J Heuristics (2007) 13:1–34
DOI 10.1007/s10732-006-9003-1

Effects of diversity control in single-objective and multi-objective genetic algorithms

Nachol Chaiyaratana · Theera Piroonratana · Nuntapon Sangkawelert

Received: 12 August 2003 / Revised: 23 April 2005 / Accepted: 24 January 2006 / Published online: 21 December 2006

Abstract This paper covers an investigation on the effects of diversity control in the search performances of single-objective and multi-objective genetic algorithms. The diversity control is achieved by means of eliminating duplicated individuals in the population and dictating the survival of non-elite individuals via either a deterministic or a stochastic selection scheme. In the case of single-objective genetic algorithm, onemax and royal road R_1 functions are used during benchmarking. In contrast, various multi-objective benchmark problems with specific characteristics are utilised in the case of multi-objective genetic algorithm. The results indicate that the use of diversity control with a correct parameter setting helps to prevent premature convergence in single-objective optimisation. Furthermore, the use of diversity control also promotes the emergence of multi-objective solutions that are close to the true Pareto optimal solutions while maintaining a uniform solution distribution along the Pareto front.

Keywords Benchmarking · Diversity control · Multi-objective genetic algorithm · Single-objective genetic algorithm

N. Chaiyaratana (✉)
Research and Development Center for Intelligent Systems, King Mongkut's Institute of Technology North Bangkok, 1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand
e-mail: nchl@kmitnb.ac.th

T. Piroonratana
Department of Production Engineering, King Mongkut's Institute of Technology North Bangkok, 1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand

N. Sangkawelert
Department of Electrical Engineering, King Mongkut's Institute of Technology North Bangkok, 1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand

1 Introduction

It is undeniable that a major factor that contributes to the success of genetic algorithms in the field of optimisation is the parallel search mechanism embedded in the algorithm itself. However, this does not prevent the occurrence of premature convergence in the situation when the similarity among individuals in the population becomes too high. Similar to other optimisation techniques, a premature convergence of the genetic algorithm would lead to the discovery of "false" solutions. As a result, the prevention of a premature convergence must also be considered during the genetic algorithm design. One of the direct approaches for achieving the necessary prevention is to maintain diversity within the population (Mauldin, 1984).

Various strategies can be used to maintain or increase the population diversity. Nonetheless, a modification on the selection operation for the task has received much attention. For instance, Mori et al. (1995) has introduced a notion of thermodynamical genetic algorithm where the survival of individuals is regulated by means of monitoring the free energy within the population. The modification on the selection operation can also be done in the cross-generational sense (Whitley, 1989; Eshelman, 1991; Shimodaira, 1996). Whitley (1989) has proposed a GENITOR system where offspring generated by standard operators are chosen for replacing parents based upon the ranks of the individuals. In contrast to Whitley (1989), Eshelman (1991) recommends the application of mating restriction while Shimodaira (1996) suggests the use of variable-rate mutation as a means to create offspring. Then a cross-generational survival selection is carried out using a standard fitness-based selection technique in both cases. In addition to the strategies involving a direct modification of selective pressure, population diversity can also be maintained by transforming the optimisation problem of interest into a multi-objective optimisation problem (Abbass and Deb, 2003; Jensen, 2003). This is achieved by attempting to obtain multiple optimal solutions, which satisfy both the original objective and a conflicting augmented objective. Since these solutions are optimal in the sense that no improvement can be achieved in one objective that does not lead to degradation in the remaining objective, each solution would represent different trade-off among two conflicting objectives. In order to obtain all these optimal solutions, the diversification of the population are subsequently achieved during the execution of the genetic algorithm. In other words, the strategy described would also lead to an indirect manipulation of selective pressure.

In addition to the early works described above, another genetic algorithm has been specifically developed by Shimodaira (1997) to handle the issue of population diversity; this algorithm is called a diversity control oriented genetic algorithm or DCGA. Similar to most genetic algorithms, offspring in the DCGA are generated using standard crossover and mutation operators. However, during the cross-generational survival selection, duplicated individuals in the merged population containing both parent and offspring individuals are first eliminated. Then, the remaining individuals are selected based on either the associate fitness or the consideration on both the fitness and the genomic similarity between the interested individual and the elite individual. The performance of the DCGA has been benchmarked using various test problems (Shimodaira, 2001). Although the algorithm appears to perform well in all test problems used, the published results are achieved after tuning the diversity control operator until the optimal setting is identified. In other words, the effects of different diversity control settings on the same problem are not discussed in the work. Furthermore, the benchmark problems at which the fitness of an individual is calculated directly from its genomic structure e.g. onemax (Ackley, 1987) and royal road (Mitchell et al., 1992) functions have not been extensively discussed in the original study. This also presents an interesting research question since the performance of genetic algorithms for this class of problems would be independent from the chromosome encoding technique and only depends on the genetic operators used. Generally, in the case of benchmark problems that involve continuous decision variables, the use of different chromosome encoding schemes would lead to different search performances.

With a minor modification, the DCGA can also be used in multi-objective optimisation. One possible approach for achieving this is to integrate the DCGA with other genetic algorithms that are specifically designed for multi-objective optimisation such as a multi-objective genetic algorithm or MOGA (Fonseca and Fleming, 1993, 1995, 1998). Such approach has been investigated by Sangkawelert and Chaiyaratana (2003) where the inclusion of cross-generational survival selection with the multi-objective genetic algorithm is equivalent to the use of elitism, which is proven to be crucial to the success of various multi-objective algorithms including a non-dominated sorting genetic algorithm II or NSGA-II (Deb et al., 2002a) and an improved strength Pareto evolutionary algorithm or SPEA-II (Zitzler et al., 2002). In addition, the similarity measurement between the non-elite individual and the elite individual required by the diversity control operator is still carried out in the genotypic space. The resulting combined algorithm, which can be uniquely referred to as a multi-objective diversity control oriented genetic algorithm or MODCGA has been successfully tested using a two-objective benchmark suite (Zitzler et al., 2000). Although some insights into the behaviour of the MODCGA have been gained through the benchmark trial by Sangkawelert and Chaiyaratana (2003), further studies can be made and are required. In particular, the initial study of the MODCGA is conducted with a similarity measurement between two individuals being carried out in genotypic space. However, in multi-objective optimisation the trade-off surface, which is the direct result from the spread of solutions, is generally defined in objective space. This means that diversity control can also be achieved by considering the similarity between objective vectors of the individuals. Moreover, in the initial study the multi-objective benchmark problems contain only two objectives. The performance study where benchmark problems contain a higher number of objectives (Deb et al., 2002b) should also be investigated.

In summary, the effects of diversity control in single-objective and multi-objective genetic algorithms will be studied in the following manners. In the case of single-objective genetic algorithm, various settings of the diversity control operator for the benchmark problems where the fitness of an individual is calculated directly from the genomic structure will be explored. On the other hand, in the context of multi-objective genetic algorithm a study on different settings of the diversity control operator where the similarity measurement between individuals is carried out in the objective space for benchmark problems with two and three objectives will be conducted.

The organisation of this paper is as follows. In Section 2, the explanation of the original DCGA and how it can be modified to cope with multi-objective optimisation problems is given. In Section 3, the single-objective benchmark problems and performance evaluation criteria are explained while the multi-objective problems and evaluation criteria are discussed in Section 4. Next, the single-objective benchmarking results of the DCGA are illustrated in Section 5. Following that, the multi-objective benchmarking results of the modified MODCGA are given in Section 6. Finally, the conclusions are drawn in Section 7.

2 DCGA and its extension

The original DCGA developed by Shimodaira (1997) can only be used to solve single-objective optimisation problems. However, the algorithm can be easily combined with other genetic algorithms. For instance, the DCGA has been successfully integrated with a co-operative co-evolutionary genetic algorithm or CCGA (Potter and De Jong, 2000) for use in container loading optimisation (Pimpawat and Chaiyaratana, 2004) and optimal control of a hysteresis system (Boonlong et al., 2004). Furthermore, the DCGA has also been integrated with a multi-objective genetic algorithm or MOGA (Fonseca and Fleming, 1993) for use in continuous and discrete benchmark problems (Sangkawelert and Chaiyaratana, 2003), time-optimal path planning and control in a robotic system (Weerayuth and Chaiyaratana, 2002) and optimal satellite attitude control (Boonlong et al., 2002). The extension of the DCGA for use in multi-objective optimisation that will be used throughout this paper also involves the integration between the DCGA and MOGA. However, in contrast to the previous work by Sangkawelert and Chaiyaratana (2003) where the similarity measurement between individuals is conducted in genotypic space, in this work the measurement will be carried out in objective space. Detailed explanation of the DCGA, MOGA and algorithm integration is given as follows.

2.1 Diversity control oriented genetic algorithm

The diversity control oriented genetic algorithm (DCGA) was first introduced by Shimodaira (1997). Similar to other single-objective steady-state genetic algorithms, the parent population and the offspring population are merged together during the DCGA run where the appropriated individuals are extracted from the merged population. However, instead of selecting the highly fit individuals from the population straightaway, the extraction process in the DCGA starts with the elimination of duplicated individuals in the merged population. The remaining individuals are then sorted according to their fitness values in descending order. Following that the best individual from the remaining individuals is determined and kept for passing onto the next generation. Then either a cross-generational deterministic survival selection (CDSS) method or a cross-generational probabilistic survival selection (CPSS) method is applied in the top-down fashion to the remaining non-elite individuals in the sorted array. In the case of the CDSS, the remaining non-elite individuals with high fitness values will have a higher chance of being selected since they reside in the top part of the array and hence have a higher selection priority than individuals with low fitness values. In contrast, a survival probability value is assigned to each non-elite individual according to its similarity to the best individual in the case of the CPSS. This survival probability is given by

$$p_s = \{(1 - c)d_h/L + c\}^{\alpha} \tag{1}$$

where p_s denotes the survival probability, d_h is the Hamming distance between the interested individual and the best individual, L is the chromosome length, c is the shape coefficient and α is the exponent coefficient. With this form of survival probability assignment, if the genomic structure of the individual interested is very close to that of the best individual, the survival probability assigned to this individual will be close to zero. On the other hand, if the chromosome structure of this individual is quite different from that of the best individual, its survival probability will be close to one. Each individual will then be selected according to the assigned survival probability where the survival selection of the sorted non-elite individuals is still carried out in the top-down manner. With the use of sorted individual array, the selection of non-elite individuals will depend entirely on the assigned fitness in the CDSS scheme. On the other hand, a decision either to select or not to select an individual according to the CPSS scheme depends on both the assigned fitness and the survival probability. Basically, the individual that have a high chance of being selected must possess high fitness and have a genomic structure that is quite different from that of the best individual. This also means that both a highly fit individual that is quite resemble to the best individual and a mediocre individual that is different from the best individual would not have a high chance of being picked. If the total number of all selected individuals including the pre-selected elite individual does not reach the required population size after the survival selection loop, randomly generated individuals will be added to the individual array until the required number is met. A flow chart of the DCGA with CDSS and CPSS schemes is displayed in Fig. 1. A comprehensive description of the DCGA and its benchmarking performance in various continuous test problems can be found in Shimodaira (2001).

2.2 Multi-objective genetic algorithm

The multi-objective genetic algorithm (MOGA) was first introduced by Fonseca and Fleming (1993). The MOGA functions by seeking to optimise the components of a vector-valued objective function. Unlike single-objective optimisation, the solution to a multi-objective optimisation problem is a family of points known as the Pareto optimal set. Each point in the set is optimal in the sense that no improvement can be achieved in one component of the objective vector that does not lead to degradation in at least one of the remaining components. Given a set of possible solutions, a candidate solution is said to be Pareto optimal if there are no other solutions in the solution set that can dominate the candidate solution. In other words, the candidate solution would be a non-dominated solution. Assuming, without loss of generality, a minimisation problem, an m-dimensional cost vector $\mathbf{u}$ is said to be dominating another m-dimensional cost vector $\mathbf{v}$ if, and only if, $\mathbf{u}$ is partially less than $\mathbf{v}$ ($\mathbf{u}\ p < \mathbf{v}$), i.e.

$$\mathbf{u}\ p < \mathbf{v} \leftrightarrow \forall i = 1, \ldots, m : u_i \leq v_i \wedge \exists i = 1, \ldots, m : u_i < v_i. \tag{2}$$

By identifying the number of solutions in the solution set that dominate the solution of interest, a rank value can be assigned to the solution. In other words, the rank of a candidate solution is given by the number of solutions in the solution set that dominate the candidate solution. After a rank has been assigned to each solution, a fitness value can then be interpolated onto the solution where a genetic algorithm can subsequently be applied in the optimisation procedure. Since the aim of a search

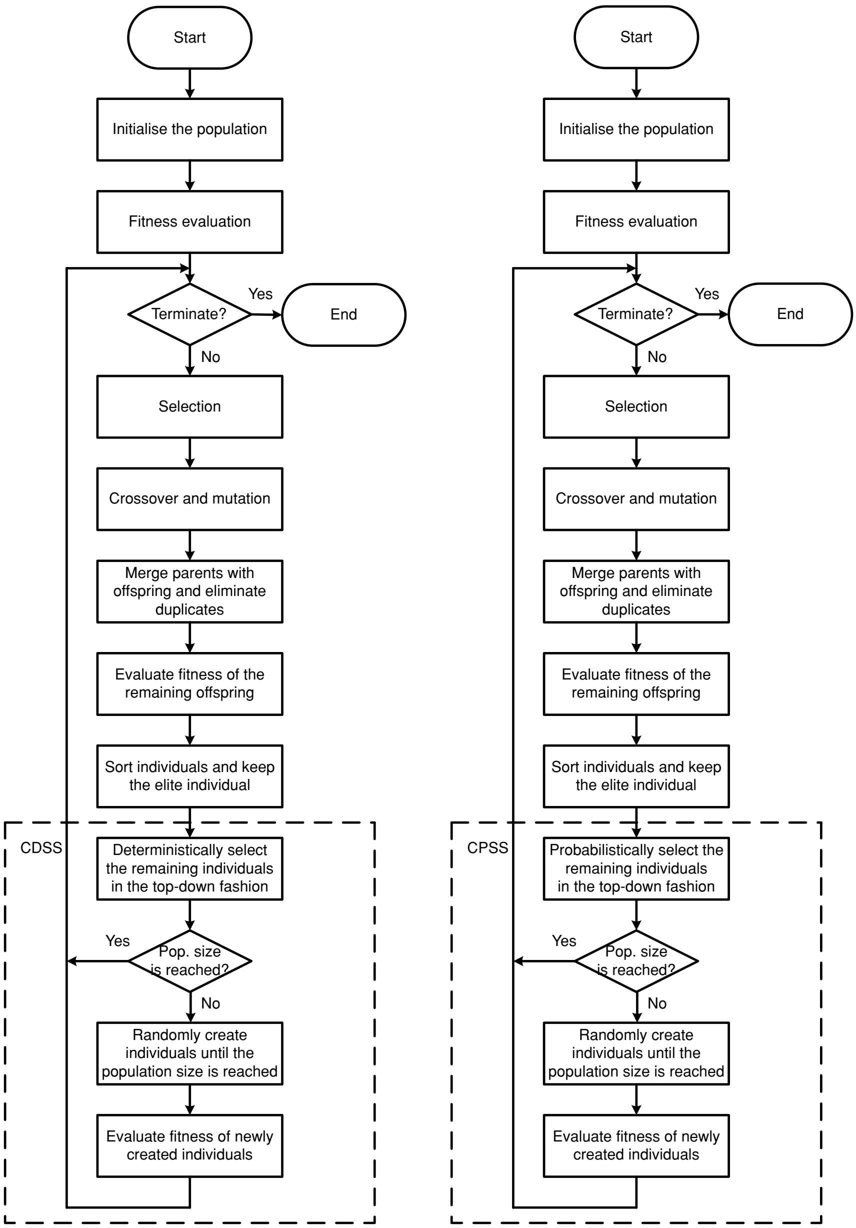
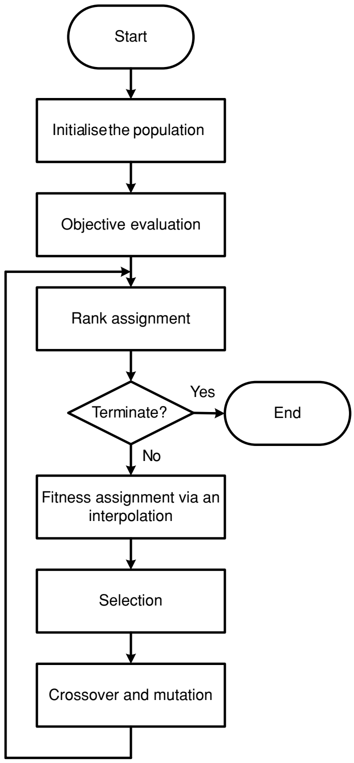


Fig. 1 Flow chart of the DCGA with CDSS and CPSS schemes

by the MOGA is to locate Pareto optimal solutions, in essence the multi-objective optimisation problem has also been treated as a multi-modal problem. Hence, the use of additional genetic operators including the fitness sharing and mating restriction procedures is also required. However, in addition to the usual application of the fitness sharing and mating restriction procedures in the decision variable space (Fonseca

Fig. 2 Flow chart of the MOGA



and Fleming, 1995), they can also be carried out in the objective space (Fonseca and Fleming, 1993). A summary of the MOGA is given in the form of a flow chart as depicted in Fig. 2. A comprehensive description of the MOGA, which covers other advanced topics including goal attainment and priority assignment strategies, can be found in Fonseca and Fleming (1998).

2.3 Genetic algorithm integration

By combining the MOGA and the DCGA together, the resulting algorithm can be referred to as a multi-objective diversity control oriented genetic algorithm or MODCGA. Similar to the MOGA, the rank of each individual will be obtained after comparing it with the remaining individuals. However, the comparison will be made among individuals in the merged population, which is the result from combining parent and offspring populations together. Since the best individuals in the MOGA are the non-dominated individuals, in the case where the CPSS method is used there will be more than one survival probability value that can be assigned to each dominated individual. In this study, the lowest value in the probability value set is chosen for each dominated individual. After the survival selection routine is completed and the fitness values have been interpolated onto the individuals, the standard genetic operations can then be applied to the population in the usual way. In the early work by Sangkawelert and Chaiyaratana (2003), a similarity measurement between dominated and non-dominated individuals, which leads to the survival probability assignment, is carried out in genotypic space. In this work, the similarity measurement will be conducted in objective space instead; two advantages are gained through this modification. Firstly, since the aim of multi-objective optimisation is to obtain multiple solutions at which together produce a trade-off objective surface that represents a Pareto front, diversity control in objective space would directly enforce this aim. Secondly, a diversity control operator that is designed for use in objective space would be independent of the chromosome encoding scheme utilised. Recent investigation into multi-objective optimisation using genetic algorithms usually involves problems with large number of decision variables (Zitzler et al., 2000; Deb et al., 2002a; Zitzler et al., 2002). The use of a binary representation would lead to an excessively long chromosome and hence degrades the algorithm performance. As a result, real-value chromosome encoding is generally employed instead. With the modification described above, in the case of CPSS scheme the survival probability as given in Eq. (1) will change to

$$p_s = \{(1 - c)d/d_{\max} + c\}^{\alpha} \tag{3}$$

where d is the distance between the interested individual and a non-dominated individual in objective space and $d_{\max}$ is the maximum distance between two individuals in the population. In order to distinguish between the early work by Sangkawelert and Chaiyaratana (2003) and the present work, the MODCGA where the similarity measurement is done in objective space will be referred throughout this paper as the MODCGA-II. In this investigation, the fitness sharing strategy utilised in the MODCGA-II is similar to the one described in Fonseca and Fleming (1993) where the fitness sharing is carried out in objective space.

In addition to the modification on the diversity control operation, the use of a preserved non-dominated solution archive is included in the MODCGA-II. Basically, the parent individuals will be picked from a population which includes both individuals obtained after the diversity control and that from the archive. Each time that a new population is created after the diversity control operation, non-dominated solutions within the archive will be updated. If the solution that survives the diversity control operation is neither dominated by any solutions in the archive nor a duplicate of a solution in the archive, then this solution will be added to the archive. At the same time, if the solution that survives the diversity control operation dominates any existing solution in the archive, the dominated solution will be expunged from the archive. In order to maintain the diversity within the preserved non-dominated solution archive, k-nearest neighbour clustering technique (Zitzler et al., 2002) is used to regulate the size of the archive. A summary of the MODCGA-II is illustrated in a flow chart as depicted in Fig. 3. In Fig. 3 the fitness evaluation for the newly created individuals within the CDSS/CPSS parts is replaced with the objective evaluation.

3 Single-objective benchmark problems and performance evaluation criteria

As mentioned earlier in Section 1, the extension on single-objective benchmarking of the DCGA will be carried out in this paper. Specifically, the test functions that will be

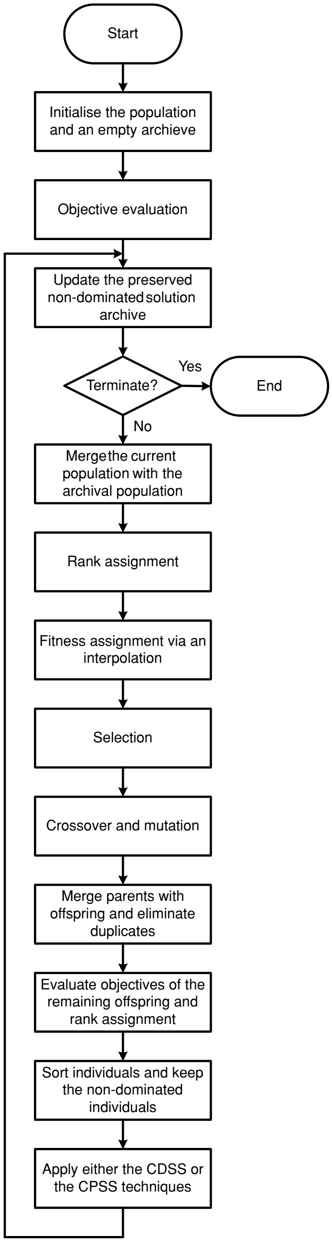


Fig. 3 Flow chart of the MODCGA-II

used here are onemax (Ackley, 1987) and royal road R_1 (Mitchell et al., 1992) functions. These two functions are chosen since in the original benchmarking results of the DCGA reported in Shimodaira (2001), the trial on functions at which the fitness is calculated directly from the bit pattern in the chromosome representation has not been extensively discussed. This class of functions is important to any benchmark trials since the performance of genetic algorithms for this type of functions will depend on the genetic operators and not on the chromosome encoding techniques. In general, a change in the chromosome encoding techniques would subsequently lead to a change in the performances of genetic algorithms in continuous optimisation problems. This sometimes makes a comparison between the currently investigated works and previously published results a difficult task. In addition, these two functions are chosen specifically since the function size and difficulty level can be scaled easily. Detailed description of these two functions follows.

For an individual, which is represented by an L-bit binary chromosome, the fitness of this individual for the onemax function is given by

$$\text{fitness} = \sum_{i=1}^{L} b_i \tag{4}$$

where b_i is the ith bit on the chromosome. The optimal solution is the solution that the value of every bit in the chromosome is equal to one and the corresponding fitness is L. On the other hand, for an individual, which is represented by an $8t$-bit binary chromosome, the fitness of this individual for the royal road R_1 function is given by

$$\text{fitness} = \sum_{i=1}^{8} \delta_i o(s_i) \tag{5}$$

where

$$\delta_i = \begin{cases} 1 & \text{if the chromosome contains schema } s_i \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

and s_i is the ith schema for the royal road R_1 function and $o(s_i)$ is the order of schema s_i, which has the value equal to t. Under a royal road $R_1(8t, t)$ function, an $8t$-bit chromosome is rewarded if it is an instance of eight non-overlapping order t, defining length t, schemata. Figure 4 illustrates eight schemata that are required to construct the optimal solution of the royal road $R_1(64, 8)$ function. From Fig. 4, the optimal solution is the solution that every bit is one and the corresponding fitness is 64. This solution is optimal since the solution contains all eight schemata where each schema has the order of eight. From Eq. (5), possible fitness values of an individual for a royal road $R_1(8t, t)$ function are $0, t, 2t, \ldots, 8t$.

```
S1 = 11111111********************************************************
S2 = ********11111111************************************************
S3 = ****************11111111****************************************
S4 = ************************11111111********************************
S5 = ********************************11111111************************
S6 = ****************************************11111111****************
S7 = ************************************************11111111********
S8 = ********************************************************11111111
```

Fig. 4 Eight schemata for the royal road $R_1(64, 8)$ function

For each test function, five problems with different chromosome lengths are explored. The fitness distance correlation coefficient (Jones and Forrest, 1995) of each problem is schematically displayed in Fig. 5. According to Jones and Forrest (1995), a benchmark problem that the fitness distance correlation coefficient is obtainable can be categorised as being either (a) misleading, or (b) difficult, or (c) straightforward. When the fitness distance correlation coefficient (r) of a problem is greater than or equal to 0.15 ($r \geq 0.15$), the problem is said to be misleading. It implies that the fitness of an individual tends to increase with distance from the global optimum. On the other hand, in a difficult test problem, the range of the fitness correlation coefficient is given by $-0.15 < r < 0.15$. For this type of problem, there is very little correlation between fitness and distance from the global optimum. In contrast, the fitness distance correlation is given by $r \leq -0.15$ in a straightforward problem. This means that the fitness of an individual tends to increase as the global optimum is approached. From Fig. 5, it is noticeable that the royal road $R_1(112, 14)$, $R_1(96, 12)$ and $R_1(80, 10)$ problems are difficult problems while the royal road $R_1(64, 8)$ and $R_1(48, 6)$ problems are straightforward problems. In contrast, all five onemax problems are straightforward problems. Although, the fitness distance correlation coefficient of the onemax function does not increase with the change on the chromosome length, the difficulty level of the function still increases with the problem size when considering the ratio between the number of optimal solutions and the total number of possible solutions.

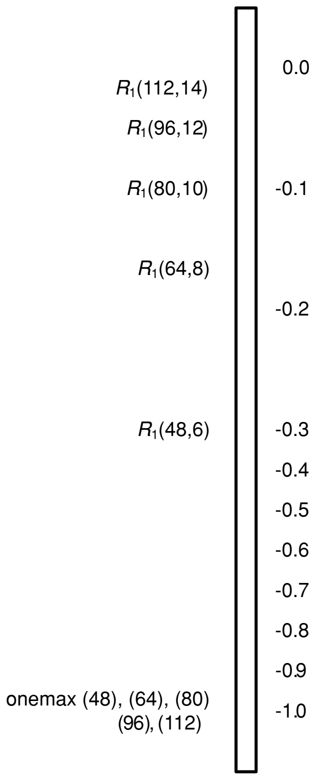


Fig. 5 Fitness distance correlation coefficients for the onemax and royal road R_1 functions

Table 1 Performance indices for single-objective benchmarking

Index	Description
SCR	Success rate (number of successful runs/total number of runs)
AVFE	Average value of function evaluation numbers in the successful runs
SDFE	Standard deviation of function evaluation numbers in the successful runs
AVG	Average value of generation numbers required in the successful runs
SDG	Standard deviation of generation numbers required in the successful runs
AVBF	Average value of the best fitness values in all runs (display in percentage of the optimal fitness value)

In order to evaluate the performance of the DCGA, Shimodaira (2001) suggests that multiple runs of the algorithm are required. Subsequently, a number of performance indices can then be calculated from the results. These indices are summarised in Table 1. In Table 1, the SCR and AVBF indices can be used to determine the stability in locating the optimal solution of the algorithm. In contrast, the AVFE, SDFE, AVG and SDG indices indicate the speed of the algorithm at locating the optimal solution. It is noted that the SCR, AVFE, SDFE and AVBF indices are taken directly from Shimodaira (2001) while the AVG and SDG indices are derived for use alongside the AVFE and SDFE indices.

4 Multi-objective benchmark problems and performance evaluation criteria

In order to assess the performance of the MODCGA-II, the algorithm will be benchmarked using six optimisation test cases developed by Deb et al. (2005). The problems are minimisation problems with n decision variables and m objectives. All six problems can be described in the following form:

$$\begin{array}{lrcl}
\text{Minimise} & T(\mathbf{x}) & = & (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\
\text{subject to} & f_1(\mathbf{x}) & = & h_1(x_1, \ldots, x_{m-1}, g(\mathbf{x}_m)), \\
 & f_2(\mathbf{x}) & = & h_2(x_1, \ldots, x_{m-1}, g(\mathbf{x}_m)), \\
 & f_3(\mathbf{x}) & = & h_3(x_1, \ldots, x_{m-2}, g(\mathbf{x}_m)), \\
 & & \vdots & \\
 & f_{m-1}(\mathbf{x}) & = & h_{m-1}(x_1, x_2, g(\mathbf{x}_m)) \\
\text{and} & f_m(\mathbf{x}) & = & h_m(x_1, g(\mathbf{x}_m))
\end{array} \tag{7}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{x}_m$ represents the last $k = (n - m + 1)$ decision variables of $\mathbf{x}$. Hence $|\mathbf{x}_m| = k$ and $g(\mathbf{x}_m)$ requires k decision variables. These test problems reflect various characteristics that can be found in real-world problems. In addition, the functions that are used to construct all test problems are also interchangeable. This means that other variants of the test problems can be easily constructed using the existing functions. Furthermore, these problems are also scalable in terms of the number of decision variables $n = m + k - 1$ and the number of objectives m. As a result, a change in the algorithm performance due to an increase or decrease of the number of decision variables and/or objectives can be easily interpreted. In this paper, the problems will be scaled by changing the number of objectives only where the interested numbers of objectives are two and three. By increasing the number of objectives, the difficulty level of the problem will also increase. Detailed description of each test problem follows.

4.1 Test problem DTLZ1

The test problem DTLZ1 has a linear Pareto front. The functions that form DTLZ1 are given by

$$\begin{aligned}
f_1(\mathbf{x}) &= \frac{1}{2}x_1x_2\cdots x_{m-1}(1+g(\mathbf{x}_m)),\\
f_2(\mathbf{x}) &= \frac{1}{2}x_1x_2\cdots(1-x_{m-1})(1+g(\mathbf{x}_m)),\\
&\vdots\\
f_{m-1}(\mathbf{x}) &= \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}_m)),\\
f_m(\mathbf{x}) &= \frac{1}{2}(1-x_1)(1+g(\mathbf{x}_m))\\
\text{and}\quad g(\mathbf{x}_m) &= 100\left[|\mathbf{x}_m| + \sum_{x_i\in\mathbf{x}_m}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]
\end{aligned} \tag{8}$$

where $|\mathbf{x}_m| = 5$ and $x_i \in [0, 1]$ for $i = 1, \ldots, n$. The Pareto front is formed with $g(\mathbf{x}_m) = 0$ where the Pareto optimal solutions correspond to $x_i = 0.5$ for all $x_i \in \mathbf{x}_m$ and the objective values lie on the linear hyper-plane $\sum_{i=1}^{m} f_i = 0.5$. A convergence of solutions to the true Pareto front is difficult to achieve since the search space contains $11^5 - 1$ local Pareto fronts. This means that the solutions generated by a genetic algorithm can be attracted to the local fronts prior to reaching the true front.

4.2 Test problem DTLZ2

The test problem DTLZ2 has a spherical Pareto front. The functions that form DTLZ2 are given by

$$\begin{aligned}
f_1(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\cos(x_{m-1}\pi/2),\\
f_2(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\sin(x_{m-1}\pi/2),\\
f_3(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(x_1\pi/2)\cdots\sin(x_{m-2}\pi/2),\\
&\vdots\\
f_m(\mathbf{x}) &= (1+g(\mathbf{x}_m))\sin(x_1\pi/2)\\
\text{and}\ g(\mathbf{x}_m) &= \sum_{x_i\in\mathbf{x}_m}(x_i-0.5)^2
\end{aligned} \tag{9}$$

where $|\mathbf{x}_m| = 10$ and $x_i \in [0, 1]$ for $i = 1, \ldots, n$. The Pareto front is formed with $g(\mathbf{x}_m) = 0$ where the Pareto optimal solutions correspond to $x_i = 0.5$ for all $x_i \in \mathbf{x}_m$ and the objective values lie on the spherical hyper-surface $\sum_{i=1}^{m} f_i^2 = 1$. This problem can be used to identify the ability of the algorithm to scale up its performance when the number of objectives is large.

4.3 Test problem DTLZ3

The test problem DTLZ3 also has a spherical Pareto front. However, the problem contains multiple local Pareto fronts. The functions that form DTLZ3 are given by

$$\begin{aligned}
f_1(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\cos(x_{m-1}\pi/2),\\
f_2(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\sin(x_{m-1}\pi/2),\\
f_3(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(x_1\pi/2)\cdots\sin(x_{m-2}\pi/2),\\
&\ \vdots\\
f_m(\mathbf{x}) &= (1+g(\mathbf{x}_m))\sin(x_1\pi/2)\\
\text{and }\ g(\mathbf{x}_m) &= 100\left[|\mathbf{x}_m| + \sum_{x_i\in\mathbf{x}_m}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]
\end{aligned} \tag{10}$$

where $|\mathbf{x}_m| = 10$ and $x_i \in [0, 1]$ for $i = 1, \ldots, n$. From Eq. (10), the structure of objective functions in the DTLZ3 problem is similar to that from the DTLZ2 problem while $g(\mathbf{x}_m)$ in the DTLZ3 problem and $g(\mathbf{x}_m)$ in the DTLZ1 problem are the same functions. As a result, both the global Pareto front of the DTLZ3 problem and the Pareto front of the DTLZ2 problem has the same shape. However, the use of $g(\mathbf{x}_m)$ as described in Eq. (10) makes the DTLZ3 problem more difficult than the DTLZ2 problem since $3^{10} - 1$ local Pareto fronts are now introduced to the problem. It is noted that the global Pareto front of the DTLZ3 problem is formed with $g(\mathbf{x}_m) = 0$ where the global Pareto optimal solutions correspond to $x_i = 0.5$ for all $x_i \in \mathbf{x}_m$ and the objective values lie on the spherical hyper-surface $\sum_{i=1}^{m} f_i^2 = 1$.

4.4 Test problem DTLZ4

The test problem DTLZ4 also has a spherical Pareto front with the shape similar to that from the DTLZ2 problem. The functions that form DTLZ4 are given by

$$\begin{aligned}
f_1(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos\left(x_1^{\beta}\pi/2\right)\cdots\cos\left(x_{m-2}^{\beta}\pi/2\right)\cos\left(x_{m-1}^{\beta}\pi/2\right),\\
f_2(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos\left(x_1^{\beta}\pi/2\right)\cdots\cos\left(x_{m-2}^{\beta}\pi/2\right)\sin\left(x_{m-1}^{\beta}\pi/2\right),\\
f_3(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos\left(x_1^{\beta}\pi/2\right)\cdots\sin\left(x_{m-2}^{\beta}\pi/2\right),\\
&\ \vdots\\
f_m(\mathbf{x}) &= (1+g(\mathbf{x}_m))\sin\left(x_1^{\beta}\pi/2\right)\\
\text{and }\ g(\mathbf{x}_m) &= \sum_{x_i\in\mathbf{x}_m}(x_i-0.5)^2
\end{aligned} \tag{11}$$

where $|\mathbf{x}_m| = 10$ and $x_i \in [0, 1]$ for $i = 1, \ldots, n$. It can be seen that the objective function in Eq. (11) are created by replacing x_i for $i = 1, \ldots, m-1$ in Eq. (9) with x_i^{β}. In this investigation, β is set to 100; this modification allows a dense set of solutions to exist near the f_m-f_1 plane. In other words, the Pareto optimal solutions are non-uniformly distributed along the Pareto front. The conditions for a solution to be Pareto optimal for this problem is the same as that given for the DTLZ2 problem.

4.5 Test problem DTLZ5

The Pareto front of the test problem DTLZ5 can be visually displayed as a curve. The functions that form DTLZ5 are given by

$$
\begin{aligned}
f_1(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(\theta_1)\cdots\cos(\theta_{m-2})\cos(\theta_{m-1}),\\
f_2(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(\theta_1)\cdots\cos(\theta_{m-2})\sin(\theta_{m-1}),\\
f_3(\mathbf{x}) &= (1+g(\mathbf{x}_m))\cos(\theta_1)\cdots\sin(\theta_{m-2}),\\
&\vdots\\
f_m(\mathbf{x}) &= (1+g(\mathbf{x}_m))\sin(\theta_1)\\
\text{and } g(\mathbf{x}_m) &= \sum_{x_i\in\mathbf{x}_m}(x_i-0.5)^2
\end{aligned}
\tag{12}
$$

where $|\mathbf{x}_m| = 10$, $x_i \in [0, 1]$ for $i = 1, \ldots, n$, $\theta_1 = x_1\pi/2$ and $\theta_i = \frac{\pi}{4(1+g(\mathbf{x}_m))}(1 + 2g(\mathbf{x}_m)x_i)$ for $i = 2, \ldots, m-1$. This test problem is created by changing the argument of the sinusoidal functions within the DTLZ2 objective functions from $x_i\pi/2$ to θ_i. As a result, this problem will test the ability of the search algorithm to produce converged solutions that form a curve. The performance can be visually observed by plotting the f_m objective with any other objectives. The conditions for a solution to be Pareto optimal for this problem are also the same as that given for the DTLZ2 problem.

4.6 Test problem DTLZ6

The test problem DTLZ6 is a harder version of the test problem DTLZ5. Similar to the modification done on the DTLZ2 problem in order to create the DTLZ3 problem, the modification made to the DTLZ5 also involves the use of a different $g(\mathbf{x}_m)$ function, which leads to the introduction of local Pareto fronts. However, in the test problem DTLZ6 the $g(\mathbf{x}_m)$ function is given by

$$
g(\mathbf{x}_m) = \sum_{x_i\in\mathbf{x}_m} x_i^{0.1} \tag{13}
$$

where $|\mathbf{x}_m|$ is also equal to 10. The true Pareto front is formed with $g(\mathbf{x}_m) = 0$ where the Pareto optimal solutions correspond to $x_i = 0$ for all $x_i \in \mathbf{x}_m$. Multiple local Pareto fronts make this problem a hard problem.

It can be seen that each test problem represents a different aspect of multi-objective optimisation problem. The characteristic of each problem is summarised in Table 2. The performance criteria used for benchmarking the MODCGA-II will be explained next.

Table 2 Summary of the characteristic of each multi-objective problem

Problem	Shape of pareto front	Difficulty
DTLZ1	Linear front	Multiple local Pareto fronts
DTLZ2	Spherical front	None
DTLZ3	Spherical front	Multiple local Pareto fronts
DTLZ4	Spherical front	Non-uniform distribution of solutions in the search space
DTLZ5	Curve	None
DTLZ6	Curve	Multiple local Pareto fronts

4.7 Performance evaluation criteria

In Section 3, the performance of the DCGA can be evaluated from the stability and efficiency of the algorithm at finding a single optimal solution. However, the performance of the MODCGA-II cannot be measured using the same criteria. This is because there is more than one optimal solution in each multi-objective benchmark problem where the solutions are optimal in the sense that they are non-dominated solutions. Zitzler et al. (2000) suggest that in order to assess the optimality of the non-dominated solutions identified by a multi-objective optimisation algorithm, these solutions should be compared either among themselves or with the true Pareto optimal solutions of the benchmark problem. Two measurement criteria described in Zitzler et al. (2000) will be used here: the average distance between the non-dominated solutions to the true Pareto optimal solutions and the distribution of the non-dominated solutions. The mathematical description of these two measurement criteria is given as follows.

Let Y' be the set of non-dominated solutions identified by the search algorithm and $\overline{Y}$ be the set of true Pareto optimal solutions where Y' and $\overline{Y}$ are subsets of the objective space Y. The average distance between the non-dominated solutions to the true Pareto optimal solutions (M_1) is given by

$$M_1(Y') = \frac{1}{|Y'|} \sum_{\mathbf{p}' \in Y'} \min\{\|\mathbf{p}' - \overline{\mathbf{p}}\|; \overline{\mathbf{p}} \in \overline{Y}\} \tag{14}$$

where $\mathbf{p}'$ is a non-dominated solution, $\overline{\mathbf{p}}$ is a true Pareto optimal solution, $|\cdot|$ denotes the size of a set and $\|\cdot\|$ represents a Euclidean norm. Similarly, the distribution of the non-dominated solutions in combination with the number of solutions found (M_2) is given by

$$M_2(Y') = \frac{1}{|Y' - 1|} \sum_{\mathbf{p}' \in Y'} |\{\mathbf{q}' \in Y'; \|\mathbf{p}' - \mathbf{q}'\| > \sigma\}| \tag{15}$$

where σ is a neighbourhood parameter and $\mathbf{q}'$ is a solution which is not in the neighbourhood of $\mathbf{p}'$. The range of M_2 is between zero and the number of solutions in the non-dominated solution set Y'; this also reflects the number of σ-niches in Y'. The higher the value of M_2, the better the distribution for a selected neighbourhood parameter. Notice from Eq. (15) that the maximum value of M_2 is limited by the maximum possible number of non-dominated solutions that can be produced by the algorithm; this can be in the form of the population size or the size of individual archive.

In addition to the average distance between the non-dominated solutions to the true Pareto optimal solutions and the distribution of the non-dominated solutions, Zitzler et al. (2000) also suggest the third measurement index for the extent of the front described by the non-dominated solutions. This index (M_3) is defined by

$$M_3(Y') = \sqrt{\sum_{i=1}^{m} \max\{\| p_i' - q_i' \|; \mathbf{p}', \mathbf{q}' \in Y'\}} \tag{16}$$

where m is the number of objectives. The function M_3 uses the maximum extent in each dimension to estimate the range of the front. In contrast to M_1 and M_2 indices, the trend of the value of M_3 index that can be used to indicate the algorithm performance is not obvious. This is because the extent of true Pareto front in each test problem has a specific value. A large M_3 index may initially appear to indicate that the algorithm is capable of spreading the solutions across the whole range of Pareto front. However, if the non-dominated solutions identified and the true Pareto optimal solutions are very far apart, the M_3 index calculated from the non-dominated solutions may be even larger than that calculated from the true Pareto optimal solutions. This phenomenon has been observed in the early work by Sangkawelert and Chaiyaratana (2003) where the obtained values of M_3 from various test problems cannot be properly used to evaluate the performance of the MODCGA. Based on the previous work, the use of M_3 index as a performance indicator is not recommended and further modification of the M_3 index is required (Maneeratana et al., 2004). Although the measurement criteria discussed here are defined for calculations using objective vectors, the alternative criteria, which are defined for decision vectors, have also been discussed in Zitzler et al. (2000). The counterpart measurement criteria for decision vectors are not used in this paper.

5 Single-objective optimisation results and discussions

In this section, the results from using a DCGA and a multi-objective genetic algorithm, as recommended by Abbass and Deb (2003) and Jensen (2003), to solve onemax and royal road R_1 problems will be presented. In the case of DCGA, both CDSS and CPSS techniques are considered. With the use of the CPSS technique, two parameters are needed in order to calculate the survival probability of an individual: the shape coefficient (c) and the exponent coefficient (α). The values of c in the investigation are 0.00, 0.25, 0.50, 0.75 and 1.00 while the values of α are set to 0.00, 0.20, 0.40, 0.60, 0.80 and 1.00. This leads to the total of 30 parameter settings. From Eq. (1), by setting either $c = 1.00$ or $\alpha = 0.00$, the CPSS technique will change to the CDSS technique since the survival probability of each non-elite individual is equal to one. The diversity control study will be conducted with other genetic parameters remain fixed throughout the trial. The common parameter setting for the DCGA that is used in both onemax and royal road R_1 problems is displayed in Table 3 while the specific parameter setting for each benchmark problem is shown in Table 4.

Table 3 Common parameter setting for the DCGA that is used in all single-objective problems

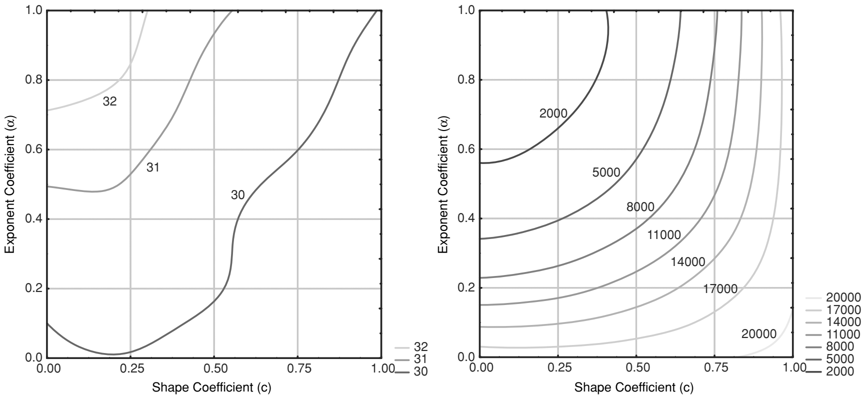
Parameter	Value
Fitness scaling method	Linear fitness scaling (Goldberg, 1989)
Selection method	Stochastic universal sampling (Baker, 1989)
Crossover method	Uniform crossover with the probability = 0.8
Mutation method	Bit-flip mutation with the probability = 0.025
Population size	100
Number of repeated runs	30

Table 4 Specific parameter setting for each single-objective problem

Problem	Chromosome length	Number of generations
48-bit onemax	48	3,600
64-bit onemax	64	6,400
80-bit onemax	80	10,000
96-bit onemax	96	14,400
112-bit onemax	112	19,600
$R_1(48, 6)$ royal road	48	3,600
$R_1(64, 8)$ royal road	64	6,400
$R_1(80, 10)$ royal road	80	10,000
$R_1(96, 12)$ royal road	96	14,400
$R_1(112, 14)$ royal road	112	19,600

For each diversity control setting, the DCGA runs for the onemax and royal road R_1 problems are repeated 30 times. The SCR, AVBF, AVFE, SDFE, AVG and SVG indices as described in Section 3 are subsequently obtained and the average values of the six indices calculated from all onemax problems and that from all royal road R_1 problems are considered. For the onemax problems, all six indices indicate that the variation in the shape coefficient (c) and the exponent coefficient (α) has a very little effect on the DCGA performance. In addition, the SCR index for the majority of the diversity control settings is also equal to one, which means that the DCGA can always solve the onemax problems regardless of the parameter setting. Similarly, the SCR index for the majority of the diversity control settings in the case of the royal road R_1 problems is also equal to one. However, the AVFE and AVG indices in the royal road R_1 problems indicate that different parameter setting leads to different speed at locating the optimal solution. For the purpose of comparison, the AVG indices from the onemax and royal road R_1 problems are displayed as contour plots in Fig. 6. From Fig. 6, it can be clearly seen that the contour is quite flat for the onemax problems while the area that have a low AVG index for the royal road R_1 problems is in the vicinity of the setting $c = 0.25$ and $\alpha = 0.80$. In this investigation, the setting of $c = 0.25$ and $\alpha = 0.80$ is hence chosen as the candidate setting that leads to high algorithm speed at locating the solution for both onemax and royal road R_1 problems. The shape of contour for the AVFE index is similar to that displayed in Fig. 6 and its illustration is hence omitted.

The search performance of the DCGA with $c = 0.25$ and $\alpha = 0.80$ will be compared with that from the approach suggested by Abbass and Deb (2003) and Jensen



(a) (b)

Fig. 6 Average value of the AVG index for each diversity control setting (a) onemax problems (b) royal road R_1 problems

(2003). Basically, the main idea behind both previous works is to transform the single-objective problem of interest into a multi-objective problem. This can be done using the original objective of the problem as the first optimisation objective and creating additional objectives that are conflicting with the first objective. After the problem transformation, the resulting multi-objective problem is then solved using a multi-objective genetic algorithm with elitism. Since multiple non-dominated solutions can generally be obtained from the multi-objective algorithm, the spread of non-dominated solutions in multi-dimensional objective space would also lead to the diversification of single-objective solutions in the objective space of the original problem. In this investigation, the approach suggested by Abbass and Deb (2003) and Jensen (2003) will be implemented by transforming the onemax and royal road R_1 functions into two-objective functions; the additional objectives for both functions can be described as follows.

For an L-bit onemax function, its conflicting function can be described as an L-bit zeromax function, which is defined by

$$\text{objective} = \sum_{i=1}^{L} (1 - b_i) \tag{17}$$

where $b_i \in [0, 1]$ is the ith bit on the chromosome. The maximum value of the zeromax function is equal to L and is obtained when every bit in the chromosome is equal to zero. However, with the use of a zeromax function to provide the conflicting objective, the resulting two-objective function will have a linear Pareto front where every possible solution in the search space also lies on the front. Figure 7 illustrates this problem for a 4-bit onemax+zeromax function. From Fig. 7, an attempt to use a multi-objective algorithm to solve the onemax+zeromax problem straightaway would probably fail since every possible solution in the search space is a non-dominated solution. Nonetheless, with the use of a goal attainment technique (Fonseca and Fleming, 1998) the Pareto front can be split into two parts: the part inside the goal and

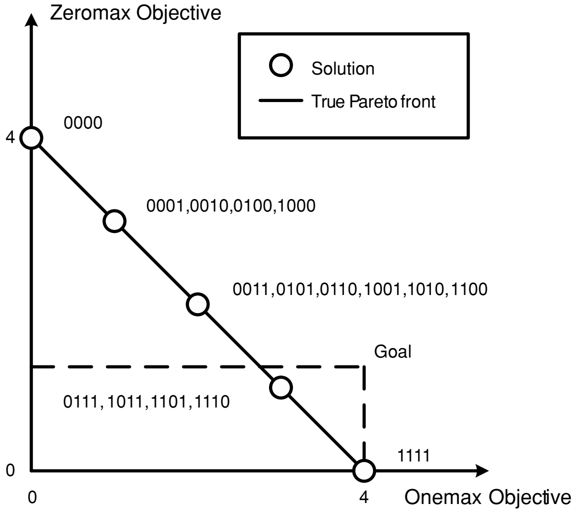


Fig. 7 The Pareto front and search space for the 4-bit onemax+zeromax function

the part outside the goal. As a result, the solutions inside the goal would dominate the solutions outside the goal and hence have more survival chances. Obviously, the optimal solutions of the onemax function must also be inside the goal. In this paper, 30% of the straight-line that describes the linear Pareto front will be inside the goal.

Similarly, the conflicting objective for an $8t$-bit royal road R_1 function can be defined by Eqs. (5) and (6). However, the eight non-overlapping order t, defining length t, schemata would make up from zero instead of one. Figure 8 illustrates eight schemata that are required to construct the optimal solution that satisfies the conflicting objective of the $R_1(64, 8)$ function. From Fig. 8, the optimal solution is the solution that every bit is zero and the corresponding objective value is 64. With the use of the conflicting objectives described above, the resulting two-objective royal road R_1 function also has a linear Pareto front. However, in contrast to the onemax+zeromax function not all solutions in the search space reside on the Pareto front in the case of the two-objective royal road R_1 function. This means that the use of a goal attainment technique is not required and the problem can be solved directly using a multi-objective algorithm.

Since the two-objective versions of both onemax and royal road R_1 functions have linear Pareto fronts, the multi-objective algorithm that are chosen to identify

S_1 = 00000000**

S_2 = ********00000000**

S_3 = ****************00000000**

S_4 = ************************00000000********************************

S_5 = ********************************00000000************************

S_6 = **00000000****************

S_7 = **00000000********

S_8 = **00000000

Fig. 8 Eight schemata for the conflicting function of the $R_1(64, 8)$ function

Table 5 Common parameter setting for the MODCGA-II that is used in all two-objective onemax and royal road R_1 problems

Parameter	Value
Chromosome coding	Binary representation
Fitness sharing	Triangular sharing function (Goldberg, 1989) with the sharing radius $= 0.25\times$ sharing radius used in Fonseca and Fleming (1993)
Fitness assignment	Linear fitness interpolation
Selection method	Stochastic universal sampling (Baker, 1989)
Crossover method	Uniform crossover with the probability = 0.8
Mutation method	Bit-flip mutation with the probability = 0.025
Population size	100
Archive size	100
Number of repeated runs	30

the non-dominated solutions in this investigation is the MODCGA-II. Specifically, the CPSS technique will be used where the shape coefficient (c) and the exponent coefficient (α) are set to 0.75 and 0.20, respectively. The reason for this decision will be apparent in Section 6 where the MODCGA-II is proven to be an efficient algorithm for a two-objective problem with a linear Pareto front. The common parameter setting for the MODCGA-II that is used in both two-objective onemax and royal road R_1 problems is displayed in Table 5 while the specific parameter setting for each benchmark problem for the MODCGA-II including the chromosome length and the number of generations is the same as that displayed in Table 4. It can be seen that the majority of the setting in Table 5 is similar to that in Table 3 except for the fitness sharing, fitness assignment and solution archive that is used in the MODCGA-II only.

The search performance of the DCGA with $c = 0.25$ and $\alpha = 0.80$ is compared with that from the MODCGA-II where the SCR, AVBF, AVFE, SDFE, AVG and SDG indices are calculated from 30 repeated runs. The search performance for the onemax function is summarised in Table 6 while that for the case of royal road R_1 function is displayed in Table 7. In Table 6, the SCR and AVBF indices indicate that the DCGA has successfully located the solutions of the onemax problems in all repeated runs while only the majority of the MODCGA-II runs that can be identified as the successful runs. Nonetheless, the major difference in the search performance of the two algorithms can be observed from the AVFE, SDFE, AVG and SDG indices. The four indices indicate that as the problem size increases, the number of objective evaluations required by both DCGA and MODCGA-II also increases. However, it is noticeable that regardless of the change in the problem size the DCGA remains much faster than the MODCGA-II at locating the optimal solution in all onemax problems. This leads to the conclusion that the most suitable algorithm for solving the onemax problems is the DCGA.

Moving onto the benchmarking results where the royal road R_1 function is used from Table 7. According to the SCR and AVBF indices, the DCGA is proven to be able to solve the royal road R_1 problems in all repeated runs while the MODCGA-II completely fails to solve the problems. This confirms the suitability of the DCGA for this class of benchmark problems. In terms of the algorithm's speed for locating the optimal solution, the AVFE, SDFE, AVG and SDG indices indicate that in a similar

Table 6 Summary of the DCGA and MODCGA-II performances on the onemax problems

Problem	Algorithm	SCR	AVFE	SDFE	AVG	SDG	AVBF
48-bit onemax	DCGA	1.00	793.33	159.60	7.93	1.60	100.00
	MODCGA-II	1.00	11,666.67	10,430.11	58.33	52.15	100.00
64-bit onemax	DCGA	1.00	1,200.00	136.46	12.00	1.36	100.00
	MODCGA-II	0.97	46,346.67	44,737.21	231.73	223.69	99.73
80-bit onemax	DCGA	1.00	1,660.00	239.76	16.60	2.40	100.00
	MODCGA-II	0.97	68,533.33	50,192.11	342.67	250.96	99.91
96-bit onemax	DCGA	1.00	2,393.33	255.87	23.93	2.56	100.00
	MODCGA-II	0.93	84,946.67	52,938.40	424.73	264.69	99.79
112-bit onemax	DCGA	1.00	3,343.33	335.98	33.43	3.36	100.00
	MODCGA-II	0.93	80,906.67	50,016.18	404.53	250.08	99.02

Table 7 Summary of the DCGA and MODCGA-II performances on the royal road R_1 problems

Problem	Algorithm	SCR	AVFE	SDFE	AVG	SDG	AVBF
$R_1(48, 6)$	DCGA	1.00	1,036.67	278.52	10.37	2.97	100.00
	MODCGA-II	0.00	–	–	–	–	76.67
$R_1(64, 8)$	DCGA	1.00	7,040.00	9,863.75	70.40	98.64	100.00
	MODCGA-II	0.00	–	–	–	–	70.00
$R_1(80, 10)$	DCGA	1.00	40,526.67	36,442.82	405.27	364.43	100.00
	MODCGA-II	0.00	–	–	–	–	65.00
$R_1(96, 12)$	DCGA	1.00	125,720.00	50,428.98	1,257.20	504.29	100.00
	MODCGA-II	0.00	–	–	–	–	44.38
$R_1(112, 14)$	DCGA	1.00	222,656.67	73,677.59	2,226.57	736.78	100.00
	MODCGA-II	0.00	–	–	–	–	36.16

way to the early results from the onemax problems as the royal road R_1 problem size increases, the number of objective evaluations required by the DCGA also increases. In addition, the speed of the DCGA at solving the royal road $R_1(48, 6)$ and $R_1(64, 8)$ problems is comparable to that of the DCGA for the cases of onemax problems while the speed of the DCGA is significantly lower in the royal road $R_1(80, 10)$, $R_1(96, 12)$ and $R_1(112, 14)$ problems. Recall from Section 3 that the royal road $R_1(48, 6)$ and $R_1(64, 8)$ problems and all onemax problems are classified as straightforward problems while the royal road $R_1(80, 10)$, $R_1(96, 12)$ and $R_1(112, 14)$ problems are identified as difficult problems according to the fitness distance correlation measurement. This means that the speed of the DCGA at solving the problems investigated depends more on the difficulty level of the problem than on the problem/search space size.

6 Multi-objective optimisation results and discussions

In this section, the results from using the MODCGA-II to solve test problems DTLZ1–DTLZ6 will be presented. The results will be benchmarked against that obtained from the non-dominated sorting genetic algorithm II or NSGA-II (Deb et al., 2002a) and the improved strength Pareto evolutionary algorithm or SPEA-II (Zitzler et al., 2002) where the executable codes for the implementation of both algorithms are obtained directly from *A Platform and Programming Language Independent Interface* *for Search Algorithms (PISA)* web site (http://www.tik.ee.ethz.ch/pisa). Similar to the investigation on single-objective optimisation, both CDSS and CPSS techniques are utilised in the implementation of the MODCGA-II. In addition, the same parameter settings for the CPSS technique used in the previous section will also be used here. The common parameter setting for the MODCGA-II, NSGA-II and SPEA-II that is used in all problems is displayed in Table 8 while the specific parameter setting for each problem is displayed in Table 9. In Table 8, the fitness sharing and mating restriction is carried out by considering the distance between the two interested individuals in the objective space. In addition, the estimation of the sharing radius using the hyper-surface approximation of the Pareto front in the objective space as suggested in Fonseca and Fleming (1993) is also carried out. Since this approximation gives an overestimated result, the sharing radius is thus scaled down. In Table 9, the number of generations for the MODCGA-II is smaller than that for the NSGA-II and SPEA-II. Again, this is because the number of fitness evaluations within one generation of the MODCGA-II would generally be larger than that in the NSGA-II and SPEA-II since the MODCGA-II allows an introduction of newly generated individuals to the population.

Table 8 Common parameter setting for the MODCGA-II, NSGA-II and SPEA-II that is used in all multi-objective problems

Parameter	Value
Chromosome coding	Real-value representation
Fitness sharing (MODCGA-II only)	Triangular sharing function (Goldberg, 1989) with the sharing radius $= 0.25\times$ sharing radius used in Fonseca and Fleming (1993)
Fitness assignment (MODCGA-II only)	Linear fitness interpolation
Selection method	Stochastic universal sampling (MODCGA-II) or tournament selection (NSGA-II and SPEA-II)
Crossover method	SBX recombination (Deb, 2001) with the probability $= 1.0$
Mutation method	Variable-wise polynomial mutation (Deb, 2001) with the probability = 1/number of decision variables
Population size	100
Archive size (MODCGA-II, SPEA-II)	100
Number of repeated runs	30

Table 9 Specific parameter setting for each multi-objective problem

	Number of variables		Number of generations		
Problem	2 Objectives	3 Objectives	MODCGA-II	NSGA-II	SPEA-II
DTLZ1	6	7	300	600	600
DTLZ2	11	12	300	600	600
DTLZ3	11	12	300	600	600
DTLZ4	11	12	300	600	600
DTLZ5	11	12	300	600	600
DTLZ6	11	12	300	600	600

Similar to the approach in the investigation on single-objective optimisation, five values of the shape coefficient (c)—0.00, 0.25, 0.50, 0.75 and 1.00—and six values of the exponent coefficient (α)—0.00, 0.20, 0.40, 0.60, 0.80 and 1.00—are used to create 30 different diversity control settings for the MODCGA-II. From Eq. (3), the settings of $c = 1.00$ and $\alpha = 0.00$ are for the implementation of the CDSS technique since the survival probability of each dominated individual is equal to one. For each setting, the MODCGA-II runs for the DTLZ1–DTLZ6 problems with two and three objectives are repeated 30 times. The M_1 and M_2 performance indices from each run are subsequently obtained and the average values of the two indices calculated from all two-objective problems and that from all three-objective problems are displayed in the form of contour plots in Fig. 9. The M_2 index is calculated using the neighbourhood parameter $\sigma = 0.488$ and the index has been normalised by the maximum attainable number of non-dominated individuals from a single run. From Fig. 9, it is noticeable that diversity control settings considered have a small effect in three-objective problems while a significant performance variation can be detected in two-objective problems.

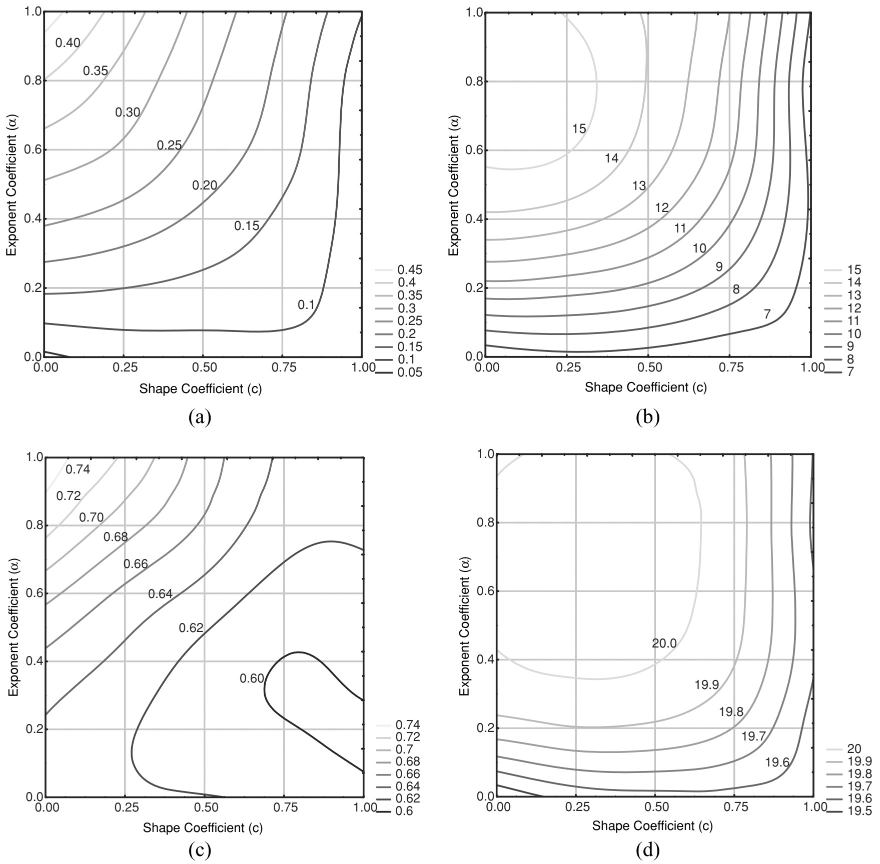


Fig. 9 Average values of M_1 and M_2 indices from all multi-objective problems for each diversity control setting (a) M_1 index from two-objective problems (b) M_2 index from two-objective problems (c) M_1 index from three-objective problems (d) M_2 index from three-objective problems

For the case of two-objective problems, the region where the M_1 index has a small value coincides with the area where the M_2 index is small. At the same time the region where the M_1 index is high is also in the vicinity of the area where the M_2 index has a large value. In a successful multi-objective search, the M_1 index should be as small as possible. Although a large M_2 index usually signifies a good solution distribution, the interpretation of the M_2 result must always be done while taken the M_1 index into consideration. This is because in the case where the solutions are further away from the true Pareto optimal solutions, the obtained value of the M_2 index is generally high since each solution would also be far apart from one another. In other words, the M_2 index has a lesser priority than the M_1 index and should be considered only when the obtained values of the M_1 index from two different algorithms or algorithm settings are close to one another. Using the above argument, multiple settings of the c and α values in Fig. 9 can be used to achieve low M_1 indices. In the current investigation, the setting where $c = 0.75$ and $\alpha = 0.20$ is chosen as the candidate setting that represents the diversity control that leads to a low M_1 value.

The search performance of the MODCGA-II with $c = 0.75$ and $\alpha = 0.20$ will be compared with that from the NSGA-II and SPEA-II. As stated in Table 8, each algorithm run will be repeated 30 times where the M_1 and normalised M_2 indices are subsequently calculated for each repeated run. After all repeated runs are completed, the individuals from all runs are merged together where the final non-dominated individuals are then extracted. The performance of the MODCGA-II, NSGA-II and SPEA-II in terms of the average and standard deviation of the M_1 and normalised M_2 indices on the two-objective and three-objective DTLZ1–DTLZ6 problems is summarised in Tables 10 and 11, respectively. In Tables 10 and 11, the neighbourhood parameter (σ) for the calculation of M_2 indices for all test problems is also set to 0.488; the parameter is set using the extent of the true Pareto front in the objective space as the guideline. In addition, the final non-dominated solutions extracted from merged individuals from all runs and the true Pareto optimal solutions in the objective

Table 10 Summary of the MODCGA-II, NSGA-II and SPEA-II performances on the two-objective DTLZ1–DTLZ6 problems

		Algorithm					
		MODCGA-II		NSGA-II		SPEA-II	
Problem	Index	Average	S.D.	Average	S.D.	Average	S.D.
DTLZ1	M_1	3.1157	4.7837	11.9186	5.1490	12.9616	5.2649
	M_2	0.4326	0.2942	0.6391	0.0474	0.7810	0.0547
DTLZ2	M_1	0.0030	0.0008	0.0148	0.0088	0.0190	0.0096
	M_2	0.5039	0.0439	0.5672	0.0310	0.5053	0.0494
DTLZ3	M_1	22.2335	18.1880	78.6069	24.8055	88.4823	22.4487
	M_2	0.5642	0.3941	0.6119	0.0814	0.7463	0.0745
DTLZ4	M_1	0.0023	0.0018	0.0238	0.0138	0.0252	0.0104
	M_2	0.3353	0.2111	0.2871	0.2353	0.3457	0.2417
DTLZ5	M_1	0.0030	0.0006	0.0148	0.0088	0.0175	0.0079
	M_2	0.4972	0.0495	0.5672	0.0310	0.5026	0.0501
DTLZ6	M_1	1.0199	0.3685	6.4295	0.3509	6.4986	0.3355
	M_2	0.8044	0.0603	0.7157	0.0493	0.8946	0.0166

Table 11 Summary of the MODCGA-II, NSGA-II and SPEA-II performances on the three-objective DTLZ1–DTLZ6 problems

		Algorithm					
		MODCGA-II		NSGA-II		SPEA-II	
Problem	Index	Average	S.D.	Average	S.D.	Average	S.D.
DTLZ1	M_1	24.8583	8.3026	7.3259	3.1478	12.2895	3.9570
	M_2	0.9923	0.0130	0.8762	0.0900	0.4542	0.0283
DTLZ2	M_1	0.0473	0.0069	0.0119	0.0095	0.0120	0.0069
	M_2	0.7281	0.0395	0.6776	0.0262	0.3502	0.0227
DTLZ3	M_1	282.8405	73.2361	67.5248	18.6048	89.5977	31.0686
	M_2	1.0000	0.0000	0.8333	0.0880	0.4575	0.0229
DTLZ4	M_1	0.0519	0.0240	0.0192	0.0097	0.0257	0.0153
	M_2	0.7229	0.0724	0.4371	0.1772	0.2714	0.0941
DTLZ5	M_1	0.0088	0.0024	0.0138	0.0119	0.0197	0.0088
	M_2	0.4916	0.0527	0.4061	0.0733	0.2080	0.0346
DTLZ6	M_1	3.9762	0.7985	5.8548	0.3046	6.3882	0.3039
	M_2	0.9847	0.0119	0.9842	0.0014	0.4999	0.0005

space for the two-objective DTLZ1–DTLZ6 problems are displayed in Figs. 10–12 while that for the three-objective counterpart are illustrated in Figs. 13–15.

Firstly, consider the results from the two-objective test problems, which are displayed in Table 10 and Figs. 10–12. In terms of the average distance from the non-dominated solutions identified to the true Pareto front or the M_1 criterion, the MODCGA-II posses the highest performance in all six test problems. Nonetheless, the MODCGA-II is unable to identify the true Pareto optimal solutions in the DTLZ3 and DTLZ6 problems. These two problems are difficult to solve in the sense that they contains multiple local Pareto fronts. Although the DTLZ1 problem also contains numerous local Pareto fronts, the majority of results from all 30 MODCGA-II runs indicate that the MODCGA-II is capable of solving this problem. This means that the shape of Pareto front in two-objective problems can also affect the algorithm performance since the DTLZ1 problem has a straight-line Pareto front while the DTLZ3 and DTLZ6 problems have curve Pareto fronts. The M_1 index also reveals that the performance of NSGA-II and SPEA-II are very similar in all six problems. Since the M_1 indices from both algorithms are quite close, a further inspection on the M_2 indices can be easily made. Again, the M_2 indices from the NSGA-II and SPEA-II are also very close to one another. This leads to the conclusion that for the two-objective problems, the capability of both NSGA-II and SPEA-II is pretty much the same.

Moving onto the results from the three-objective test problems, which are displayed in Table 11 and Figs. 13–15. In terms of the M_1 criterion, the NSGA-II posses the highest performance in the DTLZ1–DTLZ4 problems while the MODCGA-II is the best algorithm for the DTLZ5 and DTLZ6 problems. However, the DTLZ1, DTLZ3 and DTLZ6 problems cannot be solved using either the MODCGA-II or the NSGA-II. These three problems contain multiple local Pareto front and hence can be classified as difficult problems. It is also noticeable that in contrast to the two-objective case, the shape of the Pareto front has no effect on the algorithm's ability to identify the correct solutions. By comparing the two-objective results with the three-objective results, it

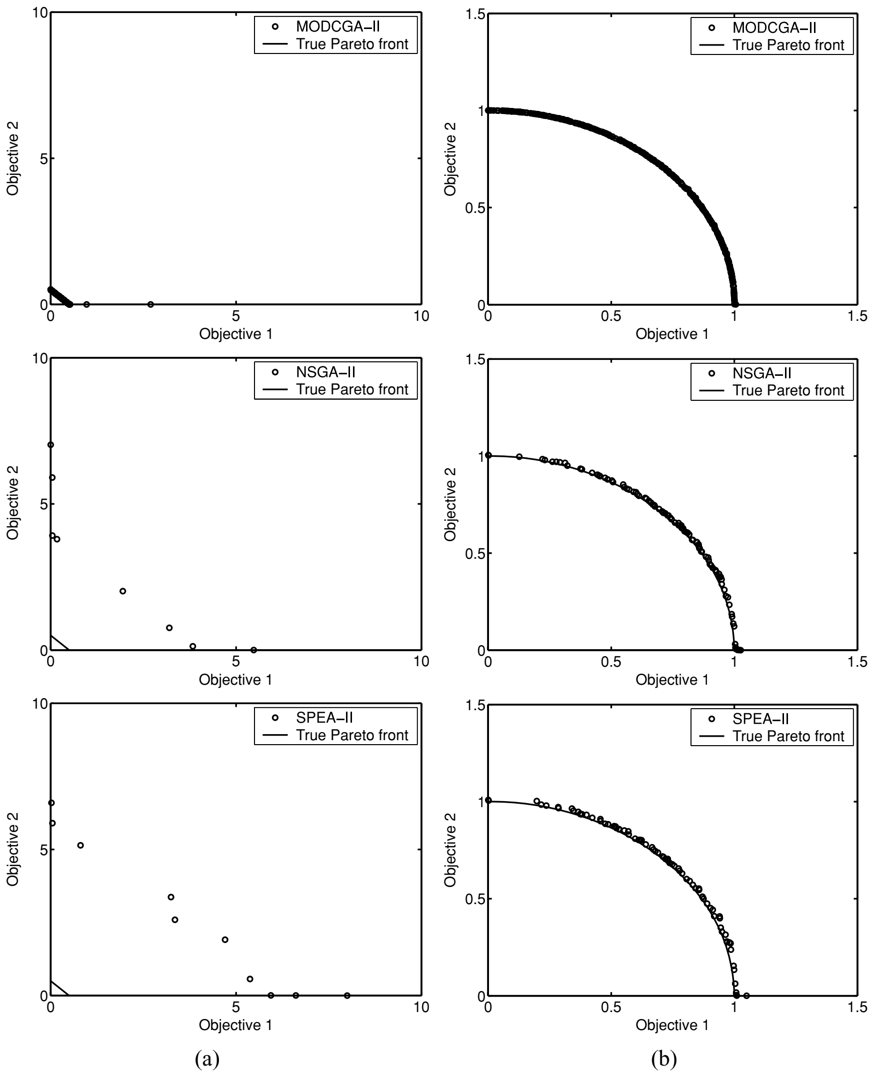


Fig. 10 Optimisation results on the two-objective problems (a) DTLZ1 (b) DTLZ2

can be seen that there is deterioration in performances in the MODCGA-II and SPEA-II while the performance of the NSGA-II remains unchanged. The deterioration of the SPEA-II performance is detectable only in terms of the solution distribution (M_2 index) and not in terms of the closeness of solutions to the true Pareto front (M_1 index). As a result, the SPEA-II solutions now have a worse distribution than that from the NSGA-II while the solutions from both algorithms are at a similar distance from the true Pareto front. On the other hand, the performance degradation in the MODCGA-II is highest when the problem involves multiple local Pareto fronts. Nonetheless, even with the performance reduction, the MODCGA-II is still better than the NSGA-II at solving the DTLZ5 and DTLZ6 problems where the Pareto fronts can be visually

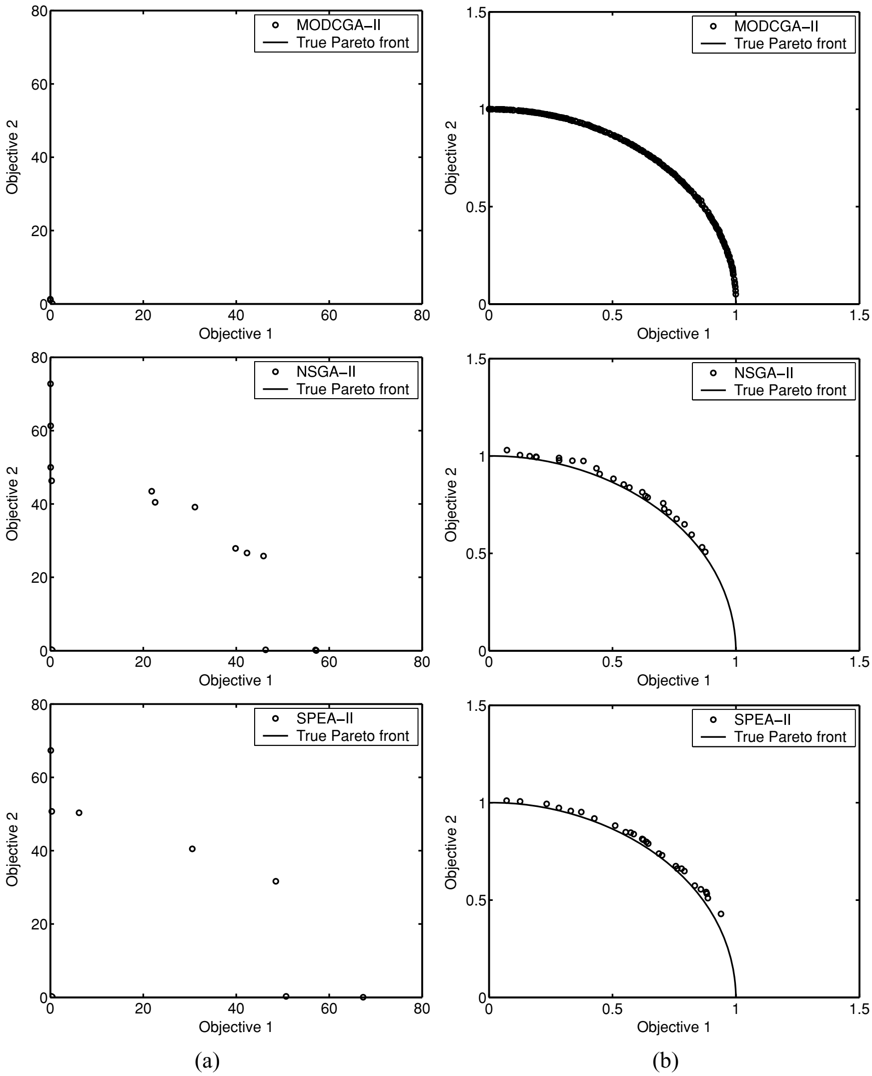
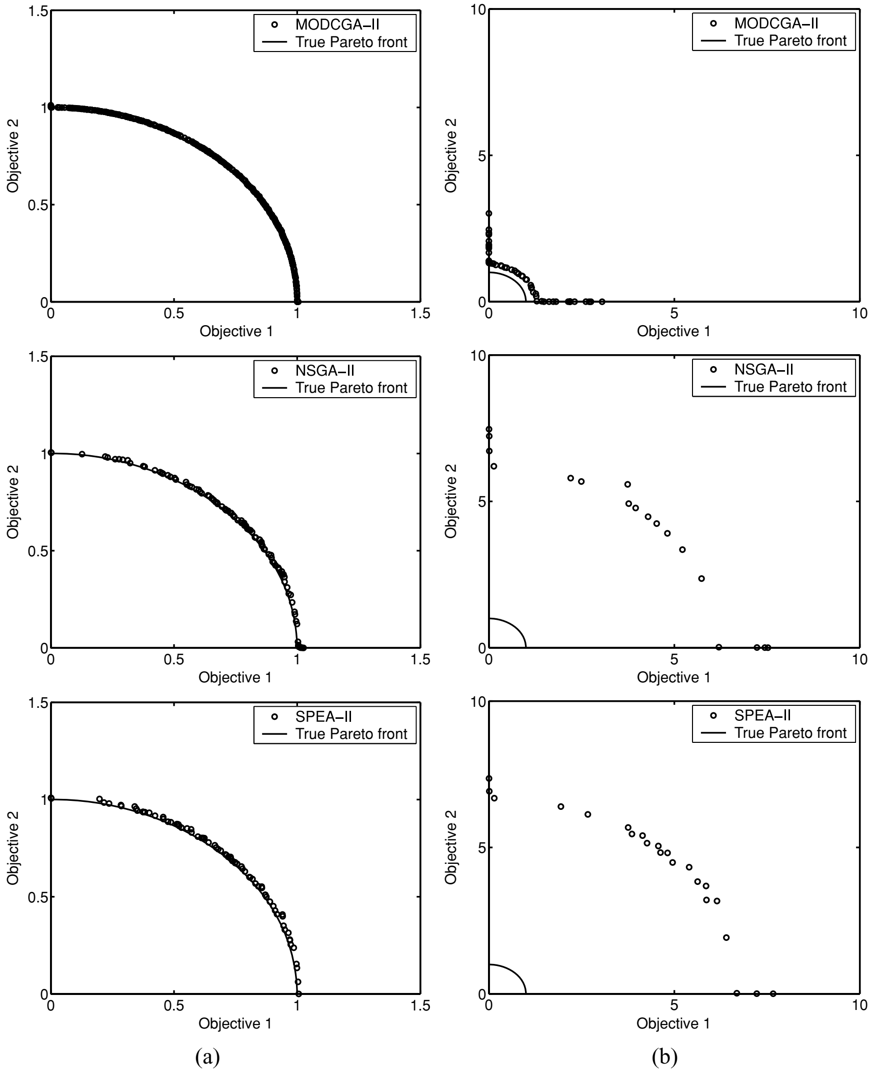


Fig. 11 Optimisation results on the two-objective problems (a) DTLZ3 (b) DTLZ4

displayed as two-objective curves. However, the difference in the search performances of the two algorithms in the DTLZ5 and DTLZ6 problems is not as evident as that in the DTLZ1–DTLZ4 problems. This can be confirmed by the M_2 indices, which clearly indicate that in the last two problems the distributions of solutions from the MODCGA-II and NSGA-II are very similar. Using both the performance indices and the consideration on the deterioration in algorithm performance after increasing the number of optimisation objectives, it can be concluded that in overall the best algorithm for the three-objective test problems is the NSGA-II. This also indicates the limitation of the MODCGA-II in multi-objective optimisation.



(a) (b)

Fig. 12 Optimisation results on the two-objective problems (a) DTLZ5 (b) DTLZ6

7 Conclusions

In this paper, the effects of diversity control on single-objective and multi-objective genetic algorithms have been investigated. The diversity control operator interested is based on that described in a diversity control oriented genetic algorithm or DCGA (Shimodaira, 1997). Similar to a steady-state genetic algorithm, the DCGA operates by merging the parent and offspring populations together where an appropriate number of individuals are subsequently extracted from the merged population. However, the DCGA also dictates the elimination of duplicated individuals in the merged population prior to the cross-generational survival selection process. Two

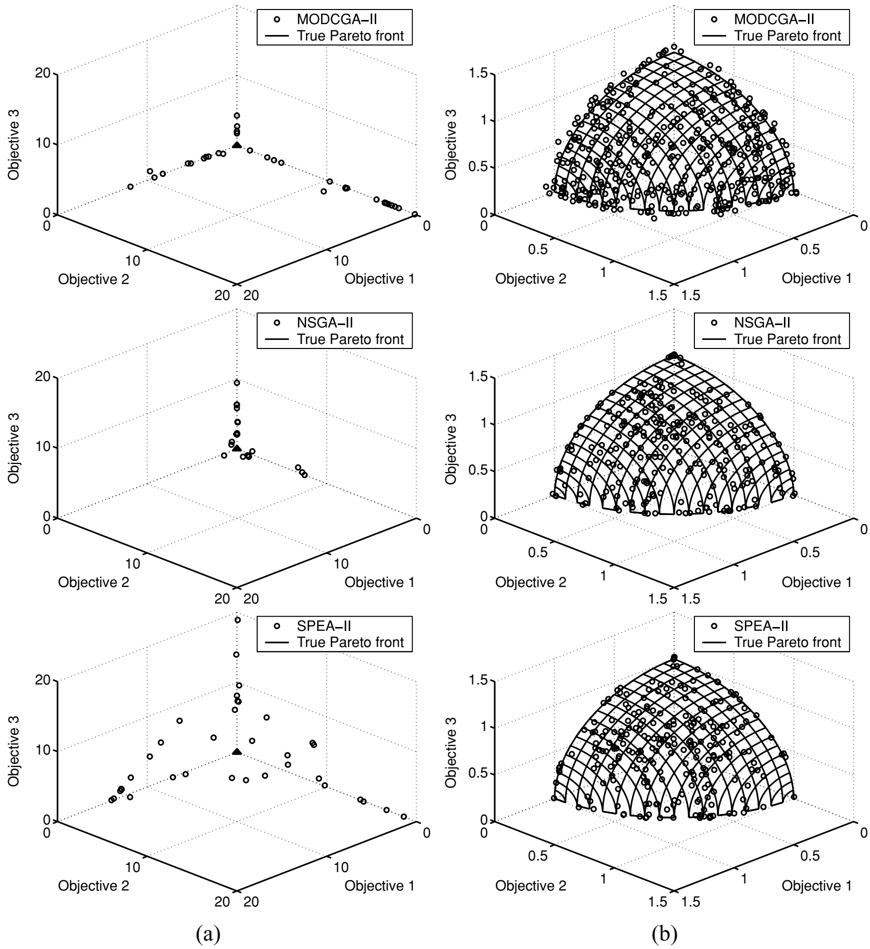


Fig. 13 Optimisation results on the three-objective problems (a) DTLZ1 (b) DTLZ2

cross-generational survival selection techniques can be applied: a deterministic one and a stochastic one. In the case of the cross-generational deterministic survival selection (CDSS), the remaining individuals in the merged population are sorted according to their fitness in descending order where the selection is carried out in the top-down fashion. In contrast, in the case of the cross-generational probabilistic survival selection (CPSS) after the selection of the elite individual, the non-elite individuals are selected based on two criteria: their fitness and their similarity to the elite individual. A non-elite individual with the genomic structure that is quite different from that of the elite individual would have a higher survival probability. In addition to the use of the CDSS and CPSS techniques, the DCGA also allows an introduction of newly generated individuals to the population in the situation when the survival selection is completed prior to the required population size is reached.

The diversity control operation explained above has been successfully tested on both single-objective and multi-objective benchmark problems. In the case of

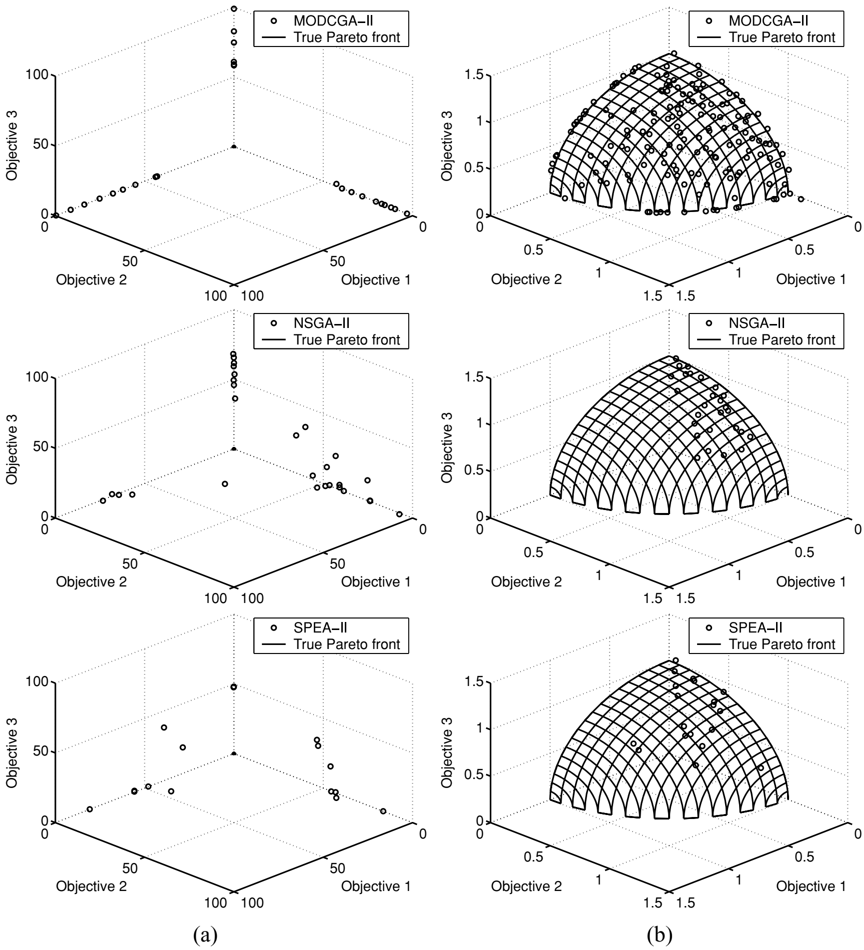


Fig. 14 Optimisation results on the three-objective problems (a) DTLZ3 (b) DTLZ4

single-objective benchmarking, two benchmark problems are utilised: onemax (Ackley, 1987) and royal road R_1 (Mitchell et al., 1992) functions. These two functions are selected for the paper since the original DCGA benchmarking results reported in Shimodaira (2001) do not extensively cover the test results on scalable benchmarking problems at which the fitness of an individual is calculated directly from the genotypic description of the chromosome. This issue is important since the performance of genetic algorithms for this class of problems would be independent from the chromosome encoding technique and only depends on the genetic operators used. The analysis indicates that the DCGA with either the CDSS technique or the CPSS technique with any settings for the shape (c) and exponent (α) coefficients can be used to solve onemax problems. Recall that by either setting $c = 1.00$ or $\alpha = 0.00$, the CPSS technique is transformed into the CDSS technique. In contrast, the DCGA with the CPSS technique where $c = 0.25$ and $\alpha = 0.80$ is found to be suitable for solving the royal road R_1 problems. With the use of a fitness distance correlation measurement

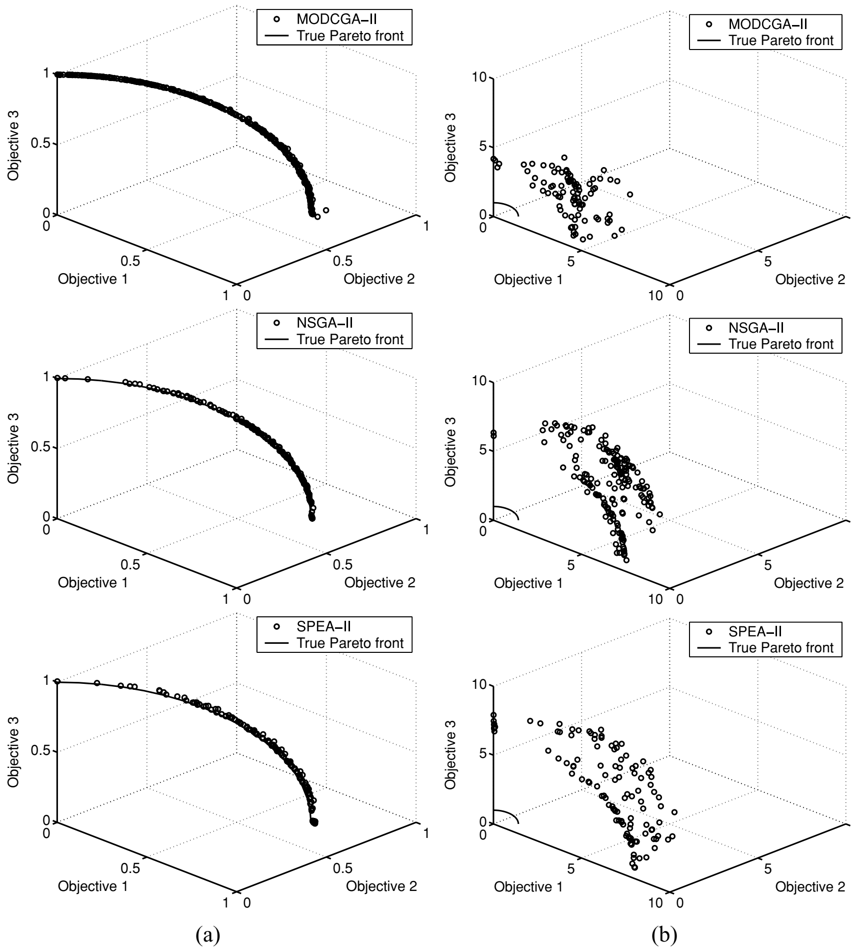


Fig. 15 Optimisation results on the three-objective problems (a) DTLZ5 (b) DTLZ6

(Jones and Forrest, 1995), it can be concluded that the DCGA is stable and fast enough to solve straightforward and difficult benchmark problems. The DCGA has also been benchmarked against the technique that involves the transformation of single-objective problems into multi-objective problems and the population diversity is maintained via preserving non-dominated solutions through a multi-objective genetic algorithm with elitism (Abbass and Deb, 2003; Jensen, 2003). The DCGA is proven to be better than the technique involving such a transformation in both cases of the onemax and royal road R_1 functions.

Moving onto the case of multi-objective optimisation using a genetic algorithm with diversity control, which is uniquely referred to as a modified multi-objective diversity control oriented genetic algorithm or MODCGA-II. The proposed algorithm differs from the MODCGA described in Sangkawelert and Chaiyaratana (2003) in the sense that the MODCGA-II performs diversity control via similarity measurement in objective space, which makes the diversity control operation becomes independent from the chromosome encoding scheme, and the use of a preserved non-dominated solution archive is also included. Six scalable benchmark problems described in Deb et al. (2005) are utilised in this case. In addition, the criteria used to assess the algorithm performance include the closeness of non-dominated solutions to the true Pareto front and the distribution of the solutions across the front (Zitzler et al., 2000). The analysis indicates that the MODCGA-II with the CPSS technique where $c = 0.75$ and $\alpha = 0.20$ can produce non-dominated solutions that are better than that generated by the non-dominated sorting genetic algorithm II or NSGA-II (Deb et al., 2002a) and the improved strength Pareto evolutionary algorithm or SPEA-II (Zitzler et al., 2002) when the number of objectives in the benchmark problems is limited to two. On the other hand, when the number of objectives increases to three, the MODCGA-II performance is worse than that of the NSGA-II and the limitation of the proposed algorithm is hence identified.

Based on the overall results, it is recommended that the diversity control with the CPSS technique using the specific settings described above is the most suitable approach for both single-objective genetic algorithm and multi-objective genetic algorithm when the number of objectives is two. This is because it has a reasonably high performance in comparison to that obtained using other diversity preservation techniques reported in early literature. Furthermore, the results also suggest that some parameter settings for the CPSS technique can lead to a situation where the diversity control operator is unable to perform to its full potential. As a result, the use of diversity control with the CPSS technique in genetic algorithm has to be done with care.

Acknowledgments This work was supported by the Thailand Research Fund (TRF) and the Commission on Higher Education.

References

Abbass, H.A. and K. Deb. (2003). "Searching under Multi-Evolutionary Pressures." *Lecture Notes in Computer Science* 2632, 391–404.

Ackley, D.H. (1987). *A Connectionist Machine for Genetic Hillclimbing*. Kluwer Academic, Boston, MA.

Baker, J.E. (1989). *An Analysis of the Effects of Selection in Genetic Algorithms*, Ph.D. thesis, Computer Science Department, Vanderbilt University, Nashville, TN.

Boonlong, K., N. Chaiyaratana, and S. Kuntanapreeda. (2002). "Time Optimal and Time-Energy Optimal Control of a Satellite Attitude Using Genetic Algorithms." In *The 2002 ASME International Mechanical Engineering Congress and Exposition*. New Orleans, LA, pp. DSC-33436/1–8.

Boonlong, K., N. Chaiyaratana, and S. Kuntanapreeda. (2004). "Optimal Control of a Hysteresis System by Means of Co-Operative Co-Evolution." *International Journal of Computational Intelligence and Applications* 4(4), 321–336.

Deb, K. (2001). *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. (2002a). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6(2), 182–197.

Deb, K., L. Thiele, M. Laumanns, and E. Zitzler. (2002b). "Scalable Multi-Objective Optimization Test Problems." In *Proceedings of the 2002 Congress on Evolutionary Computation*, Honolulu, HI, pp. 825–830.

Deb, K., L. Thiele, M. Laumanns, and E. Zitzler. (2005). "Scalable Test Problems for Evolutionary Multi-Objective Optimization." In A. Abraham, L.C. Jain, and R. Goldberg (eds.), *Evolutionary Multi-objective Optimization: Theoretical Advances and Applications*, Springer, London, UK, pp. 105–145.

Eshelman, L.J. (1991). "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination." In G.J.E. Rawlins (ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, vol. 1, pp. 265–283.

Fonseca, C.M. and P.J. Fleming. (1993). "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization." In *Proceedings of the Fifth International Conference on Genetic Algorithms*, Urbana-Champaign, IL, pp. 416–423.

Fonseca, C.M. and P.J. Fleming. (1995). "Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction." In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sheffield, UK, pp. 45–52.

Fonseca, C.M. and P.J. Fleming. (1998). "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms–Part 1: A Unified Formulation." *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans* 28(1), 26–37.

Goldberg, D.E. (1989). *Genetic Algorithms: In Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Jensen, M.T. (2003). "Guiding Single-Objective Optimization Using Multi-Objective Methods." *Lecture Notes in Computer Science* 2611, 268–279.

Jones, T. and S. Forrest. (1995). "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms." In *Proceedings of the Sixth International Conference on Genetic Algorithms*, Pittsburgh, PA, pp. 184–192.

Maneeratana, K., K. Boonlong, and N. Chaiyaratana. (2004). "Multi-Objective Optimisation by Co-Operative Co-Evolution." *Lecture Notes in Computer Science* 3242, 772–781.

Mauldin, M.L. (1984). "Maintaining Diversity in Genetic Search." In *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, pp. 247–250.

Mitchell, M., S. Forrest, and J.H. Holland. (1992). "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance." In *Proceedings of the First European Conference on Artificial Life. Toward a Practice of Autonomous Systems*, Paris, France, pp. 245–254.

Mori, N., J. Yoshida, H. Tamaki, H. Kita, and Y. Nishikawa. (1995). "A Thermodynamical Selection Rule for the Genetic Algorithm." In *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, Perth, Australia, pp. 188–192.

Pimpawat, C. and N. Chaiyaratana. (2004). "Three-Dimensional Container Loading Using a Cooperative Co-Evolutionary Genetic Algorithm." *Applied Artificial Intelligence* 18(7), 581–601.

Potter, M.A. and K.A. De Jong. (2000). "Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents." *Evolutionary Computation* 8(1), 1–29.

Sangkawelert, N. and N. Chaiyaratana. (2003). "Diversity Control in a Multi-Objective Genetic Algorithm." In *Proceedings of the 2003 Congress on Evolutionary Computation*, Canberra, Australia, pp. 2704–2711.

Shimodaira, H. (1996). "A New Genetic Algorithm Using Large Mutation Rates and Population-Elitist Selection (GALME)." In *Proceedings of the Eighth IEEE International Conference on Tools with Artificial Intelligence*, Toulouse, France, pp. 25–32.

Shimodaira, H. (1997). "DCGA: A Diversity Control Oriented Genetic Algorithm." In *Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Glasgow, UK, pp. 444–449.

Shimodaira, H. (2001). "A Diversity-Control-Oriented Genetic Algorithm (DCGA): Performance in Function Optimization." In *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, Korea, pp. 44–51.

Weerayuth, N. and N. Chaiyaratana. (2002). "Closed-Loop Time-Optimal Path Planning Using a Multi-Objective Diversity Control Oriented Genetic Algorithm." In *Proceedings of the 2002 IEEE International Conference on Systems, Man, and Cybernetics*, Hammamet, Tunisia, pp. WA2G4/1–7.

Whitley, D. (1989). "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproduction Trials is Best." In *Proceedings of the Third International Conference on Genetic Algorithms*, Fairfax, VA, pp. 116–121.

Zitzler, E., K. Deb, and L. Thiele. (2000). "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *Evolutionary Computation* 8(2), 173–195.

Zitzler, E., M. Laumanns, and L. Thiele. (2002). "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization." In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty (eds.), *Evolutionary Methods for Design, Optimisation and Control*, International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, pp. 95–100.