to selection pressure grounded by Charles Darwin’s theory of “survival of the fittest”. GA has made many success stories of solving optimization problems [5, 6], scheduling problems [7-9], pattern recognition [10], etc. that have otherwise been proven NP-hard to be solvable within reasonable polynomial timeframe. Nonetheless, again and again, GA has fallen short into traps of local optima, leading to premature convergence. Compellingly, almost over five decades of researches have given rise to plethora variants of GAs enumerated briefly as island models [11], cellular models [12], hierarchical models [13, 14], and heterogeneous population-based [15-17]. The first three models represent spatial differentiation approaches where a population commonly segmented into sub-populations, grid-based or hierarchical-based. Restricted interactions are permitted across the boundaries usually at specific time interval. Meanwhile in heterogeneous population-based approach, different groups of interacting individuals exist in the form of male-female attractor, predator-prey, or competitor-co-operator players.

While literature is by far dominated by studies of interactive properties among individuals in GA populations whether through spatial segregation or group differences, few have investigated the essence of interactive roles between parents and their offspring, and even lesser to our knowledge on the possibilities of monogamous parenting.

[18] incorporates Baldwin effect that influences an individual’s innate fitness. At the same time, offspring’s fitness is also under the influence of their parents. However, mating partners are not preserved as selection for mating and survival is done through elitism and binary tournament selection.

Perhaps, the concept of family introduced in [19-21] has better resemblance to monogamous behaviour. In the Genetic Invariance Genetic Algorithm or GIGA, a pair of parents generates a family that is a set of pairs produced by a set of crossover operations on the single pair parents. The best pair is then selected from the family to replace the parents. However, the main difference contrasting our approach to GIGA is that in our approach, no parent selection is needed after the initial stage. Furthermore, we allow only a pair of children to be generated per generation and whenever the children pair-fitness exceeds their parents’, they will replace their parents in the next generation.

Our approach has somewhat more similarities to fixed parents strategy explored in the work [22]. The authors introduce a complementary-parent strategy to initialize and reproduce individuals with complementary chromosomes. As there exists no selection strategies, mutation, and incest prevention, the algorithm is coined the Pseudo Genetic Algorithm (PGA). In contrast to our proposed method, PGA replaces parents with their offspring at every generation, irrespective of their fitness values. The authors admit that after many generations of breeding, the average population fitness may be kept unimproved. It is merely an elitist-searching process. We overcome this problem by introducing the concept of infidelity commonly found in monogamous society. On the other hand, as the race in producing better GAs to alleviate the notorious premature convergence problems heats on, the danger of overly complex solutions, ignoring the practicality and feasibility of basic algorithms continues in some researches. In this paper, we propose a new variant of GA with decent complexity of $3O(NK)$, where N and K being the population size and chromosome length, respectively, without losing the search power. Inspired by social monogamy, we present the Monogamous Pairs Genetic Algorithm (MopGA). The framework allows us to drop a component of conventional GA, namely the parent selection and leads to smaller footprint. Monogamous pairs reproduce continually throughout the entire evolution and will only be replaced by their better offspring.

Manuscript published on 30 June 2013.

* Correspondence Author (s)

Lim Ting Yee, School of Computer Science, Universiti Sains Malaysia (USM), Penang, Malaysia.

Ahamad Tajudin Khader, School of Computer Science, Universiti Sains Malaysia (USM), Penang, Malaysia.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

In order to maintain higher diversity, we introduce the concept of artificial infidelity that is also a common phenomenon observed from the natural monogamous society.

The remaining of this paper is organised as follows. Section II describes each component of MopGA in greater details. The ensuing section (Section III) is devoted to experimental setup, test suite description, and results and discussion. The paper ends with some concluding remarks in Section IV.

II. MONOGAMOUS PAIRS GA (MOPGA)

A. Inspiration & Hypothesis

Though relatively rare phenomenon (e.g. 3-5% from a total of 4000 mammalian species [23]), social monogamy is rather common at least in most bird species (90% of birds are monogamic and usually exhibit biparental care of young [23]). The direct and indirect benefits of such behaviour have long been studied [1, 2, 4, 24, 25], though surprisingly little have been known; most likely due to the complexity of these bonds.

Here, we develop and explore the hypothesis that the norms that compose the package of monogamous pairs GA is favoured by suppression of parent selections. Finding mates can be expensive, especially when it has to be implemented in every generation of the evolutionary process. Using rank-based selection scheme for instance, requires that a population to be first sorted by fitness. Sorting could be an expensive burden to the algorithm if the population involves is large, especially. Generally, rank selection, roulette wheel selection, steady state selection, and tournament selection has a complexity of $O(N \log N)$, $O(N^2)$, $O(N \log N)$, and $O(N)$, each respectively, where N is the population size [26]. MopGA forsake parent selection, and instead allowing better offspring to automatically book a slot into the next generation.

Monogamous bonding could also insure mutual defence on the pair (more details in section B).

Furthermore, if every member of the population constitutes a point in the search space, and crossover restricts the search spaces between the two parents, then repeatedly crossing over the same parents until no better child is found may be beneficial after all.

B. Monogamous Pairs

Unlike conventional GAs, parent pairings are decided randomly during initialization stage and stay constant throughout the evolutionary process. The product of crossing over and mutation is a new pair of potential parents that will replace their parents if they possess better pair-fitness. This way, free elitism is achieved as better pair automatically survives into the next generation.

As noted, monogamous pair also contributes to pair-fitness. On the flip side, conventional GAs identify individuals with their personal fitness values. In theory, a monogamous pairing insures mutual defence on both mates. ‘Weaker’ individuals can have better chances of survival when paired with ‘stronger’ individuals. With such, domination of population by strong individuals commonly seen in SGA can be avoided.

Empirically, each mate contributes to the pair-fitness as follow:

$$f_{pair} = 0.7f_{strong} + 0.3f_{weak} \quad (1)$$

Stronger mate, f_{strong} has higher contribution on the final pair-fitness, f_{pair} , compared to weaker mate, f_{weak} .

C. Infidelity

An interesting phenomenon observed in monogamous society is that in most populations, at least a few offspring in each generation is the result of extrapair copulations, i.e. matings with partners other than a pair member.

In MopGA, infidelity automatically takes place when crossover fails. Hence, no additional parameter tuning is added into the algorithm. During this stage, a random pair is selected from the population. Next, a new ‘extramarital’ pair is formed by randomly selecting one member from each pair. Finally, the extramarital pair replaces the old pair in the next generation (*despite their pair-fitness differences*). Elitism is broken this way to make room for simplicity and provide chances to accept bad ‘moves’ but possibility fruitful output in future generations.

The positive effect of infidelity has been significantly displayed on the empirical results. In the obvious, infidelity has helped to promote diversity by spreading information between different partners that may be restricted by initial monogamous norms. Consequently, this leads to better chances of escaping local traps.

D. MopGA Framework

MopGA begins by random initialization and pairings of individuals in the population pool. After this startup phase, each pair undergoes the process of reproduction: crossover and mutation. The children pair that has better pair-fitness than their parents will replace their parents in the next generation.

The one-cut-point crossover integrated with an arithmetical operator derived from convex set theory [27-29] is adopted. After randomly selecting one cutting point, the right parts of the two parents are exchanged and the linear combinations at the cut-point genes are calculated to generate new offspring. For instance, let two parents be $v = (v_1, v_2, \dots, v_N)$ and $w = (w_1, w_2, \dots, w_N)$. Further, let k -th be the cut-point. Finally, the resulting offspring are:

$$\begin{aligned} v' &= (v_1, v_2, \dots, v'_k, w_{k+1}, w_{k+2}, \dots, w_N) \\ w' &= (w_1, w_2, \dots, w'_k, v_{k+1}, v_{k+2}, \dots, v_N) \end{aligned} \quad (2)$$

where $v'_k = v_k + b(w_k - v_k)$, $w'_k = w_k + b(v_k - w_k)$, l_k and u_k are the domain of v_k , and b is a random value subjected to $b \in \{0, 0.1, 0.2, \dots, 0.9, 1\}$. Note that the new value v'_k is generated by the discrete random value b with v_k and w_k , which is likely to have different values in different generations.

Following the same idea, the mutation operator used is also derived from convex set theory [27, 29]. This method is specially designed to enhance fine-tuning capabilities. The mutation operator operates on two randomly selected genes on an individual. For example, let

$v = (v_1, v_2, \dots, v_i, v_j, v_k, \dots, v_N)$ represents an individual to be mutated, and v_i and v_k are the randomly selected mutation genes. Then the resulting genes are given by:

$$v'_i = (1 - b)v_i + bv_k, v'_k = bv_i + (1 - b)v_k \quad (3)$$

Observe that the new genes are now closer stepwise.

Furthermore, dynamic mutation ranging from [0.1, 0.3] is also applied following the function [28]:

$$P_m = 0.1 + \frac{0.2t}{G} \quad (4)$$

where t denotes the current generation and G is the maximum generations.

On the other hand, infidelity occurs whenever crossover fails. During this stage, the monogamous pair may find random mates from the population. The new pair automatically gets promoted into the next generation. Like in nature, infidelity plays a role in generating diversity.

Finally, the evolutionary process terminates when the maximum predefined generation has been met. Figure 1 illustrates the overall workflow of MopGA.

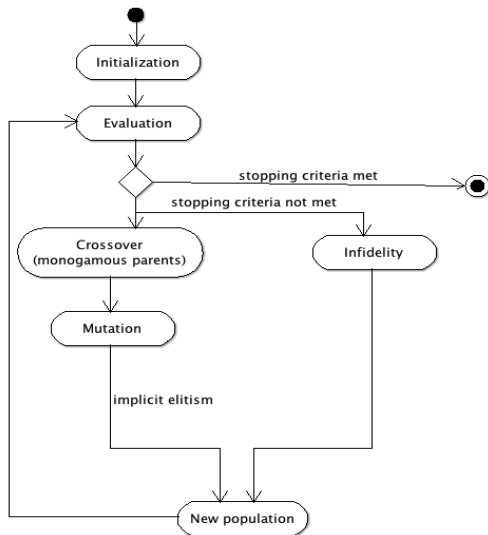


Figure 1. MopGA

III. EXPERIMENTS AND RESULTS

This section presents the experiments, empirical results and discussions to validate the proposed MopGA techniques.

A. Test suite

A collection of nine benchmark test functions comprising both unimodal and multimodal problems are selected. All functions are tested on 30 and 50 dimensions (*the symbol d*,

if necessary, will substitute the term “dimensions” in the rest of the paper). This section provides short reference and justification to the choice of test suite and a summary is presented in TABLE 1. All functions have global optimum of zero. For practical purposes, acceptance level is set for each function at which the algorithm is considered successful in finding the target solution.

The first four functions (sphere, axis parallel hyper-ellipsoid, rotated hyper-ellipsoid, and sum of different power) being unimodal contain only one optimum, whereas the other test functions (Rosenbrock’s, Rastrigin’s, noncontinuous Rastrigin’s, Griewangk’s and Ackley’s) are multimodal optimizations problems, i.e. containing many local optima, but only one global optimum.

Sphere (f_{sphere}) belongs to DeJong’s first test function. It is smooth, strongly convex, and symmetric. A variant of this is the axis parallel hyper-ellipsoid (f_{aphe}) function that is also known as weighted sphere model. Next, an extension to f_{aphe} is the Schwefel’s function 1.2 or the Rotated hyper-ellipsoid function (f_{rhe}). It is continuous, and convex. The last unimodal function appearing in this paper is the sum of different power function (f_{sdp}).

Interestingly, in two dimensions, Rosenbrock’s function (f_{rosen}) is unimodal but considered multimodal for higher ones [30]. Also known as banana function, f_{rosen} has global optimum that lies inside a long, narrow, parabolic shaped flat valley. It is known to be challenging for GAs to solve.

The Rastrigin’s function (f_{rast}) is separable and has many suboptimal peaks whose values increase as the distance from the global optimum point increases. It is considered fairly difficult problem for GAs due to its large search space and large number of local minimums with a complexity of $O(n \ln n)$, where n is the number of parameters.

Since all the test functions are continuous, we especially included a discontinuous function, namely the noncontinuous Rastrigin (f_{rastNC}) [31]. As opposed to continuous functions, discontinuous functions commonly display characteristics such as having breaks, holes or jumps in the graph.

Like f_{rosen} , interdependencies between variables exist in Griewangk’s function (f_{grie}). The terms of summation produce a parabola with local optima stay above the parabola level. The dimensions of search range increase on the basis of the product, resulting in the decrease of the local minimums. It has a complexity of $O(n \ln n)$.

Finally, the Ackley’s function (f_{ackley}) is not separable and it appears unimodal at low resolution but the second exponential term covers the landscape with many small peaks and valleys. It is also a widely used multimodal test function.

TABLE 1. Summary of numerical function test suite

No	Functions	Modal	d	Optimum	Acceptance
1.	Sphere function: $f_{sphere}(x) = \sum_{i=1}^n x_i^2 \quad [-5.12, 5.12]$	Unimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50
2.	Axis parallel hyper-ellipsoid: $f_{aphe}(x) = \sum_{i=1}^n (i \cdot x_i^2) \quad [-5.12, 5.12]$	Unimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50

Monogamous Pairs Genetic Algorithm (MopGA)

No	Functions	Modal	d	Optimum	Acceptance
3.	Rotated hyper-ellipsoid: $f_{rhe}(x) = \sum_{i=1}^n \sum_{j=1}^i x_j^2 \quad x_i \in [-65.536, 65.536]$	Unimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50
4.	Sum of different Powers: $f_{sdp}(x) = \sum_{i=1}^n x_i ^{(i+1)} \quad x_i \in [-1, 1]$	Unimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50
5.	Rosenbrock's valley: $f_{rosen}(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad x_i \in [-2.048, 2.048]$	Unimodal. But multimodal when d > 2	30	0.0000E+00	1.0000E+2
			50	0.0000E+00	1.0000E+2
6.	Rastrigin's function: $f_{rast}(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad x_i \in [-5.12, 5.12]$	Multimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50
7.	Noncontinuous Rastrigin's function: $f_{rastinc}(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad x_i \in [-5.12, 5.12]$ <p>where</p> $x_i = \begin{cases} x_i & x_i < 0.5 \\ \frac{\text{round}(2x_i)}{2} & x_i \geq 0.5 \end{cases}$	Multimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50
8.	Griewangk's function: $f_{grie}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad x_i \in [-300, 300]$	Multimodal	30	0.0000E+00	1.0000E-50
			50	0.0000E+00	1.0000E-50
9.	Ackley's function: $f_{ack}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(-\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e \quad x_i \in [-32.768, 32.768]$	Multimodal	30	0.0000E+00	1.0000E-15
			50	0.0000E+00	1.0000E-15

B. Experimental Setup

This section describes the experimental setup to compare the efficacy of MopGA against SGA on nine test functions with 30 and 50 dimensions each. For fair comparison, all common parameters are kept the same for both algorithms as shown in TABLE 2.

Real-coded representation is used to achieve higher accuracy and it works well under the mutation and one-point crossover with arithmetical operator derived from convex theory described earlier (II.D). While mutation operation is dynamic between the ranges of 0.1 to 0.3, the crossover probability has been set constant at 0.85.

For SGA, binary tournament selection is used. Moreover, elitism of 10% of the population size (six in this case) or at least one elite individual, whichever is higher, is preserved for SGA.

All algorithms will run for a maximum of 25,000 generations over 30 independent trials each and their mean results are used in comparison.

Both algorithms are coded in Java under MAC OS X version 10.6.8 platform on an Intel Core i5 2.4GHz processor with 4GB of RAM.

C. Performance measures

For comparison purposes, three performance metrics are used are, namely (1) accuracy, (2) speed, and (3) success rate.

Firstly, the best objective function value (**Mean Best**) on the mean of all best objective function values for all 30 independent trials indicates the accuracy belonging to a specific test. The standard deviation (**Std dev.**) provides a cue on the spread of the best objective function values.

Next, the second main performance measure is speed, which is measured by the mean of fitness evaluations (**Mean FEs**) before termination. For instance, if a GA with population size of 20, and each generation generating 20 offspring terminates at the 1000-th generation, then the total FEs will be 20,000. In the worst case when the GA never finds acceptable solution, this value equals the maximum number of fitness evaluations.

Lastly, the success rate (**Success rate**) defines the percentage of runs where an acceptable solution has been found. For instance, a 0% indicates that the GA never finds any acceptable solution. On contrary, a 100% indicates that the GA always finds acceptable solutions in all runs. The higher the success rate, the more reliable the algorithm is.



TABLE 2. Parameters for MopGA and SGA

Parameters	Algorithms	
	SGA	MopGA
Crossover probability, P_c	0.85	
Crossover type	One-point crossover with arithmetical operator derived from convex theory	
Mutation probability, P_m	[0.1, 0.3]	
Mutation type	Mutation operation derived from convex theory	
Population size, N	6	
Maximum generation, G	25,000	
Representation	Real value-coded	
Parent selection	Binary tournament selection	-
Elitism	10% of N or 1, whichever is higher	Free elitism

D. Accuracy

The simulation results are summarized in TABLE 3. Values in bold indicate best results found between the two comparative algorithms.

MopGA records highest accuracy on 14 out of 18 tests (including both 30- and 50- dimension problems) with 9 wins (f_{rosen} , f_{aphe} , f_{rhe} , f_{ack} , and $50d_{frast}$), and 5 even results (f_{sdp} , $frastNC$, and $30d_{frast}$), compared to SGA. The standard deviation values are also smaller for MopGA, indicating that the results are more stable.

As with the theorem of “no free lunch” [32], any one algorithm cannot possibly offer better performance on every kind of problem compared to all other algorithms. Such case is also observed in our results. SGA is more superior in f_{rosen} and f_{grie} . Both functions have the reputations of being challenging for most GAs.

E. Convergence Speed

At first glance, TABLE 3 reveals that SGA comprises more functions with smaller mean FEs, especially on multimodal problems, except for f_{ack} , while MopGA continues to chalked better means FEs on unimodal problems, except for f_{sdp} .

TABLE 3. Search comparison between SGA and MopGA on 9 benchmark test functions.

Function	d	SGA				MopGA			
		Mean Best	Std.dev	Mean FEs	Success rate (%)	Mean Best	Std.dev	Mean FEs	Success rate (%)
f_{sphere}	30	2.0196E-143	1.0812E-142	35,766.78	100.00	0.0000E+00	0.0000E+00	18,265.20	100.00
	50	1.1076E-79	5.9640E-79	80,583.60	100.00	0.0000E+00	0.0000E+00	39,075.00	100.00
f_{aphe}	30	1.6948E-139	9.1244E-139	33,189.60	100.00	0.0000E+00	0.0000E+00	21,746.40	100.00
	50	1.2242E-79	6.5136E-79	83,397.18	100.00	0.0000E+00	0.0000E+00	40,121.82	100.00
f_{rhe}	30	4.7131E-141	2.5370E-140	34,102.38	100.00	0.0000E+00	0.0000E+00	20,225.58	100.00
	50	1.5567E-83	6.9161E-83	75,871.80	100.00	0.0000E+00	0.0000E+00	42,245.58	100.00
f_{sdp}	30	0.0000E+00	0.0000E+00	9,720.60	100.00	0.0000E+00	0.0000E+00	23,640.42	100.00
	50	0.0000E+00	0.0000E+00	14,940.18	100.00	0.0000E+00	0.0000E+00	50,455.02	100.00
f_{rosen}	30	4.1810E+01	2.4066E+01	7,171.38	100.00	6.4105E+01	4.0245E+01	49,598.82	80.00
	50	1.0075E+02	4.6800E+01	92,925.18	50.00	1.4489E+02	4.2745E+01	144,991.38	10.00
f_{rast}	30	0.0000E+00	0.0000E+00	5,791.80	100.00	0.0000E+00	0.0000E+00	14,916.42	100.00
	50	1.8948E-15	1.0204E-14	14,554.98	96.67	0.0000E+00	0.0000E+00	29,999.82	100.00
$frastNC$	30	0.0000E+00	0.0000E+00	5,760.78	100.00	0.0000E+00	0.0000E+00	13,464.00	100.00

The average mean FEs calculated for all tests however indicates that MopGA offers the better overall speed of 44,514.42 FEs compared to 47,287.78 FEs for SGA.

F. Reliability

Returning to TABLE 3, MopGA also offers higher percentage of trials to reach acceptable solutions averaged over all the test functions. It recorded an average reliability of 88.33% while SGA recorded only 87.78%.

On a closer inspection, we again observe that SGA performs better on f_{rosen} and f_{grie} functions. Meanwhile MopGA continues its superiority in f_{ack} and $50d_{frast}$. Both algorithms record similar results (100% reliability) for the rest of the tests.

G. Computation complexity

One of the major strength of MopGA highlighted in this paper is the reduction in computational complexity, yet preserving the search ability. Sections D, E, and F justified the second half of the claim. In this section, we verify the computational complexity of MopGA.

Let us assume N is the population size and K is the chromosome length. Broadly, when considering only the major components in a GA, TABLE 4 reveals that the overall complexity of SGA is $5O(NK)$ per generation (previously analysed by [22]). In contrast, MopGA has a complexity of only $3O(NK)$ per generation. This is achieved through the omission of parent selection and free elitism. During normal reproduction, the worst complexity for MopGA is $3O(NK)$, but during infidelity mode, the complexity may drop further to $2O(NK)$ as no crossover or mutation is needed.

In solving real-world problems, computational cost is expensive. An increase of $2O(NK)$ per generation may overwhelm the algorithm overhead. Hence, in view of practicality and feasibility, striving for lower computational complexity is inevitable. MopGA has clearly made this possible without losing the generality of search power.

	50	0.0000E+00	0.0000E+00	11,263.38	100.00	0.0000E+00	0.0000E+00	27,361.80	100.00		
f_{grie}	30	1.7838E-02	4.1189E-02	66,019.02	80.00	1.1370E-01	1.6429E-01	103,842.60	53.33		
	50	2.9893E-02	7.6514E-02	73,081.38	83.33	2.0472E-01	2.3370E-01	106,720.98	46.67		
f_{ack}	30	3.3751E-15	3.6864E-15	65,650.98	63.33	8.8818E-16	0.0000E+00	18,963.78	100.00		
	50	1.4033E-14	6.4310E-15	141,388.98	6.67	8.8818E-16	0.0000E+00	35,625.00	100.00		
Overall average				47,287.78	87.78	Overall average				44,514.42	88.33

TABLE 4. Comparison on complexity

Components	Algorithms	
	SGA	MopGA
Evaluation	O(NK)	O(NK)
Parent selection (binary tournament selection)	O(N)	-
Crossover	O(NK)	O(NK)
Mutation	O(NK)	O(NK)
Elitism	O(N)	-
Infidelity	-	[O(NK)]
Total complexity	5O(NK)	3O(NK)

IV. The assertion is later confirmed empirically as shown in TABLE 5 and

FIGURE 2. The results reveal that SGA spends more time in processing per generation compared to MopGA on all test functions. On eight test functions in particular (f_{sphere} , f_{aphe} , $50d_{frhe}$, $50d_{fgrie}$, $50d_{frosen}$, and $50d_{fack}$), the processing mean time per generation for SGA even doubled that of MopGA.

V. FINAL REMARKS

In accordance with natural social monogamy, we presented the monogamous pairs GA. It is not difficult to see that a pure monogamous framework could easily lose diversity and lead to suboptimal solutions. Hence, the infidelity concept has been incorporated, allowing occasional scrambling of information between extra partners. The final results are indeed satisfying.

By omitting parent selection at every generation and allowing same pairs (monogamous pairs) to reproduce continuously, we have shown that MopGA can reduce the algorithm complexity by 2O(NK) per generation. Such a reduction could be essential cost saving for real-world problem solving.

TABLE 5. Comparison on mean time taken per generation

Function	d	SGA	MopGA
		Mean time per generation (s)	Mean time per generation(s)
f_{sphere}	30	0.1010	0.0391
	50	0.1846	0.0380
f_{aphe}	30	0.0872	0.0379
	50	0.1789	0.0387
f_{rhe}	30	0.1023	0.0627
	50	0.2198	0.1027
f_{sdp}	30	0.0412	0.0380

Function	d	SGA	MopGA
		Mean time per generation (s)	Mean time per generation(s)
f_{rosen}	50	0.0644	0.0427
	30	0.0702	0.0405
f_{rast}	50	0.1199	0.0449
	30	0.0431	0.0401
f_{rastNC}	50	0.0561	0.0463
	30	0.0466	0.0411
f_{grie}	50	0.0609	0.0469
	30	0.0648	0.0446
f_{ack}	50	0.1091	0.0504
	30	0.0574	0.0433
f_{ack}	50	0.1151	0.0449

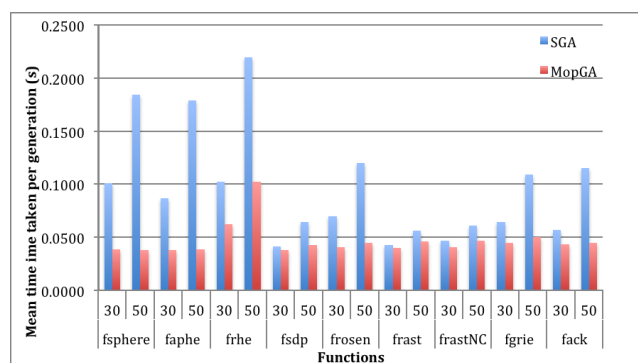


Figure 2. Comparison on mean time taken per generation for SGA and MopGA on nine test functions

At the same time, as depicted in the benchmark tests, MopGA is mostly comparable to SGA and even achieves better overall average reliability and speed. In terms of accuracy, MopGA is superior to SGA in most problems, except for f_{rosen} and f_{grie} . Both functions have the reputations of being difficult to solve by most GAs.

Nonetheless, it is noteworthy that the results are preliminary in nature, though providing some initial evidence of the potential problem solving capabilities of MopGA. Further refinement of the approach along with additional comparisons with other existing optimization techniques would be obligatory in future.

REFERENCES

- [1] B. Hatchwell, A. Russell, D. Ross, and M. Fowlie, "Divorce in cooperatively breeding long-tailed tits: a consequence of inbreeding avoidance?," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 267, pp. 813-819, 2000.
- [2] R. G. Oldfield and H. A. Hofmann, "Neuropeptide regulation of social behavior in a monogamous cichlid fish," *Physiology & Behavior*, vol. 102, pp. 296-303, 2011.



- [3] K. A. Young, K. L. Gobrogge, Y. Liu, and Z. Wang, "The neurobiology of pair bonding: insights from a socially monogamous rodent," *Frontiers in neuroendocrinology*, vol. 32, pp. 53-69, 2011.
- [4] S. T. Leu, P. M. Kappeler, and C. M. Bull, "The influence of refuge sharing on social behaviour in the lizard *Tiliqua rugosa*," *Behavioral ecology and sociobiology*, vol. 65, pp. 837-847, 2011.
- [5] Y. Sakurai, K. Takada, N. Tsukamoto, T. Onoyama, R. Knauf, and S. Tsuruta, "Inner Random Restart Genetic Algorithm to optimize delivery schedule," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, 2010, pp. 263-270.
- [6] Y. Sakurai, K. Takada, N. Tsukamoto, T. Onoyama, R. Knauf, and S. Tsuruta, "Backtrack and Restart Genetic Algorithm to Optimize Delivery Schedule," in *Signal-Image Technology and Internet-Based Systems (SITIS), 2010 Sixth International Conference on*, 2010, pp. 85-92.
- [7] R. Chandrasekharan, V. Vinod, and S. Subramanian, "Genetic algorithm for test scheduling with different objectives," *Integration, the VLSI Journal*, vol. 17, pp. 153-161, 1994.
- [8] C. Giri, D. K. R. Tipparthi, and S. Chattopadhyay, "Genetic algorithm based approach for hierarchical SOC test scheduling," 2007, pp. 141-145.
- [9] C. Sai Ho and C. Hing Kai, "A Two-Level Genetic Algorithm to Determine Production Frequencies for Economic Lot Scheduling Problem," *Industrial Electronics, IEEE Transactions on*, vol. 59, pp. 611-619, 2012.
- [10] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application," *Applied Soft Computing*, vol. 12, pp. 1592-1606, 2012.
- [11] B. Artyushenko, "Analysis of global exploration of island model genetic algorithm," 2009, pp. 280-281.
- [12] Alba, Enrique, Hugo Alfonso, and Bernabé Dorronsoro, "Advanced models of cellular genetic algorithms evaluated on SAT," presented at the Proceedings of the 2005 conference on Genetic and evolutionary computation, Washington DC, USA, 2005.
- [13] A. C. M. Oliveira, S. L. Ma, L. A. N. Lorena, S. Stephany, and A. J. Preto, "An Hierarchical Fair Competition Genetic Algorithm for Numerical Optimization," 2004.
- [14] J. J. Hu and E. D. Goodman, "The hierarchical fair competition (HFC) model for parallel evolutionary algorithms," 2002, pp. 49-54.
- [15] S. Wagner and M. Affenzeller, "SexualGA: Gender-specific selection for genetic algorithms," 2005, pp. 76-81.
- [16] J. Sanchez-Velazco and J. A. Bullinaria, "Sexual selection with competitive/co-operative operators for genetic algorithms," 2003.
- [17] R. Lahoz-Beltra, G. Ochoa, and U. Aickelin, "Cheating for problem solving: a genetic algorithm with social interactions," 2009, pp. 811-818.
- [18] M. Zhang, S. Zhao, and X. Wang, "A hybrid self-adaptive genetic algorithm based on sexual reproduction and Baldwin effect for global optimization," 2009, pp. 3087-3094.
- [19] J. C. Culberson and U. o. A. D. o. C. Science, *Genetic invariance: A new paradigm for genetic algorithm design*: Citeseer, 1992.
- [20] J. C. Culberson, "Mutation-crossover isomorphisms and the construction of discriminating functions," *Evolutionary Computation*, vol. 2, pp. 279-311, 1994.
- [21] J. C. Culberson and U. o. A. D. o. C. Science, *GIGA program description and operation*: Citeseer, 1992.
- [22] Q. Chen, Y. Zhong, and X. Zhang, "A pseudo genetic algorithm," *Neural computing & applications*, vol. 19, pp. 77-83, 2010.
- [23] G. Schuiling, "The benefit and the doubt: why monogamy?," *Journal of Psychosomatic Obstetrics & Gynecology*, vol. 24, pp. 55-61, 2003.
- [24] A. M. J. Anacker, T. H. Ahern, L. J. Young, and A. E. Ryabinin, "The Role of Early Life Experience and Species Differences in Alcohol Intake in Microtine Rodents," *PLoS one*, vol. 7, p. e39753, 2012.
- [25] M. U. Gruebler and B. Naef - Daenzer, "Survival benefits of post - fledging care: experimental approach to a critical part of avian reproductive strategies," *Journal of Animal Ecology*, vol. 79, pp. 334-341, 2010.
- [26] S. Gandhi, D. Khan, and V. S. Solanki, "A Comparative Analysis of Selection Scheme," *International Journal of Soft Computing*, vol. 2, 2012.
- [27] T. Jinn-Tsong, L. Tung-Kuan, and C. Jyh-Horn, "Hybrid Taguchi-genetic algorithm for global numerical optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 8, pp. 365-377, 2004.
- [28] S. T. Hsieh, T. Y. Sun, and C. C. Liu, "Potential offspring production strategies: An improved genetic algorithm for global numerical optimization," *Expert Systems with Applications*, vol. 36, pp. 11088-11098, 2009.
- [29] X. Pan, L. Jiao, and F. Liu, "An improved multi-agent genetic algorithm for numerical optimization," *Natural Computing*, vol. 10, pp. 487-506, 2011.
- [30] Y.-W. Shang and Y.-H. Qiu, "A Note on the Extended Rosenbrock Function," *Evolutionary Computation*, vol. 14, pp. 119-126, 2006/03/01 2006.
- [31] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, pp. 1362-1381, 2009.
- [32] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, pp. 67-82, 1997.

Ting Yee Lim is currently a PhD student in the School of Computer Science at the Universiti Sains Malaysia (USM), Penang, Malaysia. She received her B.Sc. (Hons) and M.Sc. (Hons) degrees in Computer Science from the same university back in years 2001 and 2004, respectively. Her research interest focuses on Evolutionary Computation, in particular the Genetic Algorithms.

Ahamad Tajudin Khader is presently the Dean of Graduate Studies & Research at Universiti Sains Malaysia (USM), Penang, Malaysia. He holds a PhD degree from Strachclyde University, U.K and both BSc. (Hons) as well as M.Sc. (Hons) from Ohio, U.S.A. Assoc. Prof. Dr Ahamad Tajudin's research focuses on Genetic Algorithms, particularly in the areas of population modeling and parameter control. He is also actively involved in research on other evolutionary algorithms and problems in scheduling, timetabling as well as planning.