Soft Computing, July 2015

Kursawe Function Optimisation using Hybrid Micro Genetic Algorithm (HMGA)

Lim Wei Jer[1], Asral Bahari Jambek[1], and Neoh Siew Chin[2]

[1] School of Microelectronic Engineering,
Universiti Malaysia Perlis,
Kampus Alam, Pauh Putra, 02600 Pauh, Perlis, Malaysia
[2] Computational Intelligence Research Group,
Department of Computing Science and Digital Technologies,
Faculty of Engineering and Environment,
University of Northumbria, Newcastle, NE1 8ST, UK

limweijer1987@gmail.com, *asral@unimap.edu.my*, siew.neoh@northumbria.ac.uk

Abstract—A Hybrid Micro Genetic Algorithm (HMGA) is proposed for Pareto optimum search focusing on the Kursawe test function. HMGA is a fusion of the Micro Genetic Algorithm (MGA) and the elitism concept of Fast Pareto Genetic Algorithm (FPGA). The effectiveness of HMGA in Pareto optimal convergence was investigated with two performance indicators (i.e. Generational Distance (GD) and Spacing (SP)). To measure HMGA's performance, a comparison study was conducted between HMGA and MGA. In this work, HMGA outperformed MGA in the search for Pareto optimal front.

Index Terms—Optimisation, Kursawe test function, ZDT test function, hybrid algorithm

1. INTRODUCTION

Multi-objective optimisation problems (MOPs) involve the simultaneous optimisation of two or more objectives. In real-world applications, conflicting objectives are often found in MOPs where the optimisation of one objective degrades the performance of another. In a practical consideration of all objectives, Pareto-based optimisers suggest a set of optimal solutions rather than a single one. The proposed solution set is known as the *Pareto optimum set*, which comprise the "trade-off" or "good" compromises among the objectives. The minimisation or maximisation of an MOP [1] is posed as a general non-linear programming as follows:

$$\left.\begin{array}{lcc} \text{Min or Max} & f_k(x), & n = 1, \dots, N; \\ \text{Subject to} & & \\ \text{Inequality constraint} & g_m(x) \geq 0, & m = 1, \dots, M; \\ \text{Equality constraint} & h_j(x) = 0, & j = 1, \dots, J; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, \dots, K; \end{array}\right\} \quad (1)$$

where $\boldsymbol{x}$ represents the vector comprising decision variables $\boldsymbol{x_i}$**.** These decision variables are restricted to the feasible upper and lower bound limits given as $\boldsymbol{x_i^{(L)}}$ and $\boldsymbol{x_i^{(U)}}$ respectively. $\boldsymbol{f_k(x)}$ is the $\boldsymbol{k}^{\boldsymbol{th}}$ objective function, whereas $\boldsymbol{g_m(x)}$ and $\boldsymbol{h_j(x)}$ represent the inequality and equality

http://link.springer.com/article/10.1007/s00500-015-1767-5
DOI 10.1007/s00500-015-1767-5

constraints required for feasible solutions. When multiple objectives are involved, there is more than one objective function whereby **k** ≥ 2. Although these objectives are summed into a single objective problem (SOP) using the weighted-sum approach, this technique is more prone to the biases of weight vector and unit value implemented in each. To observe the optimum of each individual objective, researchers are focusing on the development of multi-objective evolutionary algorithms (MOEAs) based on the concept of Pareto dominance [2, 3].

2. BACKGROUND OF MOEAs

MOP differs from SOP in that several objectives must be optimised simultaneously subject to acceptable performance ranges over all objectives [4]. Therefore, MOEAs are useful for finding "trade-off" or solution sets rather than single solutions [5]. Genetic Algorithm (GA), Evolutionary Strategies (ES), Genetic Programming (GP) and Evolutionary Programming (EP) are the four main evolutionary paradigms [6], each having a different focus area. For instance, GA is the original form of evolutionary computation, where the evolutionary operators are used to change and improve a population of solutions to a problem [7]. ES and GP are similar to GA, but the former focuses more on the mutation activity [8], whereas the latter extends the genetic forms using tree and graph expressions [9]. Unlike the other three paradigms, EP considers the interaction of species rather than individuals.

In addition to the algorithms given above, heuristic algorithms, such as tabu search [10, 11], scatter search [12], swarm intelligence [13] and simulated annealing [14] are widely proposed to solve MOPs. The Niched-Pareto Genetic Algorithm (NPGA), Multi-Objective Genetic Algorithm (MOGA) and Non-dominated Sorting Genetic Algorithm (NSGA) are among the first generation MOEAs.

NSGA [15], proposed by Srivinas and Deb, uses a layer-based classification suggested by Goldberg [3], sharing dummy fitness values among the layers of non-dominated individuals to maintain population diversity. The population is ranked before the selection is performed. However, repetitive Pareto ranking decreases the efficiency of NSGA. NPGA [16] uses a Pareto dominance scheme in the tournament selection. Comparison is performed on two randomly chosen individuals whereby the non-dominated individual is always selected in the tournament and fitness sharing decides the result of the tournament if there is a tie in the comparison. In MOGA [17], a rank-based fitness assignment method is implemented wherein rank values within the population guide the selection procedure.

The second-generation MOEAs include the Strength Pareto Evolutionary Algorithm (SPEA), Pareto Archived Evolution Strategy (PAES), Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Strength Pareto Evolutionary Algorithm 2 (SPEA2).

SPEA [18] introduces an external population and preserves population diversity using a Pareto dominance relationship. The drawback of external population is the time-consuming search process and growing size of the external population. Thus, a pruning technique is applied to the external non-dominated population to maintain its size below a certain threshold. As for SPEA2 [19], it is enhanced over SPEA in three ways: 1) individual domination using fine-grained fitness assignment, 2) more precise guidance in the search via nearest neighbour density estimation, and 3) new archive truncation methods to preserve the boundary of the Pareto optimal set.

In PAES [20], a single parent generates a single offspring using an elitism-based archive approach. A crowding procedure is applied to maintain diversity in the Pareto optimal set.

Although there are some similarities between PAES and MGA, the addition of population in PAES is computationally expensive. To address this issue, MGA uses replaceable and non-replaceable memory to maintain population diversity [21].

NSGA-II [22] is an enhanced version of NSGA that uses elitism in (μ+λ) selection and a crowd comparison operator. Crowding distance is used to calculate the distance between the individual and its neighbour. NSGA-II prefers to select the non-dominated solution. If two solutions are in the same non-dominated rank, the less crowded region is preferred.

2.1 Pareto Terminology

The "good" result for MOPs is not a single solution but a set of solutions or "trade-offs" in the objective space [23]. Solutions proposed by the EA may not be the best or most satisfying due to incommensurability and the conflicting nature of the multiple objectives. Therefore, the Pareto concept was derived by researchers to determine and define the trade-off solutions in MOPs.

Pareto dominance: Solution vector $q = (q_1, q_2, \dots, q_a)$ is said to dominate another solution vector $v = (v_1, v_2, \dots, v_a)$ if and only if q is less than v for minimisation. The mathematical form of Pareto dominance is shown in Equation (2) where i is the number of vectors in solution space a.

$$\forall i \in \{1, \dots, a\}, q_i \le v_i \exists i \in \{1, \dots, a\}: q_i < v_i \quad (2)$$

2.2 Research Aims

In many real-world MOEA implementations, it can take minutes or hours to perform a single solution evaluation [24]. Some applications have potentially incommensurable ("black-box") objective functions that can increase computation complexity [25]. Therefore, it is often time consuming to run MOEAs for complicated problems. Few studies have applied the evolutionary algorithm to optimise and solve the real case problem in [26-28].

In this paper, we propose the HMGA, which is an extension of the MGA. The Kursawe test function [29] is used as the benchmark problem to evaluate the efficiency of the HMGA. The bootstrap method, which re-creates the relationship between the population and the sample by considering the sample as an epitome of the underlying population, is used to quantify the results. Therefore, the bootstrapped outcomes help us generalise our experimental results and project the stability of the proposed model. The results of the GD [30] and SP indicators [30] are compared with those of the MGA [21]. From the results, it is anticipated that our work will contribute towards formalizing an MOEA with better convergence capabilities in terms of GD and SP indicators while preserving the traditional MGA principles.

3 KURSAWE MOP

Kursawe's MOP, as proposed by [29], includes two objective functions. Equation (3) and (4) depict the formulation of the two objective functions $f_1(x)$ and $f_2(x)$. x_i represents the decision variable in the i dimension, and the limitation of variable x is shown in (4). Many researchers have tested the Kursawe function as a research benchmark [24, 31-33]. The Kursawe test function's true Pareto front, $\boldsymbol{P_{true}}$, is disconnected and asymmetric. The goal of optimisation is to assess the capability of MOEAs to approximate the true Pareto front of the Kursawe test function $\boldsymbol{P_{true}}$:

DOI 10.1007/s00500-015-1767-5

$$\min f_1(x) = \sum_{i=1}^{n-1}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2}))\ (3)$$

$$\min f_2(x) = \sum_{i=1}^{n}(|x_i|^{0.8} + 5sinx_i^3)\ (4)$$

$$where\ -5 \leq x_i \leq 5,\ \forall i = 1,2,3 \dots n\ (5)$$

4 MOEA INDICATORS

4.1 Generational Distance

In this experiment, the MOEA performance indicator known as GD [34-37] is proposed to determine how far the elements computed can approximate those in the optimal Pareto front. GD is defined as

$$Generational\ Distance = \frac{\sqrt{\sum_{i=1}^{j} d_i^2}}{j}.\ (6)$$

Note that $\boldsymbol{j}$ is the number of solutions in the optimal set, and $\boldsymbol{d_i}$ is the Euclidean distance between each solution, with the nearest member in the optimal Pareto front. As a result, the GD with the value of zero indicates that the real optimal Pareto front has been reached.

4.2 Spacing (SP)

SP is a measurement of the vector spread throughout the non-dominated solutions found in the Pareto front [38, 39]. Since the Pareto front is found after optimisation, the SP metric is suitable to judge how well the non-dominated solutions are distributed along the front. Schott [40] proposed the SP metric to measure the distance variance of neighbouring solutions in the Pareto front as shown in (13).

$$S \triangleq \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \qquad (7)$$

where $\boldsymbol{d_i} = min_j\left(\left|f_1^i(x) - f_1^j(x)\right| + \left|f_2^i(x) - f_2^j(x)\right|\right),\ i,j = 1,\dots,n,\ \boldsymbol{\bar{d}}$ is the mean of all $\boldsymbol{d_i}$, and $\boldsymbol{n}$ is the number of non-dominated solutions found so far. A value of zero for this SP indicates that all solutions in the Pareto front are spread evenly.

5 HYBRID MICRO GENETIC ALGORITHM (HMGA)

5.1 Crowding Distance

Crowding distance (**CDassignment**) is used to assign fitness for the first-rank non-dominated solutions. The crowding-distance assignment calculates the distance of two solutions on either side of a particular solution relative to each objective. As a result, the densities of the solutions surrounding any solution in the population can be estimated. The higher the crowding-distance value the less crowded the surrounding area of the solutions.

```
j = |ℑ|
for each i, set ℑ[i] distance = 0
for each objective m
 ℑ = sort (ℑ,m)
 ℑ [l]distance = ℑ [j]distance = ∞
 For i = 2 to (j-1)
 ℑ [i]distance = ℑ [i]distance + (ℑ [i+1]. m – ℑ [i-1].m)/( $f_m^{\max}$ – $f_m^{\min}$ )
```

Fig. 1. Crowding-distance pseudocode [22]

Fig. 1 shows the pseudocode [22] for calculating the crowding distance whereby j is the total number of individuals in $\Im$. First, the distance of each solution **i** of $\Im$ is initialized to zero. Then, solutions in $\Im$ are sorted according to each objective ***m***. For boundary points, infinite distance is set so that boundary points are always selected. **[i].*m*** refers to the **m**th objective function value of the **i**th individual and the maximum and minimum values of the **m**th objective function.

After the crowding-distance assignment, the non-dominated solutions are compared using distance tournament selection to give higher priority to less crowded solutions. Such action mainly aims to preserve the diversity of the Pareto front by selecting solutions with higher variation for the next generation when the size of the archive that stores non-dominated solutions exceeds the predefined size.

5.2 Strength Fitness Assignment

For each solution in the second rank, a comparison is conducted with all other solutions in the population, and fitness is assigned based on the solutions it dominates. The strength fitness (**STassignment**) concept from SPEA [19] is adopted to calculate the fitness for solutions in the second rank. Following the method proposed by [41], both dominated and dominating solutions are considered for each solution in the second rank. A net strength $\boldsymbol{S(x_i)}$ that indicates the number of solutions dominated by $\boldsymbol{x_i}$ is formulated as

$$S(x_i) - \left|\left\{x_j \middle| \forall x_j \ \in P \bigwedge x_i \succ x_j \bigwedge j \neq i\right\}\right|, \ (14)$$

meaning that solution $\boldsymbol{x_i}$ dominates solution $\boldsymbol{x_j}$.

From the net strength, each dominated solution in the second rank is assigned a fitness $\boldsymbol{F(x_i)}$ as

$$F(x_i) = \sum_{x_i \succ x_j} S(x_j) - \sum_{x_k \succ x_i} S(x_k)\, , \forall x_j, x_k \in \ P_t \bigwedge j \neq i \neq k \ \ (15)$$

where $\boldsymbol{F(x_i)}$ is equal to the subtraction of the total strength values of all solutions dominated by $\boldsymbol{x_i}$ and the total strength values of all solutions by which $\boldsymbol{x_i}$ is dominated.

By considering both dominating and dominated strengths, this approach preserves the diversity of the search wherein the chance of two solutions having the same fitness value is reduced. Therefore, three conditions are possible when solutions are compared:

1) In the situation where the solutions are chosen from two different ranks, the better rank is

preferred.

2) Two selected solutions differ in values, but in the same rank, the higher fitness value is preferred.
3) If two selected solutions have the same fitness and rank values, one of them will be selected randomly with equal probability.

By adopting the FPGA fitness assignments above, less computational time is required in the development of HMGA elitism because the strength fitness assignment only sets the fitness according to subtraction of the total dominating and dominated values of $\boldsymbol{x_i}$. There is also no complexity in the diversity preservation in the fitness assignment because the crowding-distance operation helps maintain the diversity along the Pareto optimal front where it assigns higher priority to a less crowded region.

5.3 Proposed HMGA

The HMGA is created by integrating the MGA concept proposed by Coello et al. [21] as the base and combining it with the proposed elitism from Fig. 2, explaining more details of the pseudocode of the HMGA.

By referring to the pseudocode in Fig. 2, a memory population **P** with size of **N** is randomly initialized and divided into two portion of memories (i.e. (1) replaceable memory **rm** and (2) non-replaceable memory **irm)**. Candidate solutions in the replaceable memory **rm** are substituted when the replacement condition is satisfied during generation while the candidate solutions in the non-replaceable memory **irm** are kept to maintain diversity during the evolutionary search. A small population (size **M** between 3 and 5) is selected randomly from both sides of the memories to represent an initial population **PP**. Genetic operators (selection, crossover, mutation) are applied to form a new generation **PPN** comprising the offspring after crossover and mutation and one elite from the PP. The insertion of one elite solution **elit** into the **PPN** preserves some information before the genetic reproduction. All the candidate solutions in **PPN** are now evaluated with HMGA elitism. During the first implementation of HMGA elitism, the external archive **y** is empty.

Therefore, the solutions are evaluated for Pareto ranking where non-dominated solutions are categorised in the first-rank and dominated solutions are grouped under the second rank. The first-rank solutions are then evaluated based on crowding distance (**CDassignment**), whereas the second-rank solutions are assessed based on strength assignment (**STassignment**). From the fitness, **M** number of elite solutions based on the priority of comparison in (**STassignment**) are taken as the parent solutions for the repetitive loop of genetic operations (selection, crossover, mutation), the creation of **PPN** and the HMGA elitism until nominal convergence is reached. The nominal convergence in this research is used to fix the number of evolutions. After nominal convergence is satisfied, two elite solutions (**elit**) are inserted into archive **y** and used to replace two contenders sequentially taken from the replaceable memory **irm**. As mentioned, this evolution is repeated until the condition for the overall replacement cycle is achieved. When the evolutionary cycle reaches the overall replacement cycle, all the collected elite solutions in archive **y** replace all the solutions in the replaceable memory **rm**. This provides a more intuitive search direction towards the evolution, as there is better chance that candidates from the best elite solutions will be selected for evolution in the next generation. After replacing all solutions in the replaceable memory, the HMGA elitism is again conducted until the termination criterion is met. In this research, the termination criterion refers to the maximum evolutionary cycle predetermined in the experiment. The maximum evolutionary cycle is fixed based on the cycle of convergence observed from a number of experiments.

In this research, the proposed HMGA model explained above is applied to the Kursawe test function. The populations of candidate solutions are generated according to the specific range and constraints of the test function. These solutions are then separated into replaceable and non-replaceable memory and optimised using the HMGA flow as given in Fig. 3. As multiple objectives are involved in these two studies, the value obtained for each objective is calculated based on the equations given in previous sections for the Kursawe test functions. The achievement of each objective is then used to categorise the solution as non-dominated (first-rank) or dominated (second-rank). Then, the evolution of optimisation is conducted using the proposed HMGA model in which the HMGA elitism is incorporated to evaluate the fitness of each solution. To match and compare with [21], the maximum iteration cycle is used as the termination criterion. The results of these benchmark studies based on the evaluation of our HMGA model are discussed in the next section.

```
Initialize parameter setting
t = 0
create initial random population P with size N.
P is separated into two portion rm and irm by ratio
i = 0
while i < maxevaluation do
  begin
    Get initial population PP from  P with size M
      repeat
        begin
        S  =  selection (PP)
        C  =  crossover (S)
        M  =  mutation (C)
        Keep one elit from PP into M
        Next generation PPN  =  M
      end
    until nominal convergence is reached
Copy two non-dominated solutions (elit) from PPN to archive
if archive is reached archivesize when trying to insert elit
  then adaptive grid (elit)
end if
Copy two elits from PPN to rm
if i mod replacement cycle
  then replace all rm memory with archive until rm is full
end if
i  =  i + 1
end while
```

Fig. 2. Pseudocode of proposed HMGA

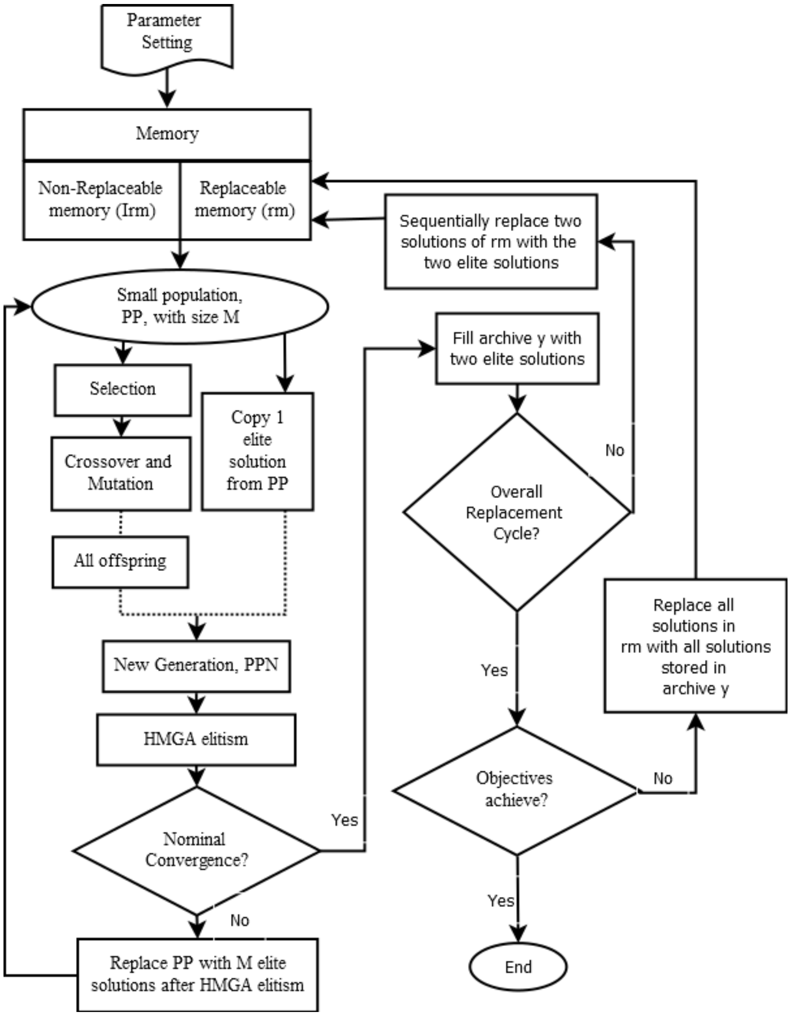
Parameter Setting
Memory
Non-Replaceable memory (Irm)
Replaceable memory (rm)
Sequentially replace two solutions of rm with the two elite solutions
Small population, PP, with size M
Fill archive y with two elite solutions
Selection
Copy 1 elite solution from PP
Crossover and Mutation
No
Overall Replacement Cycle?
All offspring
Replace all solutions in rm with all solutions stored in archive y
New Generation, PPN
Yes
HMGA elitism
Objectives achieve?
No
Yes
Nominal Convergence?
Yes
No
Replace PP with M elite solutions after HMGA elitism
End

Fig. 3. Proposed HMGA flow chart

6 RESULTS AND DISCUSSION

In this study, the experimental settings for the proposed HMGA are most similar to the MGA model in [21]. Table I shows the settings of evolutionary parameters in the HMGA and MGA models.

In Table I, the first column shows the parameters of the Kursawe function experiment and the second and third columns present the setting values for the MGA and HMGA respectively. The only setting difference is in elite size, as the MGA uses elite size 3 and our proposed HMGA uses elite size 2. The elite size is reduced to decrease computational complexity. As shown in Table I and the HMGA mechanism, the optimisation process begins with memory size 50, external memory size 100 and population size 4 for the evolution before reaching nominal convergence. The ratio of replaceable memory is 70%, and the crossover probability is 0.8 for both algorithms. Binary tournament selection is used as the selection method. The overall replacement cycle is activated every 25 evolutions such that all solutions in the replaceable memory are replaced with

DOI 10.1007/s00500-015-1767-5

the solutions stored in archive **y**. Two performance indicators (GD and SP) are used in this experiment.

TABLE I
MGA and HMGA Parameter Settings for Kursawe Test Function

Parameter of Kursawe Function Experiment	MGA value [21]	HMGA value
Population Size, M	4	4
Maximum Iteration	12000	12000
Memory Size, N	50	50
Ratio of Replaceable Memory	70%	70%
External Memory Size	100	100
Bi-sections for Adaptive Grid	5	5
Nominal Convergence	2	2
Probability of Two-Point Crossover	0.8	0.8
Probability of Uniform Mutation	x=1	x=1
Perturbation Index of Uniform Mutation	1	1
Selection Method	Binary Tournament	Binary Tournament
Replacement Cycle	25	25
Elite Size	3	2
MOEA Indicator	GD, SP	GD, SP

One-hundred experiments have been conducted on the Kursawe test function to evaluate the performance of the proposed HMGA. To achieve a more comprehensive comparison between the HMGA and MGA, the best, worst, mean, median and standard deviation values obtained from the performance indicators are included in the analysis. As mentioned in section 4, the smaller the GD value the closer the result to the optimum. Table II presents the comparison between the MGA and HMGA in terms of GD. The MGA result reported in [21] is presented in Table II where the best, worst, mean, median and standard deviation values are reported as 0.00680344, 0.0103437, 0.008456311, 0.008489235 and 0.00098659, respectively. The statistical values for HMGA in Table II are calculated based on the accumulated results of 100 experiments (Table I). The statistical bound of the HMGA with the best and worst GD (ranging 0.00032050–0.00128554) are lower than the reported MGA's best GD (0.00680344). Thus, the HMGA shows better performance than the MGA in terms of GD. The HMGA has a mean 94%-increment advantage over the MGA, which indicates a more effective search mechanism.

DOI 10.1007/s00500-015-1767-5

TABLE II
Generational Distance

GD	MGA [21]	HMGA
Best	0.00680344	0.00032050
Worst	0.0103437	0.00128554
Mean	0.008456311	0.00054287
Median	0.008489235	0.00051027
Standard Deviation	0.00098659	0.00015325

TABLE III
Spacing (SP)

SP	MGA [21]	HMGA
Best	0.0716859	0.01355709
Worst	0.203127	0.03337811
Mean	0.12889499	0.02138230
Median	0.126655	0.02130097
Standard Deviation	0.02993154	0.00283567

The Kursawe test function-indicator analysis for SP was also carried out on both the MGA and HMGA (Table III). Importantly, the statistical bound of MGA fluctuates significantly, ranging 0.203127–0.0716859, whereas the more stable SP values for the HMGA range 0.03337811–0.01355709. There is no dramatic difference between the best and worst SP values, and the standard deviation of the HMGA is comparatively smaller than that of the MGA. Therefore, the HMGA's performance throughout the 100 conducted experiments is consistent. Observations from the SP values prove that the HMGA results distribute more evenly compared to the results of the MGA.

7 CONCLUSION

The experiment results for the Kursawe test function indicate that the HMGA directs the search towards the Pareto optimal front effectively. Two indicators (GD and SP) are used to evaluate the performance of the HMGA in Kursawe test function. The results show that the HMGA outperforms the MGA and overcomes the difficulties in the test function. The crowding-distance selection used in the HMGA is most likely a main factor in the HGMA's more even distribution of solutions along the Pareto optimal front. The HMGA is a potential evolutionary algorithm for MOPs.

ACKNOWLEDGMENTS

This research was funded by a Knowledge Transfer Program (KTP) Grant in collaboration with Unimap and Myreka Sdn Bhd.

REFERENCES

1. Deb, K., *Recent Developments in Evolutionary Multi-Objective Optimization*, in *Trends in Multiple Criteria Decision Analysis*, M. Ehrgott, J.R. Figueira, and S. Greco, Editors. 2010, Springer US. p. 339-368.
2. Forrest, S., *Genetic algorithms: principles of natural selection applied to computation.* Science, 1993. **261**(5123): p. 872-878.
3. Goldberg, D.E. and J.H. Holland, *Genetic algorithms and machine learning.* Machine learning, 1988. **3**(2): p. 95-99.
4. Coello, C.C. and G.B. Lamont, *An introduction to multi-objective evolutionary algorithms and their applications.* Applications of Multi-Objective Evolutionary Algorithms, 2005: p. 1-28.
5. Deb, K., *Introduction to Evolutionary Multiobjective Optimization*, in *Multiobjective Optimization*, J. Branke, et al., Editors. 2008, Springer Berlin Heidelberg. p. 59-96.
6. Lipinski, P., *Practical applications of evolutionary computation to financial engineering: robust techniques for forecasting, trading, and hedging (iba, h. and aranha, c.c.; 2012)[book review]*. Computational Intelligence Magazine, IEEE, 2012. **7**(2): p. 75-76.
7. Holland, J.H., *Adaptation in Natural and Artificial Systems. The University of Michigan Press.* 1975.
8. Eiben, A.E. and T. Bäck, *Empirical investigation of multiparent recombination operators in evolution strategies.* Evolutionary Computation, 1997. **5**(3): p. 347-365.
9. Iba, H. and C. Aranha, *Introduction to Genetic Algorithms*, in *Practical Applications of Evolutionary Computation to Financial Engineering*. 2012, Springer Berlin Heidelberg. p. 1-17.
10. Carcangiu, S., A. Fanni, and A. Montisci, *Multiobjective Tabu Search Algorithms for Optimal Design of Electromagnetic Devices.* Magnetics, IEEE Transactions on, 2008. **44**(6): p. 970-973.
11. Ho, S.L., et al., *A tabu method to find the Pareto solutions of multiobjective optimal design problems in electromagnetics.* Magnetics, IEEE Transactions on, 2002. **38**(2): p. 1013-1016.
12. Nebro, A.J., et al., *AbYSS: Adapting Scatter Search to Multiobjective Optimization.* Evolutionary Computation, IEEE Transactions on, 2008. **12**(4): p. 439-457.
13. Dasheng, L., et al., *A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2007. **37**(1): p. 42-50.

14. Smith, K.I., et al., *Dominance-Based Multiobjective Simulated Annealing.* Evolutionary Computation, IEEE Transactions on, 2008. **12**(3): p. 323-342.
15. Srinivas, N. and K. Deb, *Muiltiobjective optimization using nondominated sorting in genetic algorithms.* Evolutionary Computation, 1994. **2**(3): p. 221-248.
16. Horn, J., N. Nafpliotis, and D.E. Goldberg. *A niched Pareto genetic algorithm for multiobjective optimization*. in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. 1994.
17. Fonseca, C.M. and P.J. Fleming. *Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization*. in *ICGA*. 1993.
18. Zitzler, E. and L. Thiele, *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach.* Evolutionary Computation, IEEE Transactions on, 1999. **3**(4): p. 257-271.
19. Zitzler, E., M. Laumanns, and L. Thiele, *SPEA2: Improving the strength Pareto evolutionary algorithm*. 2001, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK).
20. Knowles, J.D. and D.W. Corne, *Approximating the nondominated front using the Pareto archived evolution strategy.* Evolutionary Computation, 2000. **8**(2): p. 149-172.
21. Coello, C.A.C. and G. Pulido, *Multiobjective structural optimization using a microgenetic algorithm.* Structural and Multidisciplinary Optimization, 2005. **30**: p. 388–403.
22. Deb, K., et al., *A fast and elitist multiobjective genetic algorithm: NSGA-II.* Evolutionary Computation, IEEE Transactions on, 2002. **6**(2): p. 182-197.
23. Hwang, C.-L. and A.S.M. Masud, *Multiple Objective Decision Making — Methods and Applications : A State-of-the-Art Survey*. Lecture Notes in Economics and Mathematical Systems. 1979: Springer Berlin Heidelberg.
24. Bartz-Beielstein, T., et al. *Particle swarm optimizers for Pareto optimization with enhanced archiving techniques*. in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*. 2003: IEEE.
25. Knowles, J., *ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems.* Evolutionary Computation, IEEE Transactions on, 2006. **10**(1): p. 50-66.
26. Bastos-Filho, C.J.A., et al., *Wavelength Assignment for Physical-Layer-Impaired Optical Networks Using Evolutionary Computation.* Optical Communications and Networking, IEEE/OSA Journal of, 2011. **3**(3): p. 178-188.
27. Precup, R.-E., et al., *Gravitational Search Algorithm-Based Tuning of Fuzzy Control Systems with a Reduced Parametric Sensitivity*, in *Soft Computing in Industrial Applications*, A. Gaspar-Cunha, et al., Editors. 2011, Springer Berlin Heidelberg. p. 141-150.
28. Lughofer, E., *A dynamic split-and-merge approach for evolving cluster models.* Evolving Systems, 2012. **3**(3): p. 135-151.
29. Kursawe, F., *A variant of evolution strategies for vector optimization*, in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Männer, Editors. 1991, Springer Berlin Heidelberg. p. 193-197.
30. Durillo, J.J. and A.J. Nebro, *jMetal: A Java framework for multi-objective optimization.* Advances in Engineering Software, 2011. **42**(10): p. 760-771.

31. Raquel, C.R. and P.C. Naval Jr. *An effective use of crowding distance in multiobjective particle swarm optimization*. in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. 2005: ACM.

32. Wang, J., et al. *Multi-Objective Particle Swarm Optimization Based on Self-Update and Grid Strategy*. in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*. 2013: Springer.

33. Leung, M.-F., et al. *A new strategy for finding good local guides in MOPSO*. in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. 2014: IEEE.

34. Van Veldhuizen, D.A. and G.B. Lamont, *Multiobjective evolutionary algorithm research: A history and analysis*. 1998, Citeseer.

35. Van Veldhuizen, D.A. and G.B. Lamont. *On measuring multiobjective evolutionary algorithm performance*. in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. 2000: IEEE.

36. Pulido, G.T. and C.A.C. Coello. *Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer*. in *Genetic and Evolutionary Computation–GECCO 2004*. 2004: Springer.

37. Yang, F.-C. and B. Ni. *Water Flow-Like Optimization Algorithm for Multi-objective Continuous Optimization Problem*. in *Proceedings of 2013 4th International Asia Conference on Industrial Engineering and Management Innovation (IEMI2013)*. 2014: Springer.

38. Baesler, F. and C. Palma, *Multiobjective parallel machine scheduling in the sawmill industry using memetic algorithms.* The International Journal of Advanced Manufacturing Technology, 2014. **74**(5-8): p. 757-768.

39. Maheta, H.H. and V.K. Dabhi. *An improved SPEA2 Multi objective algorithm with non dominated elitism and Generational Crossover*. in *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*. 2014.

40. Schott, J.R., *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. 1995, DTIC Document.

41. Eskandari, H. and C. Geiger, *A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems.* Journal of Heuristics, 2008. **14**(3): p. 203-241.