Statistical Exploratory Analysis of Mask-fill Reproduction Operators of Genetic Algorithms

Hari Mohan Pandey

Department of Computer Science, Edge Hill University, Ormskirk, UK, Pandeyh@edgehill.ac.uk

Marcello Trovati

Department of Computer Science, Edge Hill University, Ormskirk, UK, marcello.trovati@edgehill.ac.uk

Nik Bessis

Department of Computer Science, Edge Hill University, Ormskirk, UK, nik.bessis@edgehill.ac.uk

Abstract— To be successful, a search algorithm needs to balance exploration and exploitation. In Genetic Algorithms (GAs) this is achieved through proportionate selection of individuals and reproduction operators. GAs can suffer premature convergence, when the diversity of the population decreases over time and search gets trapped in a local optimum, returning a very poor quality solution. Mask-fill reproduction operators utilizes bit-masking oriented data structure (BMODS) that maintains a good ratio between exploration and exploitation. Mask-fill reproduction operators have been utilised in various applications, but the exploration and exploitation ability of mask-fill reproduction operators have not been compared against other reproduction operators. This paper describes a rigorous and practical statistical methodology for the exploratory analysis of the mask-fill reproduction operators. First, the issues of robust experimental design and setting the control parameters for implementing a GA is addressed. Second, the impact of various reproduction operator combinations are analysed. In this study, three crossover operators and five mutation operators are considered which creates fifteen crossover-mutation operator combinations. Third, the methodology is demonstrated by considering grammatical inference (GI) problem as domain of inquiry. A hybrid genetic algorithm integrated with Sequitur algorithm (GAWS) is proposed for GI. Numerical results are presented to describe the effect of crossover-mutation operator combinations. The performance of the proposed GAWS is compared against the state-of-the-art algorithms. Statistical test are conducted to determine the performance significance of both crossover-mutation operator combinations and the proposed GAWS algorithm.

31 **Keywords**—Bit-Mask-Oriented Genetic Algorithm, Mask-fill Crossover Operator, Mask-fill
32 Mutation Operator, Genetic Algorithm, Exploration and Exploitation.

33

34 **Terms used:**

35 *Ndv*: Number of design variables

36 *L*: Binary string full length

37 *R*: Bit resolution related to each design variable

38 *C*: a random cut value ranging from 1 to L-1

39 *RND*: Random

40 *Sum (i)*: sum of the bit resolutions up to the current i^{th} element

41 *P1/P2*: Parent strings

42 *tr*: bit transition from P1 and P2

43 *R (i)*: i^{th} bit resolution related to each design variable

44 *POP_SIZE*: population size

45 *CS*: chromosome size

46 *CR*: crossover rate

47 *MR*: mutation rate

48 *Gmax*: maximum number of generations

49 *FV*: Fitness function value

50 *TR*: threshold value

51 *CM*: crossover mask

52 *MM*: mutation mask

53 *T1, T2, T3 and T4*: Temporary individual

54 *OS*: offspring

55 *PS*: positive samples

56 *NS*: negative samples

57 Δ : increase in the population size

58 *VLG*: Variable block gene from middle of the chromosome

59 *FLG*: Fixed block gene from middle of the chromosome

60 *rand*: random value

61 *CI*: cut crossover mask-fill

62 C2: bit by bit crossover mask-fill
63 C3: local cut crossover mask-fill
64 M0: no mutation
65 M1: block copier with random length mutation
66 M2: mutation mask-fill
67 M3: block copier fixed length mutation
68 M4: inverse mutation
69 $Ci-Mj$ ($i=1, 2, 3$ and $j = 0, 1, 2, 3, 4$): represent the reproduction operator (crossover and mutation)
70 combinations

71 1. Introduction

72 Genetic Algorithms (GAs) are one of the best-known Evolutionary Algorithms (EAs), which
73 are drawn on Darwin's principle of the "*survival of the fittest*" [1] [2]. The objective of a GA is to
74 find an optimal solution or at least a good solution given that optimality cannot normally be
75 guaranteed. A GA may converge to a poor solution, so called premature convergence where the
76 diversity of the population decreases over generations [3].

77 Exploration and exploitation play a central role of problem solving by search and a successful
78 search algorithm needs to find a balance between the two [4] [5]. Exploration aims to visit entirely
79 new regions of a search space, whereas exploitation focuses on those regions of a search space
80 recently visited [4].

81 The current scientific literature shows that selection and reproduction operators (crossover and
82 mutation) are able manage a good balance between exploration and exploitation. Control of the
83 parameter settings played a vital role in achieving a good ratio between exploration and
84 exploitation. If crossover and mutation rates are very high, much of the search space will be
85 explored, but it comes at the cost of possibly missing good solutions. In this situation, a GA will
86 tend towards being a random search. On the other hand, if crossover and mutation rates are low,
87 large part of the search space remain unexplored. In this scenario, a GA will tend to become a hill
88 climbing algorithm. The optimization of crossover and mutation rates cannot be done
89 independently. Hence, combinations of crossover and mutation rates are usually experimented
90 with. Every GA developer (or user) faces the challenge of finding a suitable combination of
91 control-parameter settings that are most likely to produce the best results. In addition, many

different crossover and mutation operators exist and users face the further challenge of which ones to select.

The fine tuning of GA control parameters is important in order to present an algorithm that is able to return good quality solutions. Rojas et al. [6] analyzed the dynamics of a GA considering two viewpoints (capacity of GA to obtain a local/global optimum and diversity of the population). Srinivas and Patnaik [7] Currently, the analysis of crossover and mutation operators has largely been presented independently. Investigations are lacking which shows the combined effect of both crossover and mutation operators in finding a good balance between exploration and exploitation in the context of a GA.

Crossover and mutation operators are the main contributors in achieving a good ratio between exploration and exploitation. Many types of crossover and mutation operators have been proposed, but finding a suitable crossover/mutation combination is critical. Iuspa et al. [8] proposed mask-fill crossover and mutation operators that use a bit-masking oriented data structure (BMODS). BMODS utilizes a 2-D array to store the population genetic inheritance through two new arrays referred to as crossover mask (CM) and mutation mask (MM). Mask-fill reproduction operators have further been utilized for various applications [9] [10] [11] [12]. The current research focuses on the performance comparison of mask-fill reproduction operators. Three crossover operators (cut crossover mask-fill, bit-by-bit crossover mask-fill and local cut mask-fill) and five mutation operators (No Mutation, Block Copier with Random Length, Mutation Mask-Fill, Block Copier with Fixed Length and Inverse Mutation) are adapted. The outcome of this research is to classify the crossover and mutation operator combinations based on their computational abilities in solving problems. We also monitor the exploration and exploitation quality of mask-fill reproduction operators. We believe that the current research is of great relevance to the EA community, particularly to GAs, and it will motivate researchers to utilize the mask-fill reproduction operators to develop new more powerful optimization algorithms.

Motivated by the aforementioned discussion, this paper proposes a rigorous and practice statistical methodology for assessing the impact of mask-fill reproduction operators. It also present a hybrid GA for grammatical inference (GI). GI, a subfield of machine learning is the process of learning grammars (extracted from artifacts written in some unknown language) from the corpus. GI finds numerous applications in natural language processing, speech recognition, image processing and computational biology [13] [14] [15] [16] [17]. A comprehensive discussion on the

state-of-the-art of GI is presented in Section 4. In particular, the key contributions of this paper are described as follows:

- Firstly, the issues of robust experimental design and setting the control parameters for implementing the GAs is addressed.
- Secondly, the impact of crossover-mutation operator combination with regards to its exploration ability of the search space is analyzed. In this study, three crossover operators (cut crossover mask-fill, bit-by-bit crossover mask-fill and local cut mask-fill) and five mutation operators (No Mutation, Block Copier with Random Length Mutation, Mutation Mask-Fill, Block Copier with Fixed Length and Inverse Mutation) are considered which creates fifteen crossover-mutation operator combinations.
- Thirdly, GI problem is a NP hard problem is considered for demonstration of our methodology. A hybrid GA integrated with sequitur algorithm (GAWS) is proposed for GI.
- Fourth, two-fold experimentations are conducted. First, numerical results are presented to describe the effect of crossover-mutation operator combinations. Then, statistical tests are conducted to determine the significance and the best crossover-mutation operator combination is identified. Second, the performance of the proposed GAWS is compared against the state-of-the-art algorithms such as classical genetic algorithm (CGA) [18], improved tabular representation (ITBL) [19] and genetic algorithm with minimum description length principle (GAWMDL) [12].
- Finally, a comprehensive discussion is presented to explain the results respectively for crossover - mutation operator combination and for the proposed GAWS. The rational on “*why one crossover-mutation operator combination is better or worse than the other?*” is also highlighted. A detailed statistical tests result are provided in *Appendix A* for reference.

The rest of the paper is structured as follows: Section 2 presents related work; Section 3 discusses the bit mask oriented genetic algorithm (BMOGA); Section 4 presents a modified version of BMOGA integrated with sequitur algorithm (proposed hybrid algorithm is referred as GAWS) which incorporates GI; the reproduction operators adapted in this research are discussed in Section 5; Section 6 shows experimental studies, results and statistical test’s result; Finally, conclusions and future research directions are discussed in Section 7.

2. Related work

Holland's theory [18] supports the use of crossover and it is accepted as the more powerful of the two operators (crossover vs mutation). De Jong [20] showed the exploratory power of crossover operators. Spears and De Jong [21] presented an empirical comparison between crossover and mutation operators and concluded that mutation alone is not sufficient to find a global solution. Proponents of Evolutionary Programming (EP) and Evolutionary Strategies (ES) believe that the mutation operator is more powerful than crossover. Fogel et al. [22] demonstrated the effect of mutation and selection to evolve a finite state automaton for various tasks. Rechenberg [23] proposed ES, which focused on mutation as a key operator. Baeck et al. [24] and Schwefel [25] showed the importance on an adaptive mutation rate for function optimization. Fogel and Atmar [26] extended the work in [25] and made a much stronger claim: "*crossover operator has no general advantage over mutation operator*".

Applying mutation alone may convert a GA into a random search, whilst applying the crossover operator alone generates sub-optimal solutions. If selection, crossover and mutation operators are applied together then it may convert a GA to a noise tolerant hill climbing search [27]. Spears [27] argued that crossover and mutation operators have different properties and they complement each other during an evolutionary process. The success of a GA can be improved through the fine tuning of control-parameters. Examples of these parameters include population size, chromosome size, crossover and mutation rates and step-size. The values of these control-parameter are most commonly set through trial and error (based on recommendation of related problems) using statistical methods or using typical values found in other research.

The existing scientific literature reveals that high diversity (differences among the individuals in a population at genotype or phenotype level) greatly contributes to the success of a GA [28]. McPhee and Hopper [29] says that "*fundamentally the progress in evolution depends on the existence of variation among the individuals*". Lack of population diversity leads to stagnation as the system finds itself trapped in local optima.

Different measures [30] [31] [32] have been utilized for diversity, but there exists no single measure that is suitable for every problem. Moreover, the diversity measure discussed in [30] [31] [32], if applied to guide an evolutionary process, then it requires an investigation of correlation (a correlation between diversity measure and fitness): "*if a positive correlation exists*". This is not always the case - and therefore, no improvement in the fitness function is seen [31]. We must take

into account that diversity is only roughly related to exploration and exploitation. Hence, high diversity does not guarantee a good balance between exploration and exploitation. In addition, high diversity does not guarantee that an individual “*fits*” rather different from each other [33]. Diversity is useful to determine “*fit*” individuals and it is achieved through “*useful diversity*” – a prerequisite to prevent premature convergence [34]. Promoting diversity at every stage of evolution might be more productive, if exploration of the search space is required. Burke et al. [31] showed the relationship between population diversity, exploration and exploitation of the search space, but this topic still requires more research.

Eiben and Schippers [35] questioned the general belief of exploration and exploitation: “*exploitation is done by selection, whilst search operators (crossover and mutation) are for exploration*”. They concluded that there is no common accepted perception for exploration and exploitation of the search space. Hence, more research is required for a deeper understanding of the fundamentals of an evolutionary search process. Wong et al. [36] proposed an oversimplified view: “*GA maintains a balance between exploration (by crossover) and exploitation (by mutation), so as to increase the chance of finding global optimum*”. Wong et al. [36] presented treatments for exploration and exploitation through search operators, whilst other factors such as a selection process (drives searching towards the best individual regions) was ignored. Back [37] showed the importance of the selection process and claimed that it is used to control the level of exploration and exploitation following the convention: “*higher selection pressure directs towards more exploitation, whereas lower selection pressure pushes towards exploration*”. Goldberg and Deb [38] presented the concept of accurate selection pressure – an essential element used for a good balance between exploration and exploitation.

Mutation operators play an essential role in GAs as it randomly modifies an individual using a mutation rate, and increases structural diversity. Hence, mutation operators are an exploratory operator utilized to recover genetic diversity, which is generally lost during a selection process. Conversely, mutation operators also show the nature of an exploitation operator as it preserves the genetic material. The nature of a mutation operator differs for different EAs. For ES, it is considered as an exploratory operator, whilst for a GA it works as an exploratory operator based on locality property [39]. A crossover operator is utilized to produce a better offspring by combining two parent strings. The idea of the crossover operator is derived from an exchange of information between selected best individuals that produces an even better offspring. Hence, a

crossover operator works as an exploitation operator [39]. However, a crossover operator has been implemented to generate an offspring in exploration as well. De Jong and Spears [40] demonstrated the exploratory power of crossover operators.

Apart from - selection and reproduction operators there exist other factors: population size and representation of individuals. These have also been shown to have an impact on exploration and exploitation [41] [42] [27] [42] [43]. The existing scientific literature shows that “*increasing population size explores the search space more and vice-versa*” [29], but without a proper balance between exploration and exploitation even resizing the population is worthless. Moreover, this process is time consuming. In addition, larger population might affect other operators. Spears [43] and Smith and Smuda [44] presented an example on this view: “*uniform crossover operator outperformed 2-points cut crossover operator on a small population and vice-versa is true too*”. Representation of an individual is also an influencing factor as it affects the how crossover and mutation operators operate. However, in some situation representation independence is achieved. For example, geometric crossover or mutation operators originate from the theories of topology and geometry [45]. Existing scientific literature [41] [46] has demonstrated the importance of representation techniques in maintaining a good ratio between exploration and exploitation.

The above discussion can be summed up with the following key points: firstly, it is hard to say that an individual generated through crossover or mutation operation will lie in an exploration or exploitation region. Secondly, there is thin line, which is blurred between exploration and exploitation. Thirdly, both population size and representation of an individual are important for exploration and exploitation.

3. Bit Mask Oriented Genetic Algorithm

Pandey et al. [9] proposed a BMOGA that utilizes mask-fill reproduction operators in conjunction with a Boolean based procedure for offspring generation. BMOGA uses three crossover operators (cut crossover mask-fill operator, bit-by-bit crossover mask-fill and local cut mask-fill) and a mutation operator. The reproduction operation is performed by creating random masks respectively for crossover and mutation operators. Algorithm-1 presents BMOGA.

Algorithm-1. Bit Mask Oriented Genetic Algorithm (BMOGA)

Input: POP_SIZE, CS, CR, MR and Gmax

1   Begin

2. Generate random initial population POP_SIZE
3. Determine FV of each chromosome.
4. While ($FV > TR$) and ($TR = Gmax$) Then
5.   Selection operation.
6.   Generate random CM and MM
7.   Apply mask-fill crossover operation.
8.   Apply mask-fill mutation operation.
9.   Offspring generation   (Algorithm-2)
10.   Arrange FV in increasing order among parents and offspring populations.
11.   Select the better chromosomes as parents of the next generation.
12.   End While
13.   End

---

242   BMOGA generate offspring through a Boolean procedure based mixture $f(P1, P2, CM, MM)$.

243   Algorithm-2 shows this procedure.

Algorithm-2. Offspring generation

Input: P1, P2, CM and MM

1.   Begin
  2.   Set $CM \leftarrow$ initialize crossover mask
  3.   Set $MM \leftarrow$ initialize mutation mask
  4.   Set $T1 \leftarrow P1 \text{ AND } CM$
  5.   Set $T2 \leftarrow P2 \text{ AND } (\text{NOT } CM)$
  6.   Set $T3 \leftarrow P2 \text{ AND } CM$
  7.   Set $T4 \leftarrow P1 \text{ AND } (\text{NOT } CM)$
  8.   Set $OS1 \leftarrow T1 \text{ OR } T2$
  9.   Set $OS2 \leftarrow T3 \text{ OR } T4$
  10.   Update $OS1 \leftarrow OS1 \text{ XOR } MM$
  11.   Update $OS2 \leftarrow OS2 \text{ XOR } MM$
  12.   End
-

Figure 1. Flowchart of the BMOGA.

The working of the BMOGA is discussed using its flowchart as presented in Figure 1. The process begins by generating random population and initializing various control parameter values like chromosome size, population size, crossover probability and mutation probability etc (Step 1). Step 2 is responsible for determining the fitness function value. Selection of choromose is done at Step 3 whilst reproduction operations (crossover and mutation) and offspring generation are performed at Step 4. The replacement of the old population with the new population is done at Step 5. Here, the term new population represents the population after implementing the reproduction operations. This process (Step 2 – Step 6) will keep on running iteratively until the termination condition meets. Two termination conditons have been defined in Figure 5. First, the

best result achieved which is higher than the threshold value. Second, Total number of runs equal to the maximum number of generations.

4. Grammatical Inference

This section shade light on GI. First, state-of-the-art work on GI is presented in Section 4.1. Then, discussion on the proposed algorithm is given in Section 4.2.

4.1 State-of-the-Art on GI

The process of the GI can be described as follows: “*it is an idealized learning procedure to acquire grammars on the basis of the evidence about a language [57][58]*”. The first learning model was proposed by Gold [59] which addressed the question “*Is the information sufficient to determine which of the possible languages the unknown language belongs to?*”. Gold’s learning model suffered because sufficient information about identification of correct grammar was not provided. Angluin [60] suggested Tell Tales to address the challenges of the Gold’s learning model [59]. Though Gold laid the foundation of learning model, but Bunke and Sanfeliu [61] proposed the 1st usable learning model which was: (a) suitable for positive samples only; (b) not fit to deal with negative samples; (c) not suitable for noisy data; and (d) not fit for a finite state machine. Angluin [62] presented a teacher query learning model is also known as an oracle. Here, oracle knows the answer, hence, it is considered as supervised learning model. This new learning model has utilized an inference system for responding to a particular type of query. The key challenge with this model was implementing an oracle which requires lot of pre-information.

Valiant [63] proposed a probably approximately correct (PAC) learning model. PAC learning model combines the merits of both Gold’s [59] and Angluin’s [62] learning models. PAC learning model suffered due to the two key reasons are: (a) inference algorithm must learn in polynomial time under all distributions, but believing on this assumption is too strict in reality; and (b) not fit for negative and NP hard equivalence results. To address these challenges, Li and Vitanyi [64] suggested a variant of the PAC learning model where simplicity was measured through Kolmogorov complexity.

Delgado and Pegalajar [65] used a multiobjective GA to analyze the optimal size of a recurrent neural network (RNN) from positive and negative examples. The drawback with neural network (NN) based GI system is reconstruction of the unknown function from the connection of a trained network [62] [77]. Wyard [66] showed the impact of different grammatical representations, with

experimental results reporting that an evolutionary algorithm (EA) using the standard context free grammar (CFG) (in BNF) outperformed others. Javed et al. [67] utilized a genetic programming (GP) approach for learning the CFG. This was an extension of the work conducted by Wyard [66]. Choubey and Kharat [68] showed a sequential chromosome structuring approach using GA for the GI where coding and decoding of the binary coded chromosomes into terminals and non-terminals and vice-versa.

A GA based GI library was proposed by Pandey [69] and Choubey et al. [70]. Pandey et al. [71] presented a case study on GI which includes a basic GA along with simple crossover (one and two-point) and bit inversion operators. Hrnčić et al. [72] and Hrnčić and Mernik [73] implemented a memetic algorithm (MA) for GI, which assists domain experts and software language engineers to develop domain-specific languages (DSLs) by automatically producing grammars.

Sakakibara and Kondo [74] presented a GA based approach for CFG induction from a finite set of positive and negative examples. They utilized tables similar to the parse table that reduces the partitioning problem of non-terminal and then the GA was implemented to address the partitioning problem. Jaworski and Unold [19] brought some improvement on the initial population block size manipulation, block deletes specialized operator and modified fitness function and experimentally showed that the improved tabular representation algorithm (ITBL) is not vulnerable to block size and population size, and it is capable of finding solutions faster. Bhalse and Gupta [75] also utilized ITBL for GI.

Stevenson and Cordy [62] [77] presented a comprehensive review on the GI and classified its contributions in two groups, namely, theorists and empiricists. Theorists group deals with languages classes and learning models to decide what is learnable and how efficiently it can be learned whilst empiricists group tackle practical problem. A detailed survey of various GI algorithms is presented in [62] [76]. Inductive inference is the process of generalizing from the input. These surveys have presented various applications, highlighted merits and demerits of different GI algorithms. In this paper, GAWS has been proposed which addresses three key challenges: (a) it is capable of avoiding premature convergence during the evolutionary process of the GA; (b) It can handle both positive and negative corpus; and (c) It utilizes merits of the sequitur algorithm, therefore, it can manage temporary grammar rules that are generated during evolutionary process.

Vázquez et al. [77] showed the application of the GI to infer the automation in developing a model that recognizes semantic components of spontaneous dialogues of telephonic queries.

Wieczorek et al. [78] proposed a GI methodology referred as the identification of the CFGs based on the presentation of samples. The simulation result revealed that the GI by answer set programming was much faster than the original SAT encoding method. Kovačević et al. [79] presented semantic inference - an extended version of the GI. Authors [79] utilized evolutionary computation method to explore and exploit the enormous search space that appears in the semantic inference. Authors [80] proposed a GA based GI system which is known as weighted grammar-based classifier system (wGCS) for inducing weighted CFG (WCFG) from the given data. In [80], weights of the rules were estimated using inside-outside constructive estimation algorithm. It was noted that wGCS was capable of learning CFGs from both positive and negative samples. Andrés et al. [81] proposed a model for the automatic generation of written dialogues using the GI method. Their model showed the capability of automatically recognizing grammars from a set of dialogues given in a training dataset.

4.2 Proposed Grammatical Inference Algorithm

This section discusses a hybrid GA integrated with sequitur algorithm (GAWS) is proposed for GI.

Figure 2. Block diagram presenting the genetic algorithm integrated with sequitur algorithm for grammatical inference.

A block diagram for the GI is depicted in Figure 2. The proposed GI system has three key components, namely, input corpus, process (learning system) and output in the form of context free grammars (CFGs). For the learning system, BMOGA integrated with the Sequitur algorithm is implemented. Algorithm-3 which is a modified version of Algorithm-1 is implemented for getting the best output. It starts generating an initial random population of grammars. To generate

initial grammars, the Sequitur algorithm [47] is used detects repetition in a sample and factor the sample out by creating grammar rules. There exit various GA based GI algorithms have been proposed but they have their own strengths and weaknesses (Section 4.1). The proposed GAWS algorithm has a greater potential due to two key reasons. First, it uses mask-fill reproduction operators to perform crossover and mutation operations and then Boolean based procedures have been implemented to generate offspring's (Algorithm 2). This process introduces diversity in the population and avoid premature convergence during evolutionary process of the GA. Second, sequitur algorithm is implemented to detect repetition in the sample while generating the grammar. This helps in handling large number of temporary grammar rules that are generated during the evolutionary process.

Algorithm-3. GA with Sequitur (GAWS) for Grammatical Inference

---

Input: PS, NS, POP_SIZE, CS, CR, MR and Gmax

Output: Best grammars

1.   Begin
  2.   generate random initial population P
  3.   For all samples S_i in PS do
  4.   generate $G_i \leftarrow \text{SEQUITUR}(S_i)$
  5.   Add G_i to P
  6.   End for
  7.   Current_G $\leftarrow 0$ (Algorithm-1)
  8.   While (FV > TR) and (TR = Gmax) Then
  9.   Selection operation.
  10.   Generate random CM and MM
  11.   Crossover ($PS, NS, P1, P2, CR$)
  12.   Mutation (PS, NS, P, MR)
  13.   Offspring generation
  14.   Generalization (PS, NS, P)   (Algorithm-4)
  15.   Select best POP_SIZE grammars
  16.   Current_G++
  17.   End While
  18.   End
-

We noticed that Sequitur algorithm does not generalize productions. Hence, we have implemented Algorithm-4 for generalization of the training data.

Algorithm-4. Generalization (PS, NS, P)

Input: PS, NS, POP_SIZE, P

```

1.   Begin
2.       For  $j \leftarrow 1 \dots \text{POP\_SIZE} + \Delta$ 
3.            $G_{\text{POP\_SIZE}+\Delta+j} \leftarrow \text{generalize}(G_j)$ 
4.            $FV \leftarrow \text{evaluate}(G_{\text{POP\_SIZE}+\Delta+j}, PS, NS)$ 
5.               If  $FV > 0$ 
6.                   Add  $G_{\text{POP\_SIZE}+\Delta+j}$  to  $P$ 
7.               End if
8.           End for
9.   End

```

After the initial population is generated, the evolution part executes (Step 7 – 17 of Algorithm-3). First, we apply selection to select two parents from the population. Next, the population grammars undergo transformation through crossover and mutation operations as adapted (Section 5). Algorithm-4 is an important part of Algorithm-3. It checks each grammar for repeating symbols, the nested structures and repeating right hand sides (RHSs) [48]. A new generated grammar inserted into the population or when existing grammar is changed, a parser (LR (1) in this case) is produced and tested on all input samples. The positive sample parsed through LR (1) is the grammar fitness, whilst negative samples, if available, are discarded. Roulette wheel selection selects the best POP_SIZE grammars for the next generation. The population of grammars are not fixed and it grows during the evolutionary cycle, but at the end of an evolutionary cycle, only POP_SIZE grammars survive. Algorithm-3 runs until it finds a TR (TR indicates a highest ranked solution) or Gmax.

5. Reproduction Operators Adapted

For this study, three crossover operators (cut crossover mask-fill, bit-by-bit crossover mask-fill and local cut mask-fill operators) and five mutation operators (no mutation, block-copier with random length mutation, mutation mask-fill, block-copier with fixed length and inverse mutation)

are utilized. Mask-fill reproduction operators uses Boolean based procedure mixture and BMODS which is able to handle both continuous and discrete variables [8] [10] [11]. Previous work [9] [12] revealed that mask-fill reproduction operators have the ability to avoid premature convergence. Although, mask-fill reproduction operators showed effective performance, but the performance of the mask-fill reproduction operators have not been tested against any other reproduction operators. In the upcoming subsections, reproduction operators adapted for this study are discussed with illustrative examples.

5.1 Crossover operators adapted

Previous research have demonstrated the exploration ability of mask-fill crossover operators [2] [8] [9] [10] [11]. Three mask-fill crossover operators are discussed in this section.

5.1.1 Cut crossover mask-fill operator

A single cut crossover operator is a well-known mating method. It is utilized to create a new offspring by applying a random of parent strings and subsequent operation on the resulting substrings. Algorithm-5 presents the cut crossover mask-fill operator.

Algorithm-5. cut_crossover_mask_fill(P1, P2)

Input: P1, P2, L, Sum (i) and Ndv

---

1. Begin
2. Set $C \leftarrow \text{RND} [1 \text{ to } L-1]$
3. Initially Set Sum (i) = 0
4. For $i \leftarrow 1$ to Ndv do
5. Set Sum (i) $\leftarrow$ Sum (i) + R (i)
6. If Sum < C then
7. Set Mask (i) $\leftarrow$ -1
8. Else
9. Set $j \leftarrow i$
10. If $j \neq \text{Ndv}$ then
11. For $k \leftarrow j + 1$ to Ndv
12. Set Mask (k) $\leftarrow$ 0
13. Else
14. Set $i \leftarrow \text{Ndv}$
15. End for

16. Set Mask (j) $\leftarrow 2^{R(j)} - 2^{Sum(i)-C}$

17. End

The genetic heritage from parent-to-child strings are maintained through high/low bit strings. The value -1 (indicates the whole binary number under 2's complement codification) filled in the first section of CM, whenever Step-6 is true. The 2nd section is filled with 0's and it indicates genetic material is coming from the P2 string (Step-11 and Step-12) (when Step-6 is false). The third section represents the P1-to-P2 transition and to fill this part equation (1) is used (Step-16).

$$CM_{tr}^C = 2^{R_{tr}} - 2^{Sum(tr)-C} \text{ with } tr \in [1 \div Ndv] \quad (1)$$

Equation (1) is an algebraic expression and it requires 2's power calculation. Equation (2) shows an equivalent relationship of equation (1) and it includes binary manipulations.

$$CM_{tr}^C = [\sim (-1 \ll R_{tr} \gg Sum - C)] \ll Sum - C \quad (2)$$

Where symbol $\sim$, $\ll$ and $\gg$ denote respectively bitwise inversion, left and right shift.

5.1.2 Bit-by-bit mask-fill crossover

A new offspring is created through a random choice from a parent string. It requires a much simpler mask-fill operations than cut crossover mask-fill operator. Algorithm-6 shows the procedure of bit-bit crossover mask-fill operator. It uses a simple integer random generator at the chromosomal level (Step-3).

Algorithm-6. bit_by_bit_mask_fill (P1, P2)

Input: P1, P2, L and Ndv

```
1. Begin
2.   For i  $\leftarrow$  1 to Ndv do
3.     Set Mask (i)  $\leftarrow$  INT (RND * R (i)) - 1
4.   End for
5. End
```

Equation (3) shows a specific range of each design variable.

$$[0 \div R_i - 1] \text{ where } i = 1, Ndv \quad (3)$$

It is interesting to note the related binary image or a randomly filled CM array is a collection of random high (-1: minus one) or low (0: zero) value.

5.1.3 Local cut crossover mask-fill operator

In case of local cut crossover mask-fill operator a single cut is applied to each design variable related binary substring. Algorithm-7 presents the local-cut crossover mask-fill operator. Step-3 performs a local cut randomly on each design variable.

Algorithm-7. local_cut_mask_fill (P1, P2)

Input: P1, P2, L and Ndv

1. Begin
  2. For $i \leftarrow 1$ to Ndv do
  3. Set $C \leftarrow \text{RND} [1 \div R(i) - 1]$
  4. Set $\text{Mask}(i) \leftarrow 2^{R(i)} - 2^C$
  5. End for
  6. End
-

Equation (4) is combined form of step-3 and 4 that shows the fill scheme used in Algorithm-7.

$$CM_i^{local} = 2^{R_i} - 2^{RND[1 \div R_i - 1]} \quad \text{where } i = 1, Ndv \quad (4)$$

5.2 Mutation operators adapted

Mutation operators are primarily used as a mechanism to maintain diversity [38]. Five different mutations (including no mutation) scheme are discussed in this section.

5.2.1 No mutation

No mutation operator is used (mutation probability is set to 0). Deb and Deb [56] considered the case of no mutation to analyze the effect of various mutation schemes on a real-parameter GA.

5.2.2 Block copier with random length mutation

Figure 3 demonstrates the working of a block copier with random length mutation (BCRLM). Here, the term random length indicates the block size value 'j' is selected randomly.

Figure 3. An example of a BCRLM method.

Algorithm-8 presents BRCLM. Each child string is created through a random block shifting up to a string length $(L)/2$ of genes from middle chromosome.

Algorithm-8. *Block Copier with Random Length Mutation*

Input: P1, P2 and L

1. Begin
  2. Set P1 ← Select a parent P1
  3. Set $i \leftarrow \text{RND}(L/2)$
  4. Set $j \leftarrow \text{RND}(VLG)$
  5. Set OS ← Copy elements from P1 upto ' i '.
  6. Set OS ← Copy elements from P1 upto ' j '.
  7. Update OS ← genes from $i + j$ upto CS
  8. Display OS.
  9. End
-

An illustrative example is depicted in Figure 1. It has a string length $L = 10$, a block cut $i = 4$ and block size $j = 3$. The value ' j ' is chosen randomly and therefore it is called as a block copier with random length mutation.

5.2.3 Mutation mask-fill

Mutation mask-fill operator is similar to a classical random binary inversion, but it performs mutation only on a specific mutation rate. Equation (5) is used to perform mutation mask-fill operator [8].

$$MM_i = \sum_{j=0}^{R_i-1} 2^j \cdot \delta_{lm} \quad i = 1, Nd \forall \quad (5)$$

$$\delta_{lm} = \begin{cases} l = m & \text{if } rand < MR \\ l \neq m & \text{otherwise} \end{cases}$$

Where, δ_{lm} is the Kronecker operator.

Equation (5) can be interpreted as “if the random value 'rand' is less than 'MR' then apply binary inversion otherwise no need to apply mutation mask-fill operation”. The mutation mask-fill operator is computationally efficient as it is only applied if exploration of the search space is required.

5.2.4 Block copier fixed length mutation

Block copier fixed length mutation (BCFLM) is different from the BCRLM because of the block size, which is fixed, i.e. the value of ' j ' does not vary throughout the mutation operation.

Figure 4. An example of a BCFLM method.

Each offspring is created by shifting a fixed length block ' j ' of genes from the middle of the chromosome to the place starting from a random location ' i '. Algorithm-9 presents BCFLM.

Algorithm-9. Block Copier with Fixed Length Mutation

Input: P1, P2 and L

1. Begin
  2. Set $P1 \leftarrow$ Select a parent P1
  3. Set $i \leftarrow \text{RND}(L/2)$
  4. Set $j \leftarrow \text{RND}(\text{FLG})$
  5. Set OS $\leftarrow$ Copy elements from P1 upto ' i '.
  6. Set OS $\leftarrow$ Copy elements from P1 upto ' j '.
  7. Update OS $\leftarrow$ genes from $i + j$ upto CS.
  8. Display OS.
  9. End
-

Figure 4 demonstrates Algorithm-9. We can see that a string length $L = 10$, random location $i = 4$ and block cut $j = 2$ is chosen.

5.2.5 Inverse mutation

Each gene in a child population is created by inverting the corresponding gene from the parent string. We can see from Figure 5 each gene present in the parent string is inverted in child string, i.e. 1 for 0 and 0 for 1.

Figure 5. Inverse mutation scheme

6. Computational Experiments

The experimental studies investigated the effect of each reproduction operator combinations. The best reproduction operator combination is used for experimentation and results are compared with the state-of-the-art algorithms. Java programming on Net Beans IDE 7.0.1, Intel Core™ 2 processor (2.8 GHz) with 2 GB RAM are used for the experiments. In this section, we explain our test problem, control-parameter settings, experimental results, discussion and statistical analysis.

6.1 Test Problem

The set of training data of varying patterns of 0's and 1's is selected as given in Table 1.

Table 1. Test Language description.

L-id	Descriptions	Standard set
L1	Balanced parentheses problem	Huijsen [49], Keller & Lutz set [50]
L2	(0^*1) over $(0+1)^*$	Dupont set [51]
L3	$(10)^*$ over $(0+1)^*$	Tomita [52]/Dupont set [51]
L4	Odd binary number ending with one	Dupont set [51]

6.2 Genetic Algorithm Implemented

We implemented the GA described in Section 4. The implementation of Algorithm-3 was deliberately simple so that a clear and concise comparison can be made.

Table 2. Configuration Of The Genetic Algorithm

S.N.	Parameter	Type/Value
1.	Variable representation	Bit string
2.	Genes	1 or 0
3.	Population size (POP_SIZE)	120
4.	Chromosome size (CS)	120
5.	Selection	Roulette wheel
6.	Crossover rate (CR)	0.9
7.	Mutation rate (MR)	0.8
8.	Performance measure	Fitness function value

Table 2 presents the control-parameters utilized for the implementation. A rigorous control-parameter tuning was conducted using orthogonal array and Taguchi method. The *POP_SIZE* and *CS* were not varied, but kept at the values presented in Table 2 and only crossover and mutation operator combinations were investigated in the present research. The same approach might be straight forwardly applied to many other reproduction operators suggested in literature.

6.3 Results and Discussion

For all the 15 reproduction operator combinations, 50 runs were performed and results were averaged. The parameter value shown in Table 2 were used because algorithm produces the best

result (correct grammar in a number of generations with minimum number of production rules that is capable of representing the language features). Table 3 shows the results obtained after implementing the algorithm on the different language sets.

Table 3. Context Free Grammar (CFG) with fitness value

L-id	Fitness	Grammar $\langle V, \Sigma, P, S \rangle$
L1	1014	$\langle \{S, M\}, \{(\cdot)\}, \{S \rightarrow ?, M \rightarrow (S)S\}, S \rangle$
L2	1014	$\langle \{S\}, \{0, 1\}, \{S \rightarrow 1, S \rightarrow 0S\}, S \rangle$
L3	1014	$\langle \{S\}, \{0, 1\}, \{S \rightarrow ?, S \rightarrow 10S\}, S \rangle$
L4	1012	$\langle \{S, M\}, \{0, 1\}, \{S \rightarrow 1M/0SM, M \rightarrow SM/?\}, S \rangle$

Figure (6) (a) – (d) shows the comparison chart (fitness value vs. generations) of the 30 iterations/generations for each C_i-M_j respectively for L1 through L4. It can be seen that reproduction operator combination C1-M2 showed the best performance than the other reproduction operator combinations. Result reveals that C1-M2 has converged to the global optimum quickly. It concludes that C1-M2 can explore and exploit a search space faster than the other C_i-M_j combinations.

(a)

(b)

(c)

Figure 6. Comparison chart for crossover and mutation (Ci-Mj) operator combinations. (a) - (d) shows the fitness value vs. generation chart, respectively, for L1, L2, L3 and L4.

Table 4. CFG matrix of Ci-Mj for languages L1 through L4

Ci-Mj	Generations															
	Gen. Range				Threshold				Mean(μ)				Std. Dev (σ)			
	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4
C1-M0	35±10	4±2	8±6	18±7	48	8	15	27	17.3	5	7	17.4	13.38	2.5	3.9	6.15
C2-M0	73±22	17±8	14±6	74±37	97	28	21	110	32.7	9.7	7.7	48.5	29.67	8.09	6.0	26.12
C3-M0	35±30	25±6	10±8	76±54	72	34	29	138	31.5	11.8	8.8	52.9	20.25	9.5	7.45	37.4
C1-M1	41±17	7±5	6±3	27±17	58	16	10	49	19.9	7.1	5.1	20.9	17.43	4.8	2.11	11.79
C1-M2	10±7	7±5	4±2	13±11	18	13	6	30	9.9	5.2	3.4	12.2	3.27	3.04	1.8	8.31
C1-M3	14±13	6±3	7±2	13±7	27	10	10	21	8.7	6.6	5.5	11.9	6.85	2.05	2.61	4.01
C1-M4	11±5	5±2	11±6	26±9	06	8	19	36	9.9	4.7	6.6	14.4	3.27	2.05	5.2	9.92
C2-M1	54±34	15±9	17±12	60±38	61	35	30	106	34.5	12.7	10.9	42.2	38.25	10.2	10.52	33.02
C2-M2	27±6	35±12	6±3	24±9	33	51	9	32	17.2	9.8	4.4	13.8	7.6	14.26	2.49	9.91
C2-M3	33±18	6±3	6±4	12±8	51	11	13	22	17.2	4.5	5.1	11.3	13.78	3.04	2.98	4.36
C2-M4	47±35	14±6	8±7	18±7	100	24	17	27	43.4	9.8	7.9	15	26.45	6.85	4.98	7
C3-M1	32±11	9±6	7±5	21±9	43	17	13	34	26.6	9.2	5.5	21.1	10.38	4.9	4.15	9.37
C3-M2	9±5	4±3	5±3	11±5	16	8	8	18	7.3	5.6	4.3	9.9	5.49	2.5	2.32	4.9
C3-M3	13±8	6±4	8±4	9±7	24	12	14	16	10.6	5	5.9	9.2	5.27	3.3	3.33	4.28
C3-M4	37±14	7±5	9±6	11±5	52	14	16	18	28.8	6.9	6.4	8.7	14.7	2.9	4.60	4.05

Result statistics are presented in Table 4. It shows the comparison based on various factors: generation range (Gen. Range), threshold value (highest ranking solution of fitness value), mean (Mean(μ)) and standard deviation (Std. Dev (σ)). This result demonstrate that C1-M2 showed better performance than the other Ci-Mj.

6.3.1 Statistical Testing

A statistical test (F-test) is conducted upon the collected samples. Equation (6) shows the hypothesis (95% level of significance) that is made to conduct the statistical test.

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = \mu_8 = \mu_9 = \mu_{10} = \mu_{11} = \mu_{12} = \mu_{13} = \mu_{14} = \mu_{15} \quad (6)$$

$$H_A : \text{At least one mean is different than others.}$$

We selected sample data of L4 for statistical tests. 30 samples are drawn from each Ci-Mj combinations. Table 5 shows the descriptive analysis for each Ci-Mj. It can be seen that C1-M2 has achieved the highest mean average fitness value 957.67.

Table 5. Descriptive analysis for each Ci-Mj combination for L4

CiMj	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
C1M0	30	810.72	119.73	21.86	766.01	855.43	675.30	1002.30
C2M0	30	766.55	100.48	18.34	729.03	804.07	626.90	923.60
C3M0	30	755.83	77.49	14.14	726.89	784.77	621.30	875.80
C1M1	30	799.74	144.76	26.43	745.68	853.79	596.90	996.00
C1M2	30	957.67	95.29	17.39	922.08	993.25	719.90	1013.90
C1M3	30	916.11	105.04	19.17	876.89	955.33	597.90	1002.50
C1M4	30	907.42	106.54	19.45	867.63	947.20	619.30	1001.50
C2M1	30	830.88	88.71	16.19	797.75	864.00	646.60	938.60
C2M2	30	937.77	95.60	17.45	902.06	973.47	617.20	1004.80
C2M3	30	938.99	111.44	20.34	897.38	980.61	598.20	1002.60
C2M4	30	859.45	121.47	22.17	814.09	904.81	607.90	1001.20
C3M1	30	840.07	110.82	20.23	798.69	881.46	589.20	962.70
C3M2	30	933.25	100.90	18.42	895.57	970.93	624.00	1002.30
C3M3	30	906.71	99.913	18.24	869.40	944.02	610.70	995.40
C3M4	30	935.99	102.842	18.77	897.58	974.39	611.50	1003.40
Total	450	873.14	123.76	5.83	861.68	884.61	589.20	1013.90

Table 6. Result of one way ANOVA test

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1945283.99	14	138948.85	12.25	0.00
Within Groups	4932168.23	435	11338.31		
Total	6877452.22	449			

Table 6 presents the results of the ANOVA test where the p-value (Sig.) is less than 0.00 < 0.05. Hence, we could reject the null hypothesis. However, this result does not indicate which Ci-Mj combination is responsible for this difference. Therefore, a paired t-test is conducted to compare two sample means. Total 14 pairs have been created to conduct paired t-test. Until now, C1-M2 showed the best results, therefore pair is made as C1-M2 with the other Ci-Mj. Table 7 - 10 shows the paired t-test results of Ci-Mj, respectively, for L1, L2, L3 and L4. These results revealed that C1-M2 showed significantly better performance than the other Ci-Mj. A comprehensive result statistics is given in the *Appendix A* (Table A.1 – A.13 is given for each language with respect to Ci-Mj combinations).

514

Table 7. Paired t-Test results of Ci-Mj combination for L1

Ci-Mj Pairs		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence				
					Interval of the Difference				
					Lower	Upper			
Pair 1	C1M0 - C1M2	-79.63	79.70	14.55	-109.39	-49.87	-5.47	29	.000
Pair 2	C2M0 - C1M2	-150.10	78.90	14.40	-179.56	-120.63	-10.42	29	.000
Pair 3	C3M0 - C1M2	-221.64	100.17	18.28	-259.04	-184.23	-12.11	29	.000
Pair 4	C1M1 - C1M2	-77.52	31.85	5.81	-89.41	-65.623	-13.32	29	.000
Pair 5	C1M3 - C1M2	-10.60	17.23	3.14	-17.03	-4.16	-3.36	29	.002
Pair 6	C1M4 - C1M2	-55.13	83.31	15.21	-86.23	-24.02	-3.62	29	.001
Pair 7	C2M1 - C1M2	-123.80	65.69	11.99	-148.33	-99.27	-10.32	29	.000
Pair 8	C2M2 - C1M2	-74.46	66.81	12.19	-99.41	-49.51	-6.10	29	.000
Pair 9	C2M3 - C1M2	-60.54	51.88	9.47	-79.91	-41.17	-6.39	29	.000
Pair 10	C2M4 - C1M2	-122.81	60.32	11.01	-145.34	-100.29	-11.15	29	.000
Pair 11	C3M1 - C1M2	-151.86	73.36	13.39	-179.25	-124.46	-11.33	29	.000
Pair 12	C3M2 - C1M2	-22.37	30.99	5.65	-33.94	-10.79	-3.95	29	.000
Pair 13	C3M3 - C1M2	-42.98	58.64	10.70	-64.88	-21.08	-4.01	29	.000
Pair 14	C3M4 - C1M2	-179.42	73.00	13.32	-206.68	-152.16	-13.46	29	.000

515

Table 8. Paired t-Test result of Ci-Mj combination for L2

Ci-Mj Pairs		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence				
					Interval of the Difference				
					Lower	Upper			
Pair 1	C1M0 - C1M2	-91.98	59.09	10.79	-114.04	-69.91	-8.52	29	.00
Pair 2	C2M0 - C1M2	-116.91	83.71	15.28	-148.17	-85.65	-7.64	29	.00
Pair 3	C3M0 - C1M2	-131.36	81.19	14.82	-161.68	-101.03	-8.86	29	.00
Pair 4	C1M1 - C1M2	-89.59	64.23	11.72	-113.58	-65.61	-7.64	29	.00
Pair 5	C1M3 - C1M2	-151.67	110.35	20.14	-192.88	-110.46	-7.52	29	.00
Pair 6	C1M4 - C1M2	-171.85	104.64	19.10	-210.93	-132.78	-8.99	29	.00
Pair 7	C2M1 - C1M2	-58.05	47.61	8.69	-75.83	-40.27	-6.67	29	.00
Pair 8	C2M2 - C1M2	-127.24	81.35	14.85	-157.62	-96.86	-8.56	29	.00
Pair 9	C2M3 - C1M2	-91.61	79.95	14.59	-121.47	-61.75	-6.27	29	.00
Pair 10	C2M4 - C1M2	-100.35	97.10	17.72	-136.61	-64.09	-5.66	29	.00
Pair 11	C3M1 - C1M2	-67.52	66.68	12.17	-92.42	-42.61	-5.54	29	.00
Pair 12	C3M2 - C1M2	-29.38	27.50	5.02	-39.65	-19.11	-5.85	29	.00
Pair 13	C3M3 - C1M2	-7.09	9.048	1.65	-10.47	-3.71	-4.29	29	.00
Pair 14	C3M4 - C1M2	-43.46	57.76	10.54	-65.03	-21.89	-4.12	29	.00

516

Table 9. Paired t-Test result of Ci-Mj combination for L3

Ci-Mj Pairs		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence				
					Interval of the Difference				
					Lower	Upper			
Pair 1	C1M0 - C1M2	-161.10	60.68	11.08	-183.76	-138.44	-14.54	29	.00
Pair 2	C2M0 - C1M2	-145.91	66.44	12.13	-170.72	-121.10	-12.02	29	.00
Pair 3	C3M0 - C1M2	-194.07	78.53	14.33	-223.39	-164.74	-13.53	29	.00
Pair 4	C1M1 - C1M2	-217.43	84.36	15.40	-248.93	-185.93	-14.11	29	.00
Pair 5	C1M3 - C1M2	-157.92	80.77	14.74	-188.08	-127.75	-10.70	29	.00
Pair 6	C1M4 - C1M2	-130.03	56.22	10.26	-151.02	-109.04	-12.66	29	.00
Pair 7	C2M1 - C1M2	-143.64	82.33	15.03	-174.38	-112.89	-9.55	29	.00
Pair 8	C2M2 - C1M2	-200.59	95.56	17.44	-236.27	-164.90	-11.49	29	.00
Pair 9	C2M3 - C1M2	-128.04	54.20	9.89	-148.28	-107.80	-12.93	29	.00
Pair 10	C2M4 - C1M2	-177.41	94.85	17.31	-212.83	-141.99	-10.24	29	.00
Pair 11	C3M1 - C1M2	-124.45	88.70	16.19	-157.57	-91.33	-7.68	29	.00
Pair 12	C3M2 - C1M2	-151.95	91.51	16.70	-186.12	-117.78	-9.09	29	.00
Pair 13	C3M3 - C1M2	-183.60	91.05	16.62	-217.60	-149.60	-11.04	29	.00
Pair 14	C3M4 - C1M2	-32.78	33.23	6.06	-45.19	-20.37	-5.40	29	.00

517

518

519

520

521

522

523

524

Table 10. Paired t-Test result of Ci-Mj combination for L4

Ci-Mj Pairs		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	C1M0 - C1M2	-146.73	93.19	17.01	-181.53	-111.93	-8.62	29	.00
Pair 2	C2M0 - C1M2	-190.90	71.93	13.13	-217.76	-164.04	-14.53	29	.00
Pair 3	C3M0 - C1M2	-201.62	72.55	13.24	-228.71	-174.53	-15.22	29	.00
Pair 4	C1M1 - C1M2	-157.71	96.99	17.70	-193.93	-121.49	-8.90	29	.00
Pair 5	C1M3 - C1M2	-41.34	25.18	4.59	-50.74	-31.94	-8.99	29	.00
Pair 6	C1M4 - C1M2	-50.03	32.42	5.91	-62.14	-37.93	-8.45	29	.00
Pair 7	C2M1 - C1M2	-126.58	54.73	9.99	-147.01	-106.14	-12.66	29	.00
Pair 8	C2M2 - C1M2	-19.69	23.10	4.21	-28.31	-11.06	-4.66	29	.00
Pair 9	C2M3 - C1M2	-18.46	27.21	4.96	-28.62	-8.30	-3.71	29	.001
Pair 10	C2M4 - C1M2	-98.00	68.50	12.50	-123.58	-72.42	-7.83	29	.00
Pair 11	C3M1 - C1M2	-117.38	49.72	9.07	-135.95	-98.81	-12.92	29	.00
Pair 12	C3M2 - C1M2	-24.20	26.82	4.89	-34.22	-14.19	-4.94	29	.00
Pair 13	C3M3 - C1M2	-50.74	39.58	7.22	-65.52	-35.96	-7.02	29	.00
Pair 14	C3M4 - C1M2	-21.470	23.03	4.20	-30.07	-12.86	-5.10	29	.00

(a)

(b)

(c)

Figure 7.(a) Profile plot for estimated marginal means of fitness value for each Ci-Mj for L4. (b) Box plot of average fitness function values achieved on L4 by different Ci-Mj. (c) Error bar plot of mean average fitness function values Vs. Ci-Mj.

Table 11. Tukey's Hinges percentiles distribution of fitness function value for each Ci-Mj

CiMj	Percentiles						
	5	10	25	50	75	90	95
C1M0			680.30	860.80	920.70		
C2M0			639.80	783.60	831.00		
C3M0			685.20	778.70	848.10		
C1M1			610.20	830.20	880.70		
C1M2			938.50	1007.25	1010.50		
C1M3			905.30	942.70	970.40		
C1M4			827.30	963.55	996.10		
C2M1			764.70	825.90	918.50		
C2M2			924.30	982.30	996.90		
C2M3			946.70	996.40	999.90		
C2M4			779.20	866.40	994.50		
C3M1			748.80	882.25	934.00		
C3M2			936.60	989.80	989.80		
C3M3			882.50	940.70	990.80		
C3M4			940.40	988.95	995.50		

Figure 7 (a) presents an estimated marginal mean plot (Appendix A is attached where Figure A.1, A.2 and A.3 respectively presents estimated marginal mean plot, box plot and error bar pl. It can be seen that C1-M2 has achieved the highest mean average fitness value, whilst C3-M0 shows the lowest mean average fitness function value. Figure 7 (b) shows the boxplot of fitness value achieved by each Ci-Mj on the L4. The boxplot revealed that C1-M2 showed the best (C2-M2, C2-M3, C3-M2 and C3-M4 showed similar), whilst both C2-M0 and C3-M0 demonstrated the worst result. Table 11 presents *Tukey's Hinges Percentile* distribution of fitness function value for each Ci-Mj. It shows 1st (25% percentile), 2nd (50% percentile/median) and 3rd (75% percentile) quartile. Figure 7 (b) and Table 11 revealed that C1-M2 has achieved the best median value (= 1007.25) with an interquartile range 72. Figure 7 (c) presents the error bar plot, which revealed that C1-M2 preformed significantly better than C1-M0, C2-M0, C3-M0, C1-M1, C2-M1, C2-M4 and C3-M1. There is no significant difference in the performance of C1-M3, C1-M4, C2-M2, C2-M3, C3-M2, C3-M3 and C3-M4. Overall, C1-M2 outperformed all the Ci-Mj combinations.

Aforementioned discussion reveals that the performance of C1-M2 is significantly better than the many other Ci-Mj combinations. This is due to the following reasons: C1 defines 3 substrings: first substring identify bit groups coming from P1 only. Second substring identify bit groups coming from P2 only. Third substring codifies P1-to-P2 bit transition, which reduces extra operation for bit manipulation. Also, equation (1) works effectively if the cut point does not lie inside a Ndv of a binary image substring. In such case, bit transition occurs at chromosomal level and the 2nd term of the right hand side is set to 1 (Sum (tr) = C), hence $CM_{tr}^C = 2^{R_r}$ (maximum value allowed by related bit transition, equivalent to value -1. These ways C1exploit the search space effectively. Previous literatures [53] [54] [55] showed that mutation is an exploratory operator and it has less impact near the beginning and more near the end of a run. M2 used in

experiments is a classical random binary inversion and it works on a specific mutation rate. For M2, if random value ‘*rand*’ (ranging on [0, 1]) is less than MR then the mutation operation will be performed by the cumulative storage of the 2’s power through equation (5). In this way, mutation operation will not be performed in each iteration.

6.3.2 Performance comparison with state-of-the-art algorithms

The performance of the proposed algorithm has been compared with three state-of-the-art algorithms, namely, classical genetic algorithm (CGA) [18], improved tabular representation (ITBL) [19] and genetic algorithm with minimum description length principle (GAWMDL) [12]. ITBL is one of the popular GI algorithm, whilst GAWMDL was proposed for both GI and to avoid premature convergence. As discussed, GI is a NP hard problem and too many temporary grammar rules get generated during evolutionary process. Handling these temporary rules are difficult. The proposed GAWS algorithm utilizes: (a) mask-fill reproduction operators and Boolean based procedure mixture with Boolean operators for offspring’s generation. These operations introduce diversity in the population and avoid premature convergence. (b) Sequitur algorithm has been implemented to handle temporary grammar rules. These two features make the proposed GAWS algorithm more effective as compare to any other algorithms. Fitness value vs generation chart for language L4 is presented in Figure 8. It can be seen that the proposed GAWS has outperformed other algorithms.

Figure 8. Fitness value vs. generation chart with respect to each algorithm for L4.

6.3.2.1 Statistical testing

Statistical tests have been conducted to determine the performance significance of the proposed GAWS algorithm. F-test has been performed at 95% level of significance considering the hypothesis given in equation (7).

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

$$H_A : \text{At least one mean is different than others.} \quad (7)$$

Table 12. Descriptive analysis for each algorithm for L4

Algorithm	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
CGA	30	695.02	48.04	8.77	677.08	712.96	553.70	742.30
ITBL	30	831.42	148.11	27.04	776.11	886.73	566.30	1006.00
GAWS	30	957.67	95.29	17.39	922.08	993.25	719.90	1013.90
GAWMDL	30	854.65	137.82	25.16	803.19	906.12	560.80	1002.50
Total	120	834.69	146.91	13.41	808.13	861.24	553.70	1013.90

Table 13. Result of one way ANOVA test

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1051205.911	3	350401.970	26.788	.000
Within Groups	1517343.104	116	13080.544		
Total	2568549.015	119			

Table 14. Multiple comparison t-test (Posthoc test)

(I) Algorithm	(J) Algorithm	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
CGA	ITBL	-136.40*	29.53	.00	-213.37	-59.42
	GAWS	-262.64*	29.53	.00	-339.62	-185.67
	GAWMDL	-159.63*	29.53	.00	-236.60	-82.65
ITBL	CGA	136.40*	29.53	.00	59.42	213.37
	GAWS	-126.24*	29.53	.00	-203.22	-49.27
	GAWMDL	-23.23	29.53	.86	-100.20	53.74
GAWS	CGA	262.64*	29.53	.00	185.67	339.62
	ITBL	126.24*	29.53	.00	49.27	203.22
	GAWMDL	103.01*	29.53	.00	26.03	179.98
GAWMDL	CGA	159.63*	29.53	.00	82.65	236.60
	ITBL	23.23	29.53	.86	-53.74	100.20
	GAWS	-103.01*	29.53	.00	-179.98	-26.03

*. The mean difference is significant at the 0.05 level.

Thirty samples were drawn from each algorithm for the L4. Table 12 presents the descriptive analysis. It presents the minimum, maximum and average fitness value. GAWS showed the maximum mean fitness function value, which is 957.67. The result of ANOVA test is depicted in Table 13. It indicates that the p-value is $0.00 < 0.05$. Hence, H_0 is rejected. It mean that one or the other samples that have been drawn might be better than the other one. Therefore, to derive a conclusion, a multiple comparison test using TukeyHSD (Tukey-Kramer Honestly Significant Difference) test has been conducted. TukeyHSD test is adapted because (a) it is less liable to Type-1 errors; and (b) it is suitable for the experiments that have an equal number of observations. Table 14 shows the result of the TukeyHSD test. It presents pairwise comparison of GAWS – CGA,

GAWS – ITBL and GAWS – GAWMDL are significantly different at 95% level of confidence interval.

(a)

(b)

(c)

Figure 9. (a) Profile plot for estimated marginal means of fitness value for each algorithm for L4. (b) Box plot of average fitness function values achieved on L4 by different algorithm. (c) Error bar plot of mean average fitness function values Vs. algorithm.

Table 15. Homogeneous subsets of algorithms using TukeyHSD test

Algorithm	N	Subset for alpha = 0.05		
		1	2	3
CGA	30	695.02		
ITBL	30		831.42	
GAWMDL	30		854.65	
GAWS	30			957.67
Sig.		1.00	.86	1.00

Means for groups in homogeneous subsets are displayed.

a. Uses Harmonic Mean Sample Size = 30.000.

Table 16. Percentiles distribution of average fitness function value using Tukey's Hinges test.

Algorithm	Percentiles					
	5	10	25	50	75	95
CGA			683.10	703.15	730.00	
ITBL			730.90	876.05	930.60	
GAWS			938.50	1007.25	1010.50	
GAWMDL			740.50	896.40	970.70	

Table 15 presents the results of homogenous subsets. Results reveals that CGA (fitness value = 695.02) falls in the group-1 with 1.00 level of significance, ITBL (fitness value = 831.42) and GAWMDL (fitness value = 854.65) fall in group-2 with 0.86 level of significance whilst the proposed GAWS ((fitness value = 957.67) fall in group-3 with 1.00 level of significance. This result indicate that GAWS showed the highest mean average fitness function value.

Figure 9 (a) graphically displays the mean of the average fitness function value of each algorithm. The X-axis represents the algorithms and the Y-axis represents the estimated marginal mean of the average fitness function value. The mean of the average fitness function value of the GAWS is the highest than the other three algorithms. Figure 9 (b) shows the boxplot for each algorithm. We can see, the proposed GAWS perform the best, ITBL and GAWMDL showed good performance, whilst CGA's performance is the worst.

The percentile distribution of the fitness value is presented in Table 16. It shows 1st (25% percentile), 2nd (50% percentile, also known as the median) and 3rd (75% percentile) quartile. Figure 9 (c) and Table 16 revealed that GAWS has achieved the best median value (= 1007.25) with interquartile range 72. GAWMDL (median = 896.40) and ITBL (median = 876.05) respectively showed the 2nd and the 3rd the best median value. CGA (median = 703.15) showed the lowest median value. Figure 9 (c) depicts error bars where interpolation line has been drawn to heighlight the difference in the performance of the algorithms. It can been seen that GAWS's performance is the best over the other three algorithms. Also, the performance of the ITBL and GAWMDL is almost same where GAWMDL showed slightly better results. All in all, the proposed GAWS algorithm outperformed the other three algorithms. From these results, it is noted that C1M2 is the most suitable reproduction operator combination to find an optimal solution. In addition, mask-fill reproduction operators successfully maintained diversity in the population during the genetic evolution which allowed the GAWS to find the global solution.

7. Conclusions and Future Research Directions

In this paper, the performance of the mask-fill crossover and mutation operators have been compared against various other reproduction operators. A comprehensive discussion on the contribution of crossover and mutation operators for the success of a GA has been presented. It also present a GA referred to as BMOGA that implements mask-fill reproduction operators. Previous studies [8] [9] [10] [11] have recognized the importance of the mask-fill reproduction

operators in GA. Mask-fill reproduction operators utilizes BMODS to perform crossover and mutation operations and then a Boolean based procedure that generates an offspring. Various crossover and mutation operators have been discussed that were adapted for the comparison. The applicability of the BMOGA for GI has been shown by creating a hybridization of the BMOGA with Sequitur algorithm. Two fold experimentations have been performed. First, fifteen crossover and mutation operator combinations have been created for experimentation. From the experimental results and subsequent analysis C1-M2 ensures the best performance over any other Ci-Mj combinations for a given standard corpus set with varying characteristics. Experimental results have been analyzed based on the three quality measures: (a) average fitness value; (b) generation range; and (c) threshold value. C1-M2 showed very effective performance for both convergence and diversity of population. Statistical tests have been conducted on the collected samples of Ci-Mj combinations. The result of statistical tests demonstrate that the performance of C1-M2 was significantly better than other Ci-Mj combinations. Secondly, the performance of the proposed GAWS algorithm has been tested against the state-of-the-art algorithms. Three algorithms, namely, CGA, ITBL and GAWMDL have used for the comparison purpose. All these algorithms have been implemented using the same control parameters. Result statistics revealed that the proposed GAWS algorithm has outperformed over the other three algorithms. Authors believe that the results reported in this paper will motivate the researchers to utilize the mask-fill crossover and mutation operators to solve complex optimization problems. In addition, it is believed that pre-knowledge of crossover and mutation operator combination develops an ease during selection of a reproduction operator combination to implement a GA. But it would be interesting to implement mask-fill reproduction operators for more complex optimization problems such as natural language grammatical inference, learning grammars from domain specific languages, solving combinatorial problems and other real world optimization problems.

References

1. C. Darwin and W.F. Bynum. "The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life". pp. 441-764, 2009. AL Burt.
2. C. Darwin. "*On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*". John Murray, London, 1859.

3. H.M. Pandey, A. Chaudhary and D. Mehrotra. "*A comparative review of approaches to prevent premature convergence in GA.*" Applied Soft Computing 24 (2014): 1047-1077.
4. M. Črepinšek, S. H. Liu and M. Mernik. "*Exploration and exploitation in evolutionary algorithms: a survey.*" ACM Computing Surveys (CSUR) 45.3 (2013): 35.
5. V. Kapoor, S. Dey and A. P. Khurana. "*Empirical analysis and random respectful recombination of crossover and mutation in genetic algorithms.*" International Journal of Computer Applications. Special issue on Evolutionary Computation for Optimization. ECOT (2010): 25-30.
6. I. Rojas, J. González, H. Pomares, J. J. Merelo, P. A. Castillo, and G. Romero. "*Statistical analysis of the main parameters involved in the design of a genetic algorithm.*" IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 32.1 (2002): 31-37.
7. M. Srinivas and L. M. Patnaik. "*Adaptive probabilities of crossover and mutation in genetic algorithms.*" IEEE Transactions on Systems, Man, and Cybernetics 24.4 (1994): 656-667.
8. L. Iuspa and F. Scaramuzzino. "*A bit-masking oriented data structure for evolutionary operators implementation in genetic algorithms.*" Soft Computing 5.1 (2001): 58-68.
9. H. M. Pandey, A. Chaudhary and D. Mehrotra. "*Grammar induction using bit masking oriented genetic algorithm and comparative analysis.*" Applied Soft Computing 38 (2016): 453-468.
10. L. Iuspa, F. Scaramuzzino and P. Petrenga. "*Optimal design of an aircraft engine mount via bit-masking oriented genetic algorithms.*" Advances in Engineering Software 34.11 (2003): 707-720.
11. L. Iuspa and E. Ruocco. "*Optimum topological design of simply supported composite stiffened panels via genetic algorithms.*" Computers & structures 86.17 (2008): 1718-1737.
12. H. M. Pandey, A. Chaudhary, D. Mehrotra and G. Kendall. "*Maintaining regularity and generalization in data using the minimum description length principle and genetic algorithm: Case of grammatical inference.*" Swarm and Evolutionary Computation 31 (2016): 11-23.
13. S. H. Muggleton, D. Lin, N. Pahlavi and A. Tamaddon-Nezhad. "Meta-interpretive learning: application to grammatical inference". Machine learning, 94(1), pp. 25-49, 2014.

14. H. Ahonen, H. Mannila and E. Nikunen. "*Forming grammars for structured documents: an application of grammatical inference.*" Grammatical Inference and Applications (1994): 153-167.
15. P. P. Cruz-Alcázar and E. Vidal. "*Two grammatical inference applications in music processing.*" Applied Artificial Intelligence 22.1-2 (2008): 53-76.
16. T. Hong and K. Clark. "*Using grammatical inference to automate information extraction from the Web.*" Principles of Data Mining and Knowledge Discovery. Pp. 216-227, 2001.
17. S. C. Zhu and D. Mumford. "*A stochastic grammar of images*". Foundations and Trends® in Computer Graphics and Vision, 2(4), pp. 259-362, 2007.
18. J. H. Holland. "*Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*". MIT press, 1992.
19. M. Jaworski and O. Unold. "Improved TBL algorithm for learning context-free grammar". In Proceedings of the International Multiconference on ISSN (Vol. 1896, p. 7094, 2007.
20. K.A. De Jong, "*An analysis of the behavior of a class of genetic adaptive systems,*" PhD thesis, University of Michigan, 1975.
21. W. M. Spears and K. D. De Jong. "*On the virtues of parameterized uniform crossover*". Naval Research Lab Washington DC, 1995.
22. L. J. Fogel, A. J. Owens and M. J. Walsh, "*Artificial Intelligence through Simulated Evolution*". Wiley Publishing, New York 1966.
23. I. Rechenberg. "*Evolution Strategy: Optimization of Technical systems by means of biological evolution.*" Fromman-Holzboog, Stuttgart 104 (1973).
24. T. Back, F. Hoffmeister and H. P. Schwefel. "*A survey of evolution strategies*". Proceedings of the Fourth International Conference on Genetic Algorithms. Vol 2. No. 9. Morgan Kaufmann Publishers, San Mateo, 1991.
25. H. P. Schwefel, "*Numerical Optimization of Computer Models*". New York: John Wiley & Sons, 1981.
26. D. B. Fogel and J. W. Atmar. "*Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems.*" Biological Cybernetics 63.2 (1990): 111-114.
27. W. M. Spears. "Crossover or mutation." Foundations of genetic algorithms 2 (1993): 221-237.

28. S. Picek, M. Golub and D. Jakobovic. "Evaluation of crossover operator performance in genetic algorithms with binary representation." *Bio-Inspired Computing and Applications*. Springer Berlin Heidelberg, 2012. pp. 223-230.
29. N. F. McPhee and N. J. Hopper. "Analysis of genetic diversity through population history". In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, 1999, (pp. 1112-1120). Morgan Kaufmann Publishers Inc.
30. I. Paenke, Y. Jin and J. Branke. "Balancing population-and individual-level adaptation in changing environments." *Adaptive Behavior* 17.2 (2009): 153-174.
31. E. K. Burke, S. Gustafson and G. Kendall. "Diversity in genetic programming: An analysis of measures and correlation with fitness." *IEEE Transactions on Evolutionary Computation* 8.1 (2004): 47-62.
32. A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke and J. Schmidhuber. "*A novel connectionist system for unconstrained handwriting recognition*." *IEEE transactions on pattern analysis and machine intelligence* 31.5 (2009): 855-868.
33. P. A. Bosman and D. Thierens. "*The balance between proximity and diversity in multiobjective evolutionary algorithms*." *IEEE transactions on evolutionary computation* 7.2 (2003): 174-188.
34. S. W. Mahfoud. "*Niching methods for genetic algorithms*." *Urbana* 51.95001 (1995): 62-94.
35. A. E. Eiben and C. A. Schippers. "*On evolutionary exploration and exploitation*." *Fundamenta Informaticae* 35.1-4 (1998): 35-50.
36. Y. Y. Wong, K. H. Lee, K. S. Leung and C. W. Ho. "*A novel approach in parameter adaptation and diversity maintenance for genetic algorithms*." *Soft Computing* 7.8 (2003): 506-515.
37. T. Bäck. "*Selective pressure in evolutionary algorithms: A characterization of selection mechanisms*." *Evolutionary Computation*, 1994. IEEE World Congress on Computational Intelligence. Proceedings of the First IEEE Conference on. IEEE, 1994.
38. D.E. Goldberg and K. Deb. "*A comparative analysis of selection schemes used in genetic algorithms*." *Foundations of genetic algorithms* 1 (1991): 69-93.
39. L. E. Galván, J. McDermott, M. O'Neill and A. Brabazon. "*Towards an understanding of locality in genetic programming*." *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010.

40. K. A. De Jong and W. M. Spears. "A formal analysis of the role of multi-point crossover in genetic algorithms." *Annals of mathematics and Artificial intelligence* 5.1 (1992): 1-26.
41. J. E. Smith and T. C. Fogarty. "Operator and parameter adaptation in genetic algorithms." *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 1.2 (1997): 81-87.
42. G. R. Harik and F. G. Lobo. "A parameter-less genetic algorithm". In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1* (pp. 258-265). Morgan Kaufmann Publishers Inc., 1999.
43. W. M. Spears. "Adapting Crossover in Evolutionary Algorithms". *Evolutionary programming*, pp. 367 – 384, 1995.
44. R. E. Smith and E. Smuda. "Adaptively resizing populations: Algorithm, analysis, and first results." *Complex Systems* 9.1 (1995): 47-72.
45. A. Moraglio, Y. H. Kim, Y. Yoon and B. R. Moon. "Geometric crossovers for multiway graph partitioning." *Evolutionary Computation* 15.4 (2007): 445-474.
46. H. Ishibuchi, N. Tsukamoto and Y. Nojima. "Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, 14(6) (2010), 985-998.
47. C. G. Nevill-Manning and I. H. Witten. "Identifying hierarchical structure in sequences: A linear-time algorithm." *Journal of Artificial Intelligence Research* 7 (1997): 67-82.
48. D. Hrnčić, M. Mernik and B. R. Bryant. "Improving grammar inference by a memetic algorithm." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.5 (2012): 692-703.
49. W. O. Huijsen. "Genetic grammatical inference." *CLIN IV. Papers from the Fourth CLIN Meeting, Vakgroep Alfa-informatica*. pp. 59 – 72, 1993.
50. B. Keller and R. Lutz. "Evolving stochastic context-free grammars from examples using a minimum description length principle." *Workshop on Automata Induction Grammatical Inference and Language Acquisition*. 1997.
51. P. Dupont. "Regular grammatical inference from positive and negative samples by genetic search: the GIG method." *Grammatical Inference and Applications*. Springer Berlin Heidelberg, 1994. 236-245.

52. Tomita. "Dynamic construction of finite-state automata from examples using hill-climbing."
Proceedings of the fourth annual cognitive science conference. 1982.
53. A. Czarn, C. MacNish, K. Vijayan, B. Turlach and R. Gupta. "*Statistical exploratory analysis of genetic algorithms*." Evolutionary Computation, IEEE Transactions on 8.4 (2004): 405-421.
54. T. P. Hong, H. S. Wang, W. Y. Lin and W. Y. Lee. "*Evolution of appropriate crossover and mutation operators in a genetic process*." Applied Intelligence 16.1 (2002): 7-17.
55. J. Hesser and R. Männer. "*Towards an optimal mutation probability for genetic algorithms*,"
Proceedings of the First Workshop PPSN-I, pp. 23-32, 1990.
56. K. Deb and D. Deb. "*Analysing mutation schemes for real-parameter genetic algorithms*." International Journal of Artificial Intelligence and Soft Computing 4.1 (2014): 1-28.
57. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms. Hillsdale, NJ: Lawrence Erlbaum, 1987.
58. D. H. Younger. Recognition and parsing of context-free languages in time n^3 . Information and control, 10(2), 189-208, 1967.
59. E. M. Gold. Language identification in the limit. Information and control, 10(5), 447-474., 1976.
60. A. L. Oliveira. Grammatical Inference: Algorithms and Applications: 5th International Colloquium, ICGI 2000, Lisbon, Portugal, September 11-13, 2000 Proceedings. Springer, 2004.
61. A.V. Aho, R. Sethi and J. D. Ullman. Compilers, Principles, Techniques and Tools. Addison Wesley, 1986.
62. A. Stevenson and J. R. Cordy. A survey of grammatical inference in software engineering. Science of Computer Programming, 96, 444-459, 2014
63. L. G. Valiant. A theory of the learnable. Communications of the ACM, 27(11), 1134-1142, 1984.
64. M. Li and P. M. B. Vitányi. Learning simple concepts under simple distributions. SIAM Journal on Computing, 20(5), 911-935., 1991.

65. M. Delgado and M. C. Pegalajar. A multiobjective genetic algorithm for obtaining the optimal size of a recurrent neural network for grammatical inference. *Pattern Recognition* 38.9 (2005): 1444-1456.
66. P. Wyard. Representational issues for context free grammar induction using genetic algorithms. In *Grammatical Inference and Applications* (pp. 222-235). Springer Berlin Heidelberg, 1994.
67. F. Javed, B. R. Bryant, M. Črepinšek, M. Mernik, and A. Sprague. "Context-free grammar induction using genetic programming." In *Proceedings of the 42nd annual Southeast regional conference*, pp. 404-405. ACM, 2004.
68. N. S. Choubey and M. U. Kharat. Sequential structuring element for CFG induction using genetic algorithm. *International Journal of Futuristic Computer Application*, 2010
69. H. M. Pandey. Context free grammar induction library using Genetic Algorithms. In *Computer and Communication Technology (ICCCT)*, 2010 International Conference on (pp. 752-758). IEEE, 2010.
70. N. S. Choubey, H. M. Pandey, and M. U. Kharat. Developing genetic algorithm library using Java for CFG induction. *International Journal of Advancement in Technology* (2011).
71. H. M. Pandey, A. Dixit, and D. Mehrotra. Genetic algorithms: concepts, issues and a case study of grammar induction. In *Proceedings of the CUBE International Information Technology Conference*, pp. 263-271. ACM, 2012.
72. D. Hrnčič and M. Mernik. Memetic grammatical inference approach for DSL embedding. In *MIPRO*, 2011 Proceedings of the 34th International Convention (pp. 919-924). IEEE, 2011.
73. D. Hrnčić, M. Mernik, and B. R. Bryant. "Improving grammar inference by a memetic algorithm." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.5 (2012): 692-703.
74. Y. Sakakibara and M. Kondo. GA-based learning of context-free grammars using tabular representations. In *ICML* (Vol. 99, pp. 354-360), 1999.
75. N. Bhalse and V. Gupta. Learning CFG using Improved TBL algorithm. *Computer Science & Engineering*, 2(1), 25, 2012
76. A. Stevenson and J. R. Cordy. Grammatical inference in software engineering: an overview of the state of the art. In *Software Language Engineering* (pp. 204-223). Springer Berlin Heidelberg, 2013.

77. Vázquez et al. "Grammatical Inference of Semantic Components in Dialogues." *Computación y Sistemas* 24.2 (2020).
78. Wieczorek et al. "Grammatical Inference by Answer Set Programming." *International Conference on Computational Science*. Springer, Cham, 2020.
79. Kovačević et al. "From Grammar Inference to Semantic Inference—An Evolutionary Approach." *Mathematics* 8.5 (2020): 816.
80. Unold, Olgierd, Mateusz Gabor, and Wojciech Wieczorek. "Unsupervised Statistical Learning of Context-free Grammar." (2020).
81. Andrés et al. "Automatic Generation of Dialogues based on Grammatical Inference and the use of a Knowledge Base." *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 2, pp. 2105-2113, 2020.

Authors Biography

Hari Mohan Pandey received the B.Tech. M.Tech. Ph.D. degrees in computer science and engineering. He worked as a Postdoctoral Research Fellow with the Middlesex University, London, U.K. He also worked on a European Commission project – Dream4car under H2020. He is a Senior Lecturer with the Department of Computer Science, Edge Hill University, Lancashire, U.K. He is specialized in computer science and engineering and his research area includes artificial intelligence, soft computing, natural language process, language acquisition, machine learning and deep learning. He is author of various books in computer science and engineering, published over 70 scientific papers in reputed journals and conferences. He has delivered keynotes, invited lectures and served as session chair for International conferences. He has served as a leading guest editor for several journals Dr. Pandey is a Fellow of the Higher Education Academy (U.K. Professional Standard Framework), Fellow of BCS, Visiting Researcher for Middlesex University, London. He received the Global Award for the best computer Science faculty of the year 2015, the award for completing Indo-Us Project Gentle, award (Certificate of Exceptionalism) from the Prime Minister of India, and award for developing Innovative Teaching And Learning Models for higher-education.

Marcello Trovati is a Professor in Computer Science in the Department of Computer Science, Edge Hill University. After having obtained his PhD in Mathematics at the University of Exeter in 2007, specialising in theoretical dynamical systems with singularities, he accepted a position as algorithm tester and research specialist at a medium sized software development company. His main responsibility was to create, test and documents state-of-the-art statistical algorithms to analyse big datasets. He then moved to the newly created Dublin IBM Research Lab to carry out research mainly in the field of knowledge discovery, text mining, and mathematical modelling, where he gained valuable business and research experience through collaboration with several scientists both at IBM, and at academic institutions. He was involved in a number of research projects in collaboration with other IBM Research Centres and academic institutions. He then joined Coventry University to take up a position as Teaching Fellow, and subsequently the University of Derby as a lecturer, during which he was involved various multi-disciplinary projects focussing on mathematical modelling, algorithm design, and big data analytics. In 2016 Marcello joined the Computer Science Department at Edge Hill University as a senior lecturer and he was recently awarded a Readership in Computer Science. He is involved in several research themes and projects. He is co-leading the STEM

Data Research centre and is actively involved in the Productivity and Innovation Lab, aiming to collaborate and support SMEs in Lancashire.

82. Marcello's main interests include: Mathematical Modelling, Data Science, Big Data Analytics, Network Theory, Machine Learning, Data and Text Mining.

906

Nik Bessis received his BA from the T.E.I. Athens and his MSc and PhD degrees from De Montfort University, UK in 1995 and 2002 respectively. He is currently a full Professor and the Head of the Department of Computer Science at Edge Hill University, UK. He is a fellow of HEA, BCS and a senior member of IEEE. His research is on social graphs for network and big data analytics as well as developing data push and resource provisioning services in IoT, FI and inter-clouds for a number of settings including disaster management. He is involved in a number of funded research and commercial projects in these areas. Prof Bessis has published over 300 works and won 4 best papers awards. He is an editor of several books and an editor-in-chief of a refereed international journal.

Appendix-A

Table A.1 Descriptive analysis for L1

CiMj	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
C1M0	30	885.1300	142.43685	26.00529	831.9432	938.3168	591.10	993.40
C2M0	30	814.6633	124.30441	22.69478	768.2473	861.0794	550.00	966.00
C3M0	30	743.1233	129.22736	23.59358	694.8690	791.3776	544.60	929.20
C1M1	30	887.2433	114.76609	20.95333	844.3890	930.0977	552.90	973.70
C1M2	30	964.7633	99.57388	18.17962	927.5818	1001.9448	640.20	1012.70
C1M3	30	954.1633	106.70843	19.48220	914.3178	994.0089	600.80	1008.40
C1M4	30	909.6333	160.33138	29.27237	849.7646	969.5021	528.10	1008.10
C2M1	30	840.9600	120.12779	21.93223	796.1035	885.8165	551.90	1002.40
C2M2	30	890.2967	125.05717	22.83221	843.5996	936.9938	622.40	1006.80
C2M3	30	904.2167	126.70212	23.13254	856.9053	951.5280	562.00	997.50
C2M4	30	841.9467	120.57389	22.01368	796.9236	886.9697	554.20	941.30
C3M1	30	812.9000	109.40407	19.97436	772.0478	853.7522	618.90	958.90
C3M2	30	942.3933	114.56767	20.91710	899.6131	985.1736	546.70	1006.50
C3M3	30	921.7767	143.65650	26.22797	868.1344	975.4189	552.00	1007.60
C3M4	30	785.3400	110.69148	20.20941	744.0071	826.6729	566.00	948.00
Total	450	873.2367	137.33425	6.47400	860.5136	885.9598	528.10	1012.70

Table A.2 ANOVA test result for L1

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1761770.737	14	125840.767	8.162	.000
Within Groups	6706681.408	435	15417.658		
Total	8468452.145	449			

(a)

(b)

(c)

Figure A.1. (a) Profile plot for estimated marginal means of fitness value for each Ci-Mj for L1. (b) Box plot of average fitness function values achieved on L1 by different Ci-Mj. (c) Error bar plot of mean average fitness function values Vs. Ci-Mj.

Table A.3 Posthoc test for L1

(I) CiMj	(J) CiMj	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
C1M0	C2M0	70.46667	32.06000	.666	-38.8911	179.8244
	C3M0	142.00667*	32.06000	.001	32.6489	251.3644
	C1M1	-2.11333	32.06000	1.000	-111.4711	107.2444
	C1M2	-79.63333	32.06000	.454	-188.9911	29.7244
	C1M3	-69.03333	32.06000	.698	-178.3911	40.3244
	C1M4	-24.50333	32.06000	1.000	-133.8611	84.8544
	C2M1	44.17000	32.06000	.989	-65.1877	153.5277
	C2M2	-5.16667	32.06000	1.000	-114.5244	104.1911
	C2M3	-19.08667	32.06000	1.000	-128.4444	90.2711
	C2M4	43.18333	32.06000	.991	-66.1744	152.5411
	C3M1	72.23000	32.06000	.626	-37.1277	181.5877
	C3M2	-57.26333	32.06000	.903	-166.6211	52.0944
	C3M3	-36.64667	32.06000	.998	-146.0044	72.7111
	C3M4	99.79000	32.06000	.118	-9.5677	209.1477
	C1M0	-70.46667	32.06000	.666	-179.8244	38.8911
	C3M0	71.54000	32.06000	.642	-37.8177	180.8977
C2M0	C1M1	-72.58000	32.06000	.618	-181.9377	36.7777
	C1M2	-150.10000*	32.06000	.000	-259.4577	-40.7423
	C1M3	-139.50000*	32.06000	.002	-248.8577	-30.1423
	C1M4	-94.97000	32.06000	.173	-204.3277	14.3877
	C2M1	-26.29667	32.06000	1.000	-135.6544	83.0611
	C2M2	-75.63333	32.06000	.547	-184.9911	33.7244
	C2M3	-89.55333	32.06000	.254	-198.9111	19.8044
	C2M4	-27.28333	32.06000	1.000	-136.6411	82.0744
	C3M1	1.76333	32.06000	1.000	-107.5944	111.1211
	C3M2	-127.73000*	32.06000	.007	-237.0877	-18.3723
	C3M3	-107.11333	32.06000	.062	-216.4711	2.2444
	C3M4	29.32333	32.06000	1.000	-80.0344	138.6811
C3M0	C1M0	-142.00667*	32.06000	.001	-251.3644	-32.6489
	C2M0	-71.54000	32.06000	.642	-180.8977	37.8177
	C1M1	-144.12000*	32.06000	.001	-253.4777	-34.7623
	C1M2	-221.64000*	32.06000	.000	-330.9977	-112.2823
	C1M3	-211.04000*	32.06000	.000	-320.3977	-101.6823
	C1M4	-166.51000*	32.06000	.000	-275.8677	-57.1523
	C2M1	-97.83667	32.06000	.138	-207.1944	11.5211
	C2M2	-147.17333*	32.06000	.001	-256.5311	-37.8156
	C2M3	-161.09333*	32.06000	.000	-270.4511	-51.7356
	C2M4	-98.82333	32.06000	.128	-208.1811	10.5344
	C3M1	-69.77667	32.06000	.682	-179.1344	39.5811
	C3M2	-199.27000*	32.06000	.000	-308.6277	-89.9123
	C3M3	-178.65333*	32.06000	.000	-288.0111	-69.2956
	C3M4	-42.21667	32.06000	.993	-151.5744	67.1411
C1M1	C1M0	2.11333	32.06000	1.000	-107.2444	111.4711
	C2M0	72.58000	32.06000	.618	-36.7777	181.9377
	C3M0	144.12000*	32.06000	.001	34.7623	253.4777
	C1M2	-77.52000	32.06000	.503	-186.8777	31.8377
	C1M3	-66.92000	32.06000	.743	-176.2777	42.4377
	C1M4	-22.39000	32.06000	1.000	-131.7477	86.9677
	C2M1	46.28333	32.06000	.983	-63.0744	155.6411
	C2M2	-3.05333	32.06000	1.000	-112.4111	106.3044
	C2M3	-16.97333	32.06000	1.000	-126.3311	92.3844
	C2M4	45.29667	32.06000	.986	-64.0611	154.6544
	C3M1	74.34333	32.06000	.577	-35.0144	183.7011
	C3M2	-55.15000	32.06000	.926	-164.5077	54.2077
	C3M3	-34.53333	32.06000	.999	-143.8911	74.8244
	C3M4	101.90333	32.06000	.099	-7.4544	211.2611
C1M2	C1M0	79.63333	32.06000	.454	-29.7244	188.9911
	C2M0	150.10000*	32.06000	.000	40.7423	259.4577

	C3M0	221.64000*	32.06000	.000	112.2823	330.9977
	C1M1	77.52000	32.06000	.503	-31.8377	186.8777
	C1M3	10.60000	32.06000	1.000	-98.7577	119.9577
	C1M4	55.13000	32.06000	.926	-54.2277	164.4877
	C2M1	123.80333*	32.06000	.011	14.4456	233.1611
	C2M2	74.46667	32.06000	.574	-34.8911	183.8244
	C2M3	60.54667	32.06000	.858	-48.8111	169.9044
	C2M4	122.81667*	32.06000	.012	13.4589	232.1744
	C3M1	151.86333*	32.06000	.000	42.5056	261.2211
	C3M2	22.37000	32.06000	1.000	-86.9877	131.7277
	C3M3	42.98667	32.06000	.991	-66.3711	152.3444
	C3M4	179.42333*	32.06000	.000	70.0656	288.7811
C1M3	C1M0	69.03333	32.06000	.698	-40.3244	178.3911
	C2M0	139.50000*	32.06000	.002	30.1423	248.8577
	C3M0	211.04000*	32.06000	.000	101.6823	320.3977
	C1M1	66.92000	32.06000	.743	-42.4377	176.2777
	C1M2	-10.60000	32.06000	1.000	-119.9577	98.7577
	C1M4	44.53000	32.06000	.988	-64.8277	153.8877
	C2M1	113.20333*	32.06000	.034	3.8456	222.5611
	C2M2	63.86667	32.06000	.802	-45.4911	173.2244
	C2M3	49.94667	32.06000	.966	-59.4111	159.3044
	C2M4	112.21667*	32.06000	.038	2.8589	221.5744
	C3M1	141.26333*	32.06000	.001	31.9056	250.6211
	C3M2	11.77000	32.06000	1.000	-97.5877	121.1277
	C3M3	32.38667	32.06000	1.000	-76.9711	141.7444
	C3M4	168.82333*	32.06000	.000	59.4656	278.1811
	C1M0	24.50333	32.06000	1.000	-84.8544	133.8611
C1M4	C2M0	94.97000	32.06000	.173	-14.3877	204.3277
	C3M0	166.51000*	32.06000	.000	57.1523	275.8677
	C1M1	22.39000	32.06000	1.000	-86.9677	131.7477
	C1M2	-55.13000	32.06000	.926	-164.4877	54.2277
	C1M3	-44.53000	32.06000	.988	-153.8877	64.8277
	C2M1	68.67333	32.06000	.706	-40.6844	178.0311
	C2M2	19.33667	32.06000	1.000	-90.0211	128.6944
	C2M3	5.41667	32.06000	1.000	-103.9411	114.7744
	C2M4	67.68667	32.06000	.727	-41.6711	177.0444
	C3M1	96.73333	32.06000	.151	-12.6244	206.0911
	C3M2	-32.76000	32.06000	1.000	-142.1177	76.5977
	C3M3	-12.14333	32.06000	1.000	-121.5011	97.2144
	C3M4	124.29333*	32.06000	.010	14.9356	233.6511
	C1M0	-44.17000	32.06000	.989	-153.5277	65.1877
	C2M0	26.29667	32.06000	1.000	-83.0611	135.6544
C2M1	C3M0	97.83667	32.06000	.138	-11.5211	207.1944
	C1M1	-46.28333	32.06000	.983	-155.6411	63.0744
	C1M2	-123.80333*	32.06000	.011	-233.1611	-14.4456
	C1M3	-113.20333*	32.06000	.034	-222.5611	-3.8456
	C1M4	-68.67333	32.06000	.706	-178.0311	40.6844
	C2M2	-49.33667	32.06000	.970	-158.6944	60.0211
	C2M3	-63.25667	32.06000	.813	-172.6144	46.1011
	C2M4	-.98667	32.06000	1.000	-110.3444	108.3711
	C3M1	28.06000	32.06000	1.000	-81.2977	137.4177
	C3M2	-101.43333	32.06000	.103	-210.7911	7.9244
	C3M3	-80.81667	32.06000	.428	-190.1744	28.5411
	C3M4	55.62000	32.06000	.921	-53.7377	164.9777
	C1M0	5.16667	32.06000	1.000	-104.1911	114.5244
	C2M0	75.63333	32.06000	.547	-33.7244	184.9911
	C3M0	147.17333*	32.06000	.001	37.8156	256.5311
	C1M1	3.05333	32.06000	1.000	-106.3044	112.4111
C2M2	C1M2	-74.46667	32.06000	.574	-183.8244	34.8911
	C1M3	-63.86667	32.06000	.802	-173.2244	45.4911
	C1M4	-19.33667	32.06000	1.000	-128.6944	90.0211
	C2M1	49.33667	32.06000	.970	-60.0211	158.6944

C2M3	C2M3	-13.92000	32.06000	1.000	-123.2777	95.4377
	C2M4	48.35000	32.06000	.975	-61.0077	157.7077
	C3M1	77.39667	32.06000	.506	-31.9611	186.7544
	C3M2	-52.09667	32.06000	.952	-161.4544	57.2611
	C3M3	-31.48000	32.06000	1.000	-140.8377	77.8777
	C3M4	104.95667	32.06000	.075	-4.4011	214.3144
	C1M0	19.08667	32.06000	1.000	-90.2711	128.4444
	C2M0	89.55333	32.06000	.254	-19.8044	198.9111
	C3M0	161.09333*	32.06000	.000	51.7356	270.4511
	C1M1	16.97333	32.06000	1.000	-92.3844	126.3311
	C1M2	-60.54667	32.06000	.858	-169.9044	48.8111
	C1M3	-49.94667	32.06000	.966	-159.3044	59.4111
	C1M4	-5.41667	32.06000	1.000	-114.7744	103.9411
	C2M1	63.25667	32.06000	.813	-46.1011	172.6144
	C2M2	13.92000	32.06000	1.000	-95.4377	123.2777
	C2M4	62.27000	32.06000	.830	-47.0877	171.6277
C2M4	C3M1	91.31667	32.06000	.226	-18.0411	200.6744
	C3M2	-38.17667	32.06000	.997	-147.5344	71.1811
	C3M3	-17.56000	32.06000	1.000	-126.9177	91.7977
	C3M4	118.87667*	32.06000	.019	9.5189	228.2344
	C1M0	-43.18333	32.06000	.991	-152.5411	66.1744
	C2M0	27.28333	32.06000	1.000	-82.0744	136.6411
	C3M0	98.82333	32.06000	.128	-10.5344	208.1811
	C1M1	-45.29667	32.06000	.986	-154.6544	64.0611
	C1M2	-122.81667*	32.06000	.012	-232.1744	-13.4589
	C1M3	-112.21667*	32.06000	.038	-221.5744	-2.8589
	C1M4	-67.68667	32.06000	.727	-177.0444	41.6711
	C2M1	.98667	32.06000	1.000	-108.3711	110.3444
	C2M2	-48.35000	32.06000	.975	-157.7077	61.0077
	C2M3	-62.27000	32.06000	.830	-171.6277	47.0877
	C3M1	29.04667	32.06000	1.000	-80.3111	138.4044
	C3M2	-100.44667	32.06000	.112	-209.8044	8.9111
C3M1	C3M3	-79.83000	32.06000	.450	-189.1877	29.5277
	C3M4	56.60667	32.06000	.910	-52.7511	165.9644
	C1M0	-72.23000	32.06000	.626	-181.5877	37.1277
	C2M0	-1.76333	32.06000	1.000	-111.1211	107.5944
	C3M0	69.77667	32.06000	.682	-39.5811	179.1344
	C1M1	-74.34333	32.06000	.577	-183.7011	35.0144
	C1M2	-151.86333*	32.06000	.000	-261.2211	-42.5056
	C1M3	-141.26333*	32.06000	.001	-250.6211	-31.9056
	C1M4	-96.73333	32.06000	.151	-206.0911	12.6244
	C2M1	-28.06000	32.06000	1.000	-137.4177	81.2977
	C2M2	-77.39667	32.06000	.506	-186.7544	31.9611
	C2M3	-91.31667	32.06000	.226	-200.6744	18.0411
	C2M4	-29.04667	32.06000	1.000	-138.4044	80.3111
	C3M2	-129.49333*	32.06000	.006	-238.8511	-20.1356
	C3M3	-108.87667	32.06000	.052	-218.2344	.4811
	C3M4	27.56000	32.06000	1.000	-81.7977	136.9177
C3M2	C1M0	57.26333	32.06000	.903	-52.0944	166.6211
	C2M0	127.73000*	32.06000	.007	18.3723	237.0877
	C3M0	199.27000*	32.06000	.000	89.9123	308.6277
	C1M1	55.15000	32.06000	.926	-54.2077	164.5077
	C1M2	-22.37000	32.06000	1.000	-131.7277	86.9877
	C1M3	-11.77000	32.06000	1.000	-121.1277	97.5877
	C1M4	32.76000	32.06000	1.000	-76.5977	142.1177
	C2M1	101.43333	32.06000	.103	-7.9244	210.7911
	C2M2	52.09667	32.06000	.952	-57.2611	161.4544
	C2M3	38.17667	32.06000	.997	-71.1811	147.5344
	C2M4	100.44667	32.06000	.112	-8.9111	209.8044
	C3M1	129.49333*	32.06000	.006	20.1356	238.8511
	C3M3	20.61667	32.06000	1.000	-88.7411	129.9744
	C3M4	157.05333*	32.06000	.000	47.6956	266.4111

C3M3	C1M0	36.64667	32.06000	.998	-72.7111	146.0044
	C2M0	107.11333	32.06000	.062	-2.2444	216.4711
	C3M0	178.65333*	32.06000	.000	69.2956	288.0111
	C1M1	34.53333	32.06000	.999	-74.8244	143.8911
	C1M2	-42.98667	32.06000	.991	-152.3444	66.3711
	C1M3	-32.38667	32.06000	1.000	-141.7444	76.9711
	C1M4	12.14333	32.06000	1.000	-97.2144	121.5011
	C2M1	80.81667	32.06000	.428	-28.5411	190.1744
	C2M2	31.48000	32.06000	1.000	-77.8777	140.8377
	C2M3	17.56000	32.06000	1.000	-91.7977	126.9177
	C2M4	79.83000	32.06000	.450	-29.5277	189.1877
	C3M1	108.87667	32.06000	.052	-.4811	218.2344
	C3M2	-20.61667	32.06000	1.000	-129.9744	88.7411
	C3M4	136.43667*	32.06000	.002	27.0789	245.7944
C3M4	C1M0	-99.79000	32.06000	.118	-209.1477	9.5677
	C2M0	-29.32333	32.06000	1.000	-138.6811	80.0344
	C3M0	42.21667	32.06000	.993	-67.1411	151.5744
	C1M1	-101.90333	32.06000	.099	-211.2611	7.4544
	C1M2	-179.42333*	32.06000	.000	-288.7811	-70.0656
	C1M3	-168.82333*	32.06000	.000	-278.1811	-59.4656
	C1M4	-124.29333*	32.06000	.010	-233.6511	-14.9356
	C2M1	-55.62000	32.06000	.921	-164.9777	53.7377
	C2M2	-104.95667	32.06000	.075	-214.3144	4.4011
	C2M3	-118.87667*	32.06000	.019	-228.2344	-9.5189
	C2M4	-56.60667	32.06000	.910	-165.9644	52.7511
	C3M1	-27.56000	32.06000	1.000	-136.9177	81.7977
	C3M2	-157.05333*	32.06000	.000	-266.4111	-47.6956
	C3M3	-136.43667*	32.06000	.002	-245.7944	-27.0789

*. The mean difference is significant at the 0.05 level.

Table A.4. Tukey's Hinges percentiles distribution of fitness function value for each Ci-Mj for L1

CiMj	Percentiles						
	5	10	25	50	75	90	95
C1M0			766.3000	971.5000	991.6000		
C2M0			725.2000	807.5000	947.8000		
C3M0			631.1000	767.5000	866.8000		
C1M1			857.4000	945.7000	959.7000		
C1M2			995.4000	1005.2500	1012.7000		
C1M3			965.1000	1006.3000	1006.3000		
C1M4			856.0000	1004.3000	1007.8000		
C2M1			794.4000	873.8000	932.2000		
C2M2			818.7000	911.5500	1006.8000		
C2M3			833.9000	977.0000	989.5000		
C2M4			757.4000	911.4000	937.9000		
C3M1			689.2000	860.5500	889.6000		
C3M2			923.5000	1001.4500	1004.2000		
C3M3			938.9000	1005.6000	1005.6000		
C3M4			705.8000	786.9000	842.0000		

Table A.5. Descriptive analysis for L2

CiMj	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
C1M0	30	899.4967	82.13465	14.99567	868.8271	930.1662	687.10	990.20
C2M0	30	874.5633	93.91316	17.14612	839.4956	909.6311	734.40	1001.20
C3M0	30	860.1167	96.98941	17.70776	823.9002	896.3331	709.60	995.70
C1M1	30	901.8800	101.73865	18.57485	863.8902	939.8698	637.60	1002.80

C1M2	30	991.4767	59.51941	10.86671	969.2518	1013.7016	740.80	1013.90
C1M3	30	839.8000	133.41547	24.35822	789.9818	889.6182	629.70	1005.00
C1M4	30	819.6200	130.98204	23.91394	770.7105	868.5295	555.50	1006.50
C2M1	30	933.4233	90.08075	16.44642	899.7866	967.0600	664.70	1007.10
C2M2	30	864.2300	107.89891	19.69956	823.9399	904.5201	650.10	1006.00
C2M3	30	899.8633	108.49144	19.80774	859.3520	940.3747	680.30	1005.20
C2M4	30	891.1200	134.59501	24.57357	840.8614	941.3786	580.60	1004.80
C3M1	30	923.9567	114.08806	20.82953	881.3555	966.5578	602.40	1006.70
C3M2	30	962.0900	79.57837	14.52896	932.3749	991.8051	674.30	1008.20
C3M3	30	984.3833	61.22559	11.17821	961.5213	1007.2453	721.70	1007.60
C3M4	30	948.0133	112.08303	20.46347	906.1608	989.8658	576.50	1007.10
Total	450	906.2689	112.74747	5.31497	895.8236	916.7142	555.50	1013.90

Table A.6 ANOVA test result for L2

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1093061.242	14	78075.803	7.360	.000
Within Groups	4614622.962	435	10608.329		
Total	5707684.204	449			

(a)

(b)

(c)

Figure A.2.(a) Profile plot for estimated marginal means of fitness value for each Ci-Mj for L2. (b) Box plot of average fitness function values achieved on L2 by different Ci-Mj. (c) Error bar plot of mean average fitness function values Vs. Ci-Mj.

Table A.7 Posthoc test for L2

(I) CiMj	(J) CiMj	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval		
					Lower Bound	Upper Bound	
C1M0	C2M0	24.93333	26.59364	1.000	-65.7785	115.6451	
	C3M0	39.38000	26.59364	.978	-51.3318	130.0918	
	C1M1	-2.38333	26.59364	1.000	-93.0951	88.3285	
	C1M2	-91.98000*	26.59364	.043	-182.6918	-1.2682	
	C1M3	59.69667	26.59364	.632	-31.0151	150.4085	
	C1M4	79.87667	26.59364	.156	-10.8351	170.5885	
	C2M1	-33.92667	26.59364	.995	-124.6385	56.7851	
	C2M2	35.26667	26.59364	.992	-55.4451	125.9785	
	C2M3	-.36667	26.59364	1.000	-91.0785	90.3451	
	C2M4	8.37667	26.59364	1.000	-82.3351	99.0885	
	C3M1	-24.46000	26.59364	1.000	-115.1718	66.2518	
	C3M2	-62.59333	26.59364	.551	-153.3051	28.1185	
	C3M3	-84.88667	26.59364	.095	-175.5985	5.8251	
	C3M4	-48.51667	26.59364	.887	-139.2285	42.1951	
	C2M0	C1M0	-24.93333	26.59364	1.000	-115.6451	65.7785
		C3M0	14.44667	26.59364	1.000	-76.2651	105.1585
C1M1		-27.31667	26.59364	.999	-118.0285	63.3951	
C1M2		-116.91333*	26.59364	.001	-207.6251	-26.2015	
C1M3		34.76333	26.59364	.993	-55.9485	125.4751	
C1M4		54.94333	26.59364	.757	-35.7685	145.6551	
C2M1		-58.86000	26.59364	.655	-149.5718	31.8518	
C2M2		10.33333	26.59364	1.000	-80.3785	101.0451	
C2M3		-25.30000	26.59364	1.000	-116.0118	65.4118	
C2M4		-16.55667	26.59364	1.000	-107.2685	74.1551	
C3M1		-49.39333	26.59364	.873	-140.1051	41.3185	
C3M2		-87.52667	26.59364	.072	-178.2385	3.1851	
C3M3		-109.82000*	26.59364	.004	-200.5318	-19.1082	
C3M4		-73.45000	26.59364	.272	-164.1618	17.2618	
C3M0		C1M0	-39.38000	26.59364	.978	-130.0918	51.3318
		C2M0	-14.44667	26.59364	1.000	-105.1585	76.2651
	C1M1	-41.76333	26.59364	.964	-132.4751	48.9485	
	C1M2	-131.36000*	26.59364	.000	-222.0718	-40.6482	
	C1M3	20.31667	26.59364	1.000	-70.3951	111.0285	
	C1M4	40.49667	26.59364	.972	-50.2151	131.2085	
	C2M1	-73.30667	26.59364	.275	-164.0185	17.4051	
	C2M2	-4.11333	26.59364	1.000	-94.8251	86.5985	
	C2M3	-39.74667	26.59364	.977	-130.4585	50.9651	
	C2M4	-31.00333	26.59364	.998	-121.7151	59.7085	
	C3M1	-63.84000	26.59364	.516	-154.5518	26.8718	
	C3M2	-101.97333*	26.59364	.012	-192.6851	-11.2615	
	C3M3	-124.26667*	26.59364	.000	-214.9785	-33.5549	
	C3M4	-87.89667	26.59364	.069	-178.6085	2.8151	
	C1M1	C1M0	2.38333	26.59364	1.000	-88.3285	93.0951
		C2M0	27.31667	26.59364	.999	-63.3951	118.0285
C3M0		41.76333	26.59364	.964	-48.9485	132.4751	
C1M2		-89.59667	26.59364	.057	-180.3085	1.1151	
C1M3		62.08000	26.59364	.565	-28.6318	152.7918	
C1M4		82.26000	26.59364	.124	-8.4518	172.9718	
C2M1		-31.54333	26.59364	.998	-122.2551	59.1685	
C2M2		37.65000	26.59364	.986	-53.0618	128.3618	
C2M3		2.01667	26.59364	1.000	-88.6951	92.7285	
C2M4		10.76000	26.59364	1.000	-79.9518	101.4718	
C3M1		-22.07667	26.59364	1.000	-112.7885	68.6351	
C3M2		-60.21000	26.59364	.618	-150.9218	30.5018	
C3M3		-82.50333	26.59364	.121	-173.2151	8.2085	
C3M4		-46.13333	26.59364	.921	-136.8451	44.5785	
C1M2		C1M0	91.98000*	26.59364	.043	1.2682	182.6918
		C2M0	116.91333*	26.59364	.001	26.2015	207.6251

	C3M0	131.36000*	26.59364	.000	40.6482	222.0718
	C1M1	89.59667	26.59364	.057	-1.1151	180.3085
	C1M3	151.67667*	26.59364	.000	60.9649	242.3885
	C1M4	171.85667*	26.59364	.000	81.1449	262.5685
	C2M1	58.05333	26.59364	.677	-32.6585	148.7651
	C2M2	127.24667*	26.59364	.000	36.5349	217.9585
	C2M3	91.61333*	26.59364	.045	.9015	182.3251
	C2M4	100.35667*	26.59364	.015	9.6449	191.0685
	C3M1	67.52000	26.59364	.415	-23.1918	158.2318
	C3M2	29.38667	26.59364	.999	-61.3251	120.0985
	C3M3	7.09333	26.59364	1.000	-83.6185	97.8051
	C3M4	43.46333	26.59364	.950	-47.2485	134.1751
C1M3	C1M0	-59.69667	26.59364	.632	-150.4085	31.0151
	C2M0	-34.76333	26.59364	.993	-125.4751	55.9485
	C3M0	-20.31667	26.59364	1.000	-111.0285	70.3951
	C1M1	-62.08000	26.59364	.565	-152.7918	28.6318
	C1M2	-151.67667*	26.59364	.000	-242.3885	-60.9649
	C1M4	20.18000	26.59364	1.000	-70.5318	110.8918
	C2M1	-93.62333*	26.59364	.035	-184.3351	-2.9115
	C2M2	-24.43000	26.59364	1.000	-115.1418	66.2818
	C2M3	-60.06333	26.59364	.622	-150.7751	30.6485
	C2M4	-51.32000	26.59364	.837	-142.0318	39.3918
	C3M1	-84.15667	26.59364	.103	-174.8685	6.5551
	C3M2	-122.29000*	26.59364	.001	-213.0018	-31.5782
C1M4	C3M3	-144.58333*	26.59364	.000	-235.2951	-53.8715
	C3M4	-108.21333*	26.59364	.005	-198.9251	-17.5015
	C1M0	-79.87667	26.59364	.156	-170.5885	10.8351
	C2M0	-54.94333	26.59364	.757	-145.6551	35.7685
	C3M0	-40.49667	26.59364	.972	-131.2085	50.2151
	C1M1	-82.26000	26.59364	.124	-172.9718	8.4518
	C1M2	-171.85667*	26.59364	.000	-262.5685	-81.1449
	C1M3	-20.18000	26.59364	1.000	-110.8918	70.5318
	C2M1	-113.80333*	26.59364	.002	-204.5151	-23.0915
	C2M2	-44.61000	26.59364	.939	-135.3218	46.1018
	C2M3	-80.24333	26.59364	.151	-170.9551	10.4685
	C2M4	-71.50000	26.59364	.315	-162.2118	19.2118
C2M1	C3M1	-104.33667*	26.59364	.009	-195.0485	-13.6249
	C3M2	-142.47000*	26.59364	.000	-233.1818	-51.7582
	C3M3	-164.76333*	26.59364	.000	-255.4751	-74.0515
	C3M4	-128.39333*	26.59364	.000	-219.1051	-37.6815
	C1M0	33.92667	26.59364	.995	-56.7851	124.6385
	C2M0	58.86000	26.59364	.655	-31.8518	149.5718
	C3M0	73.30667	26.59364	.275	-17.4051	164.0185
	C1M1	31.54333	26.59364	.998	-59.1685	122.2551
	C1M2	-58.05333	26.59364	.677	-148.7651	32.6585
	C1M3	93.62333*	26.59364	.035	2.9115	184.3351
	C1M4	113.80333*	26.59364	.002	23.0915	204.5151
	C2M2	69.19333	26.59364	.371	-21.5185	159.9051
C2M2	C2M3	33.56000	26.59364	.995	-57.1518	124.2718
	C2M4	42.30333	26.59364	.960	-48.4085	133.0151
	C3M1	9.46667	26.59364	1.000	-81.2451	100.1785
	C3M2	-28.66667	26.59364	.999	-119.3785	62.0451
	C3M3	-50.96000	26.59364	.844	-141.6718	39.7518
	C3M4	-14.59000	26.59364	1.000	-105.3018	76.1218
	C1M0	-35.26667	26.59364	.992	-125.9785	55.4451
	C2M0	-10.33333	26.59364	1.000	-101.0451	80.3785
	C3M0	4.11333	26.59364	1.000	-86.5985	94.8251
	C1M1	-37.65000	26.59364	.986	-128.3618	53.0618
	C1M2	-127.24667*	26.59364	.000	-217.9585	-36.5349
	C1M3	24.43000	26.59364	1.000	-66.2818	115.1418
	C1M4	44.61000	26.59364	.939	-46.1018	135.3218
	C2M1	-69.19333	26.59364	.371	-159.9051	21.5185

C2M3	C2M3	-35.63333	26.59364	.991	-126.3451	55.0785
	C2M4	-26.89000	26.59364	1.000	-117.6018	63.8218
	C3M1	-59.72667	26.59364	.631	-150.4385	30.9851
	C3M2	-97.86000*	26.59364	.021	-188.5718	-7.1482
	C3M3	-120.15333*	26.59364	.001	-210.8651	-29.4415
	C3M4	-83.78333	26.59364	.107	-174.4951	6.9285
	C1M0	.36667	26.59364	1.000	-90.3451	91.0785
	C2M0	25.30000	26.59364	1.000	-65.4118	116.0118
	C3M0	39.74667	26.59364	.977	-50.9651	130.4585
	C1M1	-2.01667	26.59364	1.000	-92.7285	88.6951
	C1M2	-91.61333*	26.59364	.045	-182.3251	-.9015
	C1M3	60.06333	26.59364	.622	-30.6485	150.7751
	C1M4	80.24333	26.59364	.151	-10.4685	170.9551
	C2M1	-33.56000	26.59364	.995	-124.2718	57.1518
	C2M2	35.63333	26.59364	.991	-55.0785	126.3451
C2M4	C2M4	8.74333	26.59364	1.000	-81.9685	99.4551
	C3M1	-24.09333	26.59364	1.000	-114.8051	66.6185
	C3M2	-62.22667	26.59364	.561	-152.9385	28.4851
	C3M3	-84.52000	26.59364	.099	-175.2318	6.1918
	C3M4	-48.15000	26.59364	.893	-138.8618	42.5618
	C1M0	-8.37667	26.59364	1.000	-99.0885	82.3351
	C2M0	16.55667	26.59364	1.000	-74.1551	107.2685
	C3M0	31.00333	26.59364	.998	-59.7085	121.7151
	C1M1	-10.76000	26.59364	1.000	-101.4718	79.9518
	C1M2	-100.35667*	26.59364	.015	-191.0685	-9.6449
	C1M3	51.32000	26.59364	.837	-39.3918	142.0318
	C1M4	71.50000	26.59364	.315	-19.2118	162.2118
	C2M1	-42.30333	26.59364	.960	-133.0151	48.4085
	C2M2	26.89000	26.59364	1.000	-63.8218	117.6018
	C2M3	-8.74333	26.59364	1.000	-99.4551	81.9685
C3M1	C3M1	-32.83667	26.59364	.996	-123.5485	57.8751
	C3M2	-70.97000	26.59364	.328	-161.6818	19.7418
	C3M3	-93.26333*	26.59364	.037	-183.9751	-2.5515
	C3M4	-56.89333	26.59364	.708	-147.6051	33.8185
	C1M0	24.46000	26.59364	1.000	-66.2518	115.1718
	C2M0	49.39333	26.59364	.873	-41.3185	140.1051
	C3M0	63.84000	26.59364	.516	-26.8718	154.5518
	C1M1	22.07667	26.59364	1.000	-68.6351	112.7885
	C1M2	-67.52000	26.59364	.415	-158.2318	23.1918
	C1M3	84.15667	26.59364	.103	-6.5551	174.8685
	C1M4	104.33667*	26.59364	.009	13.6249	195.0485
	C2M1	-9.46667	26.59364	1.000	-100.1785	81.2451
	C2M2	59.72667	26.59364	.631	-30.9851	150.4385
	C2M3	24.09333	26.59364	1.000	-66.6185	114.8051
	C2M4	32.83667	26.59364	.996	-57.8751	123.5485
C3M2	C3M2	-38.13333	26.59364	.984	-128.8451	52.5785
	C3M3	-60.42667	26.59364	.612	-151.1385	30.2851
	C3M4	-24.05667	26.59364	1.000	-114.7685	66.6551
	C1M0	62.59333	26.59364	.551	-28.1185	153.3051
	C2M0	87.52667	26.59364	.072	-3.1851	178.2385
	C3M0	101.97333*	26.59364	.012	11.2615	192.6851
	C1M1	60.21000	26.59364	.618	-30.5018	150.9218
	C1M2	-29.38667	26.59364	.999	-120.0985	61.3251
	C1M3	122.29000*	26.59364	.001	31.5782	213.0018
	C1M4	142.47000*	26.59364	.000	51.7582	233.1818
	C2M1	28.66667	26.59364	.999	-62.0451	119.3785
	C2M2	97.86000*	26.59364	.021	7.1482	188.5718
	C2M3	62.22667	26.59364	.561	-28.4851	152.9385
	C2M4	70.97000	26.59364	.328	-19.7418	161.6818
	C3M1	38.13333	26.59364	.984	-52.5785	128.8451
	C3M3	-22.29333	26.59364	1.000	-113.0051	68.4185
	C3M4	14.07667	26.59364	1.000	-76.6351	104.7885

C3M3	C1M0	84.88667	26.59364	.095	-5.8251	175.5985
	C2M0	109.82000*	26.59364	.004	19.1082	200.5318
	C3M0	124.26667*	26.59364	.000	33.5549	214.9785
	C1M1	82.50333	26.59364	.121	-8.2085	173.2151
	C1M2	-7.09333	26.59364	1.000	-97.8051	83.6185
	C1M3	144.58333*	26.59364	.000	53.8715	235.2951
	C1M4	164.76333*	26.59364	.000	74.0515	255.4751
	C2M1	50.96000	26.59364	.844	-39.7518	141.6718
	C2M2	120.15333*	26.59364	.001	29.4415	210.8651
	C2M3	84.52000	26.59364	.099	-6.1918	175.2318
	C2M4	93.26333*	26.59364	.037	2.5515	183.9751
	C3M1	60.42667	26.59364	.612	-30.2851	151.1385
	C3M2	22.29333	26.59364	1.000	-68.4185	113.0051
	C3M4	36.37000	26.59364	.990	-54.3418	127.0818
C3M4	C1M0	48.51667	26.59364	.887	-42.1951	139.2285
	C2M0	73.45000	26.59364	.272	-17.2618	164.1618
	C3M0	87.89667	26.59364	.069	-2.8151	178.6085
	C1M1	46.13333	26.59364	.921	-44.5785	136.8451
	C1M2	-43.46333	26.59364	.950	-134.1751	47.2485
	C1M3	108.21333*	26.59364	.005	17.5015	198.9251
	C1M4	128.39333*	26.59364	.000	37.6815	219.1051
	C2M1	14.59000	26.59364	1.000	-76.1218	105.3018
	C2M2	83.78333	26.59364	.107	-6.9285	174.4951
	C2M3	48.15000	26.59364	.893	-42.5618	138.8618
	C2M4	56.89333	26.59364	.708	-33.8185	147.6051
	C3M1	24.05667	26.59364	1.000	-66.6551	114.7685
	C3M2	-14.07667	26.59364	1.000	-104.7885	76.6351
	C3M3	-36.37000	26.59364	.990	-127.0818	54.3418

*. The mean difference is significant at the 0.05 level.

960 **Table A.8.** Tukey's Hinges percentiles distribution of fitness function value for each Ci-Mj for L2

CiMj	Percentiles						
	5	10	25	50	75	90	95
C1M0			826.1000	912.7500	980.0000		
C2M0			795.0000	827.8500	970.1000		
C3M0			768.7000	826.1000	970.6000		
C1M1			865.6000	910.4000	980.7000		
C1M2			1008.4000	1013.9000	1013.9000		
C1M3			793.8000	806.8000	990.2000		
C1M4			709.6000	827.8500	950.2000		
C2M1			895.4000	976.5000	995.4000		
C2M2			793.8000	874.6000	950.7000		
C2M3			810.6000	950.3500	990.8000		
C2M4			822.7000	985.7000	999.1000		
C3M1			880.2000	970.6000	1002.3000		
C3M2			984.1000	989.4500	994.8000		
C3M3			1002.7000	1004.8000	1007.6000		
C3M4			967.9000	992.9500	1005.0000		

961
962 **Table A.9.** Descriptive analysis for L3

CiMj	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
C1M0	30	810.6033	105.98789	19.35065	771.0268	850.1799	579.10	990.10
C2M0	30	825.7977	83.80856	15.30128	794.5030	857.0923	731.80	980.20
C3M0	30	777.6400	107.89184	19.69827	737.3525	817.9275	645.20	980.20
C1M1	30	754.2767	99.64700	18.19297	717.0679	791.4855	621.30	950.40
C1M2	30	971.7100	65.48949	11.95669	947.2558	996.1642	818.20	1013.40
C1M3	30	813.7900	119.64057	21.84328	769.1155	858.4645	630.20	1003.00
C1M4	30	841.6733	83.15842	15.18258	810.6215	872.7252	708.00	1003.50

C2M1	30	828.0667	125.06851	22.83428	781.3653	874.7680	625.50	1002.30
C2M2	30	771.1200	130.70749	23.86381	722.3130	819.9270	570.30	1004.70
C2M3	30	843.6633	73.16497	13.35804	816.3431	870.9836	661.60	1003.20
C2M4	30	794.2933	119.46903	21.81196	749.6829	838.9038	614.20	1002.20
C3M1	30	847.2567	133.18323	24.31582	797.5252	896.9881	631.10	1005.00
C3M2	30	819.7533	130.84154	23.88829	770.8963	868.6104	645.00	1007.10
C3M3	30	788.1033	136.98775	25.01043	736.9513	839.2554	577.40	1002.20
C3M4	30	788.1033	136.98775	25.01043	736.9513	839.2554	577.40	1002.20
Total	450	818.3901	121.22000	5.71437	807.1598	829.6203	570.30	1013.40

Table A.10. ANOVA test result for L3

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1085222.181	14	77515.870	6.117	.000
Within Groups	5512513.081	435	12672.444		
Total	6597735.262	449			

(a)

(b)

(c)

Figure A.3.(a) Profile plot for estimated marginal means of fitness value for each Ci-Mj for L3. (b) Box plot of average fitness function values achieved on L3 by different Ci-Mj. (c) Error bar plot of mean average fitness function values Vs. Ci-Mj.

Table A.11 Posthoc test for L3

(I) CiMj	(J) CiMj	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
C1M0	C2M0	-15.19433	29.06595	1.000	-114.3392	83.9506
	C3M0	32.96333	29.06595	.998	-66.1816	132.1082
	C1M1	56.32667	29.06595	.833	-42.8182	155.4716
	C1M2	-161.10667*	29.06595	.000	-260.2516	-61.9618
	C1M3	-3.18667	29.06595	1.000	-102.3316	95.9582
	C1M4	-31.07000	29.06595	.999	-130.2149	68.0749
	C2M1	-17.46333	29.06595	1.000	-116.6082	81.6816
	C2M2	39.48333	29.06595	.990	-59.6616	138.6282
	C2M3	-33.06000	29.06595	.998	-132.2049	66.0849
	C2M4	16.31000	29.06595	1.000	-82.8349	115.4549
	C3M1	-36.65333	29.06595	.995	-135.7982	62.4916
	C3M2	-9.15000	29.06595	1.000	-108.2949	89.9949
	C3M3	22.50000	29.06595	1.000	-76.6449	121.6449
	C3M4	22.50000	29.06595	1.000	-76.6449	121.6449
C2M0	C1M0	15.19433	29.06595	1.000	-83.9506	114.3392
	C3M0	48.15767	29.06595	.945	-50.9872	147.3026
	C1M1	71.52100	29.06595	.471	-27.6239	170.6659
	C1M2	-145.91233*	29.06595	.000	-245.0572	-46.7674
	C1M3	12.00767	29.06595	1.000	-87.1372	111.1526
	C1M4	-15.87567	29.06595	1.000	-115.0206	83.2692
	C2M1	-2.26900	29.06595	1.000	-101.4139	96.8759
	C2M2	54.67767	29.06595	.861	-44.4672	153.8226
	C2M3	-17.86567	29.06595	1.000	-117.0106	81.2792
	C2M4	31.50433	29.06595	.999	-67.6406	130.6492
	C3M1	-21.45900	29.06595	1.000	-120.6039	77.6859
	C3M2	6.04433	29.06595	1.000	-93.1006	105.1892
	C3M3	37.69433	29.06595	.994	-61.4506	136.8392
	C3M4	37.69433	29.06595	.994	-61.4506	136.8392
C3M0	C1M0	-32.96333	29.06595	.998	-132.1082	66.1816
	C2M0	-48.15767	29.06595	.945	-147.3026	50.9872
	C1M1	23.36333	29.06595	1.000	-75.7816	122.5082
	C1M2	-194.07000*	29.06595	.000	-293.2149	-94.9251
	C1M3	-36.15000	29.06595	.996	-135.2949	62.9949
	C1M4	-64.03333	29.06595	.663	-163.1782	35.1116
	C2M1	-50.42667	29.06595	.921	-149.5716	48.7182
	C2M2	6.52000	29.06595	1.000	-92.6249	105.6649
	C2M3	-66.02333	29.06595	.612	-165.1682	33.1216
	C2M4	-16.65333	29.06595	1.000	-115.7982	82.4916
	C3M1	-69.61667	29.06595	.520	-168.7616	29.5282
	C3M2	-42.11333	29.06595	.982	-141.2582	57.0316
	C3M3	-10.46333	29.06595	1.000	-109.6082	88.6816
	C3M4	-10.46333	29.06595	1.000	-109.6082	88.6816
C1M1	C1M0	-56.32667	29.06595	.833	-155.4716	42.8182
	C2M0	-71.52100	29.06595	.471	-170.6659	27.6239
	C3M0	-23.36333	29.06595	1.000	-122.5082	75.7816
	C1M2	-217.43333*	29.06595	.000	-316.5782	-118.2884
	C1M3	-59.51333	29.06595	.768	-158.6582	39.6316
	C1M4	-87.39667	29.06595	.155	-186.5416	11.7482
	C2M1	-73.79000	29.06595	.415	-172.9349	25.3549
	C2M2	-16.84333	29.06595	1.000	-115.9882	82.3016
	C2M3	-89.38667	29.06595	.130	-188.5316	9.7582
	C2M4	-40.01667	29.06595	.989	-139.1616	59.1282
	C3M1	-92.98000	29.06595	.093	-192.1249	6.1649
	C3M2	-65.47667	29.06595	.626	-164.6216	33.6682
	C3M3	-33.82667	29.06595	.998	-132.9716	65.3182
	C3M4	-33.82667	29.06595	.998	-132.9716	65.3182
C1M2	C1M0	161.10667*	29.06595	.000	61.9618	260.2516
	C2M0	145.91233*	29.06595	.000	46.7674	245.0572

	C3M0	194.07000*	29.06595	.000	94.9251	293.2149
	C1M1	217.43333*	29.06595	.000	118.2884	316.5782
	C1M3	157.92000*	29.06595	.000	58.7751	257.0649
	C1M4	130.03667*	29.06595	.001	30.8918	229.1816
	C2M1	143.64333*	29.06595	.000	44.4984	242.7882
	C2M2	200.59000*	29.06595	.000	101.4451	299.7349
	C2M3	128.04667*	29.06595	.001	28.9018	227.1916
	C2M4	177.41667*	29.06595	.000	78.2718	276.5616
	C3M1	124.45333*	29.06595	.002	25.3084	223.5982
	C3M2	151.95667*	29.06595	.000	52.8118	251.1016
	C3M3	183.60667*	29.06595	.000	84.4618	282.7516
	C3M4	183.60667*	29.06595	.000	84.4618	282.7516
C1M3	C1M0	3.18667	29.06595	1.000	-95.9582	102.3316
	C2M0	-12.00767	29.06595	1.000	-111.1526	87.1372
	C3M0	36.15000	29.06595	.996	-62.9949	135.2949
	C1M1	59.51333	29.06595	.768	-39.6316	158.6582
	C1M2	-157.92000*	29.06595	.000	-257.0649	-58.7751
	C1M4	-27.88333	29.06595	1.000	-127.0282	71.2616
	C2M1	-14.27667	29.06595	1.000	-113.4216	84.8682
	C2M2	42.67000	29.06595	.980	-56.4749	141.8149
	C2M3	-29.87333	29.06595	.999	-129.0182	69.2716
	C2M4	19.49667	29.06595	1.000	-79.6482	118.6416
	C3M1	-33.46667	29.06595	.998	-132.6116	65.6782
	C3M2	-5.96333	29.06595	1.000	-105.1082	93.1816
C1M4	C3M3	25.68667	29.06595	1.000	-73.4582	124.8316
	C3M4	25.68667	29.06595	1.000	-73.4582	124.8316
	C1M0	31.07000	29.06595	.999	-68.0749	130.2149
	C2M0	15.87567	29.06595	1.000	-83.2692	115.0206
	C3M0	64.03333	29.06595	.663	-35.1116	163.1782
	C1M1	87.39667	29.06595	.155	-11.7482	186.5416
	C1M2	-130.03667*	29.06595	.001	-229.1816	-30.8918
	C1M3	27.88333	29.06595	1.000	-71.2616	127.0282
	C2M1	13.60667	29.06595	1.000	-85.5382	112.7516
	C2M2	70.55333	29.06595	.496	-28.5916	169.6982
	C2M3	-1.99000	29.06595	1.000	-101.1349	97.1549
	C2M4	47.38000	29.06595	.951	-51.7649	146.5249
C2M1	C3M1	-5.58333	29.06595	1.000	-104.7282	93.5616
	C3M2	21.92000	29.06595	1.000	-77.2249	121.0649
	C3M3	53.57000	29.06595	.879	-45.5749	152.7149
	C3M4	53.57000	29.06595	.879	-45.5749	152.7149
	C1M0	17.46333	29.06595	1.000	-81.6816	116.6082
	C2M0	2.26900	29.06595	1.000	-96.8759	101.4139
	C3M0	50.42667	29.06595	.921	-48.7182	149.5716
	C1M1	73.79000	29.06595	.415	-25.3549	172.9349
	C1M2	-143.64333*	29.06595	.000	-242.7882	-44.4984
	C1M3	14.27667	29.06595	1.000	-84.8682	113.4216
	C1M4	-13.60667	29.06595	1.000	-112.7516	85.5382
	C2M2	56.94667	29.06595	.821	-42.1982	156.0916
C2M2	C2M3	-15.59667	29.06595	1.000	-114.7416	83.5482
	C2M4	33.77333	29.06595	.998	-65.3716	132.9182
	C3M1	-19.19000	29.06595	1.000	-118.3349	79.9549
	C3M2	8.31333	29.06595	1.000	-90.8316	107.4582
	C3M3	39.96333	29.06595	.989	-59.1816	139.1082
	C3M4	39.96333	29.06595	.989	-59.1816	139.1082
	C1M0	-39.48333	29.06595	.990	-138.6282	59.6616
	C2M0	-54.67767	29.06595	.861	-153.8226	44.4672
	C3M0	-6.52000	29.06595	1.000	-105.6649	92.6249
	C1M1	16.84333	29.06595	1.000	-82.3016	115.9882
	C1M2	-200.59000*	29.06595	.000	-299.7349	-101.4451
	C1M3	-42.67000	29.06595	.980	-141.8149	56.4749
	C1M4	-70.55333	29.06595	.496	-169.6982	28.5916
	C2M1	-56.94667	29.06595	.821	-156.0916	42.1982

C2M3	C2M3	-72.54333	29.06595	.446	-171.6882	26.6016
	C2M4	-23.17333	29.06595	1.000	-122.3182	75.9716
	C3M1	-76.13667	29.06595	.360	-175.2816	23.0082
	C3M2	-48.63333	29.06595	.940	-147.7782	50.5116
	C3M3	-16.98333	29.06595	1.000	-116.1282	82.1616
	C3M4	-16.98333	29.06595	1.000	-116.1282	82.1616
	C1M0	33.06000	29.06595	.998	-66.0849	132.2049
	C2M0	17.86567	29.06595	1.000	-81.2792	117.0106
	C3M0	66.02333	29.06595	.612	-33.1216	165.1682
	C1M1	89.38667	29.06595	.130	-9.7582	188.5316
	C1M2	-128.04667*	29.06595	.001	-227.1916	-28.9018
	C1M3	29.87333	29.06595	.999	-69.2716	129.0182
	C1M4	1.99000	29.06595	1.000	-97.1549	101.1349
	C2M1	15.59667	29.06595	1.000	-83.5482	114.7416
	C2M2	72.54333	29.06595	.446	-26.6016	171.6882
C2M4	C2M4	49.37000	29.06595	.933	-49.7749	148.5149
	C3M1	-3.59333	29.06595	1.000	-102.7382	95.5516
	C3M2	23.91000	29.06595	1.000	-75.2349	123.0549
	C3M3	55.56000	29.06595	.846	-43.5849	154.7049
	C3M4	55.56000	29.06595	.846	-43.5849	154.7049
	C1M0	-16.31000	29.06595	1.000	-115.4549	82.8349
	C2M0	-31.50433	29.06595	.999	-130.6492	67.6406
	C3M0	16.65333	29.06595	1.000	-82.4916	115.7982
	C1M1	40.01667	29.06595	.989	-59.1282	139.1616
	C1M2	-177.41667*	29.06595	.000	-276.5616	-78.2718
	C1M3	-19.49667	29.06595	1.000	-118.6416	79.6482
	C1M4	-47.38000	29.06595	.951	-146.5249	51.7649
	C2M1	-33.77333	29.06595	.998	-132.9182	65.3716
	C2M2	23.17333	29.06595	1.000	-75.9716	122.3182
	C2M3	-49.37000	29.06595	.933	-148.5149	49.7749
C3M1	C3M1	-52.96333	29.06595	.888	-152.1082	46.1816
	C3M2	-25.46000	29.06595	1.000	-124.6049	73.6849
	C3M3	6.19000	29.06595	1.000	-92.9549	105.3349
	C3M4	6.19000	29.06595	1.000	-92.9549	105.3349
	C1M0	36.65333	29.06595	.995	-62.4916	135.7982
	C2M0	21.45900	29.06595	1.000	-77.6859	120.6039
	C3M0	69.61667	29.06595	.520	-29.5282	168.7616
	C1M1	92.98000	29.06595	.093	-6.1649	192.1249
	C1M2	-124.45333*	29.06595	.002	-223.5982	-25.3084
	C1M3	33.46667	29.06595	.998	-65.6782	132.6116
	C1M4	5.58333	29.06595	1.000	-93.5616	104.7282
	C2M1	19.19000	29.06595	1.000	-79.9549	118.3349
	C2M2	76.13667	29.06595	.360	-23.0082	175.2816
	C2M3	3.59333	29.06595	1.000	-95.5516	102.7382
	C2M4	52.96333	29.06595	.888	-46.1816	152.1082
C3M2	C3M2	27.50333	29.06595	1.000	-71.6416	126.6482
	C3M3	59.15333	29.06595	.776	-39.9916	158.2982
	C3M4	59.15333	29.06595	.776	-39.9916	158.2982
	C1M0	9.15000	29.06595	1.000	-89.9949	108.2949
	C2M0	-6.04433	29.06595	1.000	-105.1892	93.1006
	C3M0	42.11333	29.06595	.982	-57.0316	141.2582
	C1M1	65.47667	29.06595	.626	-33.6682	164.6216
	C1M2	-151.95667*	29.06595	.000	-251.1016	-52.8118
	C1M3	5.96333	29.06595	1.000	-93.1816	105.1082
	C1M4	-21.92000	29.06595	1.000	-121.0649	77.2249
	C2M1	-8.31333	29.06595	1.000	-107.4582	90.8316
	C2M2	48.63333	29.06595	.940	-50.5116	147.7782
	C2M3	-23.91000	29.06595	1.000	-123.0549	75.2349
	C2M4	25.46000	29.06595	1.000	-73.6849	124.6049
	C3M1	-27.50333	29.06595	1.000	-126.6482	71.6416
	C3M3	31.65000	29.06595	.999	-67.4949	130.7949
	C3M4	31.65000	29.06595	.999	-67.4949	130.7949

C3M3	C1M0	-22.50000	29.06595	1.000	-121.6449	76.6449
	C2M0	-37.69433	29.06595	.994	-136.8392	61.4506
	C3M0	10.46333	29.06595	1.000	-88.6816	109.6082
	C1M1	33.82667	29.06595	.998	-65.3182	132.9716
	C1M2	-183.60667*	29.06595	.000	-282.7516	-84.4618
	C1M3	-25.68667	29.06595	1.000	-124.8316	73.4582
	C1M4	-53.57000	29.06595	.879	-152.7149	45.5749
	C2M1	-39.96333	29.06595	.989	-139.1082	59.1816
	C2M2	16.98333	29.06595	1.000	-82.1616	116.1282
	C2M3	-55.56000	29.06595	.846	-154.7049	43.5849
	C2M4	-6.19000	29.06595	1.000	-105.3349	92.9549
	C3M1	-59.15333	29.06595	.776	-158.2982	39.9916
	C3M2	-31.65000	29.06595	.999	-130.7949	67.4949
	C3M4	.00000	29.06595	1.000	-99.1449	99.1449
C3M4	C1M0	-22.50000	29.06595	1.000	-121.6449	76.6449
	C2M0	-37.69433	29.06595	.994	-136.8392	61.4506
	C3M0	10.46333	29.06595	1.000	-88.6816	109.6082
	C1M1	33.82667	29.06595	.998	-65.3182	132.9716
	C1M2	-183.60667*	29.06595	.000	-282.7516	-84.4618
	C1M3	-25.68667	29.06595	1.000	-124.8316	73.4582
	C1M4	-53.57000	29.06595	.879	-152.7149	45.5749
	C2M1	-39.96333	29.06595	.989	-139.1082	59.1816
	C2M2	16.98333	29.06595	1.000	-82.1616	116.1282
	C2M3	-55.56000	29.06595	.846	-154.7049	43.5849
	C2M4	-6.19000	29.06595	1.000	-105.3349	92.9549
	C3M1	-59.15333	29.06595	.776	-158.2982	39.9916
	C3M2	-31.65000	29.06595	.999	-130.7949	67.4949
	C3M3	.00000	29.06595	1.000	-99.1449	99.1449

*. The mean difference is significant at the 0.05 level.

975

Table A.12. Tukey's Hinges percentiles distribution of fitness function value for each Ci-Mj for L3

CiMj	Percentiles						
	5	10	25	50	75	90	95
C1M0			770.4000	812.3000	895.3000		
C2M0			760.8000	789.7000	880.7000		
C3M0			731.8000	746.8000	840.6000		
C1M1			679.1000	712.8500	820.4000		
C1M2			973.4000	1005.8500	1012.9000		
C1M3			690.6000	830.1000	880.2000		
C1M4			790.5000	828.0000	880.2000		
C2M1			731.8000	820.6000	895.3000		
C2M2			621.3000	760.8000	868.6000		
C2M3			820.4000	830.5000	880.2000		
C2M4			690.5000	770.4000	919.8000		
C3M1			710.1000	840.8000	980.0000		
C3M2			650.6000	852.3500	895.3000		
C3M3			645.2000	815.4000	915.3000		
C3M4			645.2000	815.4000	915.3000		

976

977

Table A.13 Posthoc test for L4

(I) CiMj	(J) CiMj	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
C1M0	C2M0	44.16667	27.49341	.957	-49.6143	137.9476
	C3M0	54.89000	27.49341	.800	-38.8909	148.6709
	C1M1	10.98000	27.49341	1.000	-82.8009	104.7609
	C1M2	-146.94667*	27.49341	.000	-240.7276	-53.1657
	C1M3	-105.39333*	27.49341	.012	-199.1743	-11.6124
	C1M4	-96.70000*	27.49341	.036	-190.4809	-2.9191
	C2M1	-20.15667	27.49341	1.000	-113.9376	73.6243
	C2M2	-127.04667*	27.49341	.000	-220.8276	-33.2657

C2M0	C2M3	-128.27333*	27.49341	.000	-222.0543	-34.4924
	C2M4	-48.73333	27.49341	.908	-142.5143	45.0476
	C3M1	-29.35333	27.49341	.999	-123.1343	64.4276
	C3M2	-122.53000*	27.49341	.001	-216.3109	-28.7491
	C3M3	-95.99000*	27.49341	.039	-189.7709	-2.2091
	C3M4	-125.26667*	27.49341	.001	-219.0476	-31.4857
	C1M0	-44.16667	27.49341	.957	-137.9476	49.6143
	C3M0	10.72333	27.49341	1.000	-83.0576	104.5043
	C1M1	-33.18667	27.49341	.997	-126.9676	60.5943
	C1M2	-191.11333*	27.49341	.000	-284.8943	-97.3324
	C1M3	-149.56000*	27.49341	.000	-243.3409	-55.7791
	C1M4	-140.86667*	27.49341	.000	-234.6476	-47.0857
	C2M1	-64.32333	27.49341	.561	-158.1043	29.4576
	C2M2	-171.21333*	27.49341	.000	-264.9943	-77.4324
	C2M3	-172.44000*	27.49341	.000	-266.2209	-78.6591
C3M0	C2M4	-92.90000	27.49341	.055	-186.6809	.8809
	C3M1	-73.52000	27.49341	.324	-167.3009	20.2609
	C3M2	-166.69667*	27.49341	.000	-260.4776	-72.9157
	C3M3	-140.15667*	27.49341	.000	-233.9376	-46.3757
	C3M4	-169.43333*	27.49341	.000	-263.2143	-75.6524
	C1M0	-54.89000	27.49341	.800	-148.6709	38.8909
	C2M0	-10.72333	27.49341	1.000	-104.5043	83.0576
	C1M1	-43.91000	27.49341	.959	-137.6909	49.8709
	C1M2	-201.83667*	27.49341	.000	-295.6176	-108.0557
	C1M3	-160.28333*	27.49341	.000	-254.0643	-66.5024
	C1M4	-151.59000*	27.49341	.000	-245.3709	-57.8091
	C2M1	-75.04667	27.49341	.291	-168.8276	18.7343
	C2M2	-181.93667*	27.49341	.000	-275.7176	-88.1557
	C2M3	-183.16333*	27.49341	.000	-276.9443	-89.3824
	C2M4	-103.62333*	27.49341	.015	-197.4043	-9.8424
C1M1	C3M1	-84.24333	27.49341	.134	-178.0243	9.5376
	C3M2	-177.42000*	27.49341	.000	-271.2009	-83.6391
	C3M3	-150.88000*	27.49341	.000	-244.6609	-57.0991
	C3M4	-180.15667*	27.49341	.000	-273.9376	-86.3757
	C1M0	-10.98000	27.49341	1.000	-104.7609	82.8009
	C2M0	33.18667	27.49341	.997	-60.5943	126.9676
	C3M0	43.91000	27.49341	.959	-49.8709	137.6909
	C1M2	-157.92667*	27.49341	.000	-251.7076	-64.1457
	C1M3	-116.37333*	27.49341	.003	-210.1543	-22.5924
	C1M4	-107.68000*	27.49341	.009	-201.4609	-13.8991
	C2M1	-31.13667	27.49341	.998	-124.9176	62.6443
	C2M2	-138.02667*	27.49341	.000	-231.8076	-44.2457
	C2M3	-139.25333*	27.49341	.000	-233.0343	-45.4724
	C2M4	-59.71333	27.49341	.685	-153.4943	34.0676
	C3M1	-40.33333	27.49341	.980	-134.1143	53.4476
	C3M2	-133.51000*	27.49341	.000	-227.2909	-39.7291
C1M2	C3M3	-106.97000*	27.49341	.010	-200.7509	-13.1891
	C3M4	-136.24667*	27.49341	.000	-230.0276	-42.4657
	C1M0	146.94667*	27.49341	.000	53.1657	240.7276
	C2M0	191.11333*	27.49341	.000	97.3324	284.8943
	C3M0	201.83667*	27.49341	.000	108.0557	295.6176
	C1M1	157.92667*	27.49341	.000	64.1457	251.7076
	C1M3	41.55333	27.49341	.974	-52.2276	135.3343
	C1M4	50.24667	27.49341	.886	-43.5343	144.0276
	C2M1	126.79000*	27.49341	.001	33.0091	220.5709
	C2M2	19.90000	27.49341	1.000	-73.8809	113.6809
	C2M3	18.67333	27.49341	1.000	-75.1076	112.4543
	C2M4	98.21333*	27.49341	.030	4.4324	191.9943
	C3M1	117.59333*	27.49341	.002	23.8124	211.3743
	C3M2	24.41667	27.49341	1.000	-69.3643	118.1976
	C3M3	50.95667	27.49341	.874	-42.8243	144.7376
	C3M4	21.68000	27.49341	1.000	-72.1009	115.4609
C1M3	C1M0	105.39333*	27.49341	.012	11.6124	199.1743
	C2M0	149.56000*	27.49341	.000	55.7791	243.3409
	C3M0	160.28333*	27.49341	.000	66.5024	254.0643
	C1M1	116.37333*	27.49341	.003	22.5924	210.1543
	C1M2	-41.55333	27.49341	.974	-135.3343	52.2276
	C1M4	8.69333	27.49341	1.000	-85.0876	102.4743
	C2M1	85.23667	27.49341	.122	-8.5443	179.0176
	C2M2	-21.65333	27.49341	1.000	-115.4343	72.1276

C1M4	C2M3	-22.88000	27.49341	1.000	-116.6609	70.9009
	C2M4	56.66000	27.49341	.760	-37.1209	150.4409
	C3M1	76.04000	27.49341	.270	-17.7409	169.8209
	C3M2	-17.13667	27.49341	1.000	-110.9176	76.6443
	C3M3	9.40333	27.49341	1.000	-84.3776	103.1843
	C3M4	-19.87333	27.49341	1.000	-113.6543	73.9076
	C1M0	96.70000*	27.49341	.036	2.9191	190.4809
	C2M0	140.86667*	27.49341	.000	47.0857	234.6476
	C3M0	151.59000*	27.49341	.000	57.8091	245.3709
	C1M1	107.68000*	27.49341	.009	13.8991	201.4609
	C1M2	-50.24667	27.49341	.886	-144.0276	43.5343
	C1M3	-8.69333	27.49341	1.000	-102.4743	85.0876
	C2M1	76.54333	27.49341	.259	-17.2376	170.3243
	C2M2	-30.34667	27.49341	.999	-124.1276	63.4343
	C2M3	-31.57333	27.49341	.998	-125.3543	62.2076
C2M1	C2M4	47.96667	27.49341	.918	-45.8143	141.7476
	C3M1	67.34667	27.49341	.479	-26.4343	161.1276
	C3M2	-25.83000	27.49341	1.000	-119.6109	67.9509
	C3M3	.71000	27.49341	1.000	-93.0709	94.4909
	C3M4	-28.56667	27.49341	.999	-122.3476	65.2143
	C1M0	20.15667	27.49341	1.000	-73.6243	113.9376
	C2M0	64.32333	27.49341	.561	-29.4576	158.1043
	C3M0	75.04667	27.49341	.291	-18.7343	168.8276
	C1M1	31.13667	27.49341	.998	-62.6443	124.9176
	C1M2	-126.79000*	27.49341	.001	-220.5709	-33.0091
	C1M3	-85.23667	27.49341	.122	-179.0176	8.5443
	C1M4	-76.54333	27.49341	.259	-170.3243	17.2376
	C2M2	-106.89000*	27.49341	.010	-200.6709	-13.1091
	C2M3	-108.11667*	27.49341	.008	-201.8976	-14.3357
	C2M4	-28.57667	27.49341	.999	-122.3576	65.2043
C2M2	C3M1	-9.19667	27.49341	1.000	-102.9776	84.5843
	C3M2	-102.37333*	27.49341	.018	-196.1543	-8.5924
	C3M3	-75.83333	27.49341	.274	-169.6143	17.9476
	C3M4	-105.11000*	27.49341	.013	-198.8909	-11.3291
	C1M0	127.04667*	27.49341	.000	33.2657	220.8276
	C2M0	171.21333*	27.49341	.000	77.4324	264.9943
	C3M0	181.93667*	27.49341	.000	88.1557	275.7176
	C1M1	138.02667*	27.49341	.000	44.2457	231.8076
	C1M2	-19.90000	27.49341	1.000	-113.6809	73.8809
	C1M3	21.65333	27.49341	1.000	-72.1276	115.4343
	C1M4	30.34667	27.49341	.999	-63.4343	124.1276
	C2M1	106.89000*	27.49341	.010	13.1091	200.6709
	C2M3	-1.22667	27.49341	1.000	-95.0076	92.5543
	C2M4	78.31333	27.49341	.225	-15.4676	172.0943
	C3M1	97.69333*	27.49341	.032	3.9124	191.4743
	C3M2	4.51667	27.49341	1.000	-89.2643	98.2976
C2M3	C3M3	31.05667	27.49341	.999	-62.7243	124.8376
	C3M4	1.78000	27.49341	1.000	-92.0009	95.5609
	C1M0	128.27333*	27.49341	.000	34.4924	222.0543
	C2M0	172.44000*	27.49341	.000	78.6591	266.2209
	C3M0	183.16333*	27.49341	.000	89.3824	276.9443
	C1M1	139.25333*	27.49341	.000	45.4724	233.0343
	C1M2	-18.67333	27.49341	1.000	-112.4543	75.1076
	C1M3	22.88000	27.49341	1.000	-70.9009	116.6609
	C1M4	31.57333	27.49341	.998	-62.2076	125.3543
	C2M1	108.11667*	27.49341	.008	14.3357	201.8976
	C2M2	1.22667	27.49341	1.000	-92.5543	95.0076
	C2M4	79.54000	27.49341	.204	-14.2409	173.3209
	C3M1	98.92000*	27.49341	.027	5.1391	192.7009
	C3M2	5.74333	27.49341	1.000	-88.0376	99.5243
	C3M3	32.28333	27.49341	.998	-61.4976	126.0643
	C3M4	3.00667	27.49341	1.000	-90.7743	96.7876
C2M4	C1M0	48.73333	27.49341	.908	-45.0476	142.5143
	C2M0	92.90000	27.49341	.055	-.8809	186.6809
	C3M0	103.62333*	27.49341	.015	9.8424	197.4043
	C1M1	59.71333	27.49341	.685	-34.0676	153.4943
	C1M2	-98.21333*	27.49341	.030	-191.9943	-4.4324
	C1M3	-56.66000	27.49341	.760	-150.4409	37.1209
	C1M4	-47.96667	27.49341	.918	-141.7476	45.8143
	C2M1	28.57667	27.49341	.999	-65.2043	122.3576

C3M1	C2M2	-78.31333	27.49341	.225	-172.0943	15.4676
	C2M3	-79.54000	27.49341	.204	-173.3209	14.2409
	C3M1	19.38000	27.49341	1.000	-74.4009	113.1609
	C3M2	-73.79667	27.49341	.318	-167.5776	19.9843
	C3M3	-47.25667	27.49341	.926	-141.0376	46.5243
	C3M4	-76.53333	27.49341	.260	-170.3143	17.2476
	C1M0	29.35333	27.49341	.999	-64.4276	123.1343
	C2M0	73.52000	27.49341	.324	-20.2609	167.3009
	C3M0	84.24333	27.49341	.134	-9.5376	178.0243
	C1M1	40.33333	27.49341	.980	-53.4476	134.1143
	C1M2	-117.59333*	27.49341	.002	-211.3743	-23.8124
	C1M3	-76.04000	27.49341	.270	-169.8209	17.7409
	C1M4	-67.34667	27.49341	.479	-161.1276	26.4343
	C2M1	9.19667	27.49341	1.000	-84.5843	102.9776
	C2M2	-97.69333*	27.49341	.032	-191.4743	-3.9124
	C2M3	-98.92000*	27.49341	.027	-192.7009	-5.1391
	C2M4	-19.38000	27.49341	1.000	-113.1609	74.4009
	C3M2	-93.17667	27.49341	.053	-186.9576	.6043
	C3M3	-66.63667	27.49341	.498	-160.4176	27.1443
	C3M4	-95.91333*	27.49341	.039	-189.6943	-2.1324
C3M2	C1M0	122.53000*	27.49341	.001	28.7491	216.3109
	C2M0	166.69667*	27.49341	.000	72.9157	260.4776
	C3M0	177.42000*	27.49341	.000	83.6391	271.2009
	C1M1	133.51000*	27.49341	.000	39.7291	227.2909
	C1M2	-24.41667	27.49341	1.000	-118.1976	69.3643
	C1M3	17.13667	27.49341	1.000	-76.6443	110.9176
	C1M4	25.83000	27.49341	1.000	-67.9509	119.6109
	C2M1	102.37333*	27.49341	.018	8.5924	196.1543
	C2M2	-4.51667	27.49341	1.000	-98.2976	89.2643
	C2M3	-5.74333	27.49341	1.000	-99.5243	88.0376
	C2M4	73.79667	27.49341	.318	-19.9843	167.5776
	C3M1	93.17667	27.49341	.053	-.6043	186.9576
C3M3	C3M3	26.54000	27.49341	1.000	-67.2409	120.3209
	C3M4	-2.73667	27.49341	1.000	-96.5176	91.0443
	C1M0	95.99000*	27.49341	.039	2.2091	189.7709
	C2M0	140.15667*	27.49341	.000	46.3757	233.9376
	C3M0	150.88000*	27.49341	.000	57.0991	244.6609
	C1M1	106.97000*	27.49341	.010	13.1891	200.7509
	C1M2	-50.95667	27.49341	.874	-144.7376	42.8243
	C1M3	-9.40333	27.49341	1.000	-103.1843	84.3776
	C1M4	-.71000	27.49341	1.000	-94.4909	93.0709
	C2M1	75.83333	27.49341	.274	-17.9476	169.6143
	C2M2	-31.05667	27.49341	.999	-124.8376	62.7243
	C2M3	-32.28333	27.49341	.998	-126.0643	61.4976
C3M4	C2M4	47.25667	27.49341	.926	-46.5243	141.0376
	C3M1	66.63667	27.49341	.498	-27.1443	160.4176
	C3M2	-26.54000	27.49341	1.000	-120.3209	67.2409
	C3M4	-29.27667	27.49341	.999	-123.0576	64.5043
	C1M0	125.26667*	27.49341	.001	31.4857	219.0476
	C2M0	169.43333*	27.49341	.000	75.6524	263.2143
	C3M0	180.15667*	27.49341	.000	86.3757	273.9376
	C1M1	136.24667*	27.49341	.000	42.4657	230.0276
	C1M2	-21.68000	27.49341	1.000	-115.4609	72.1009
	C1M3	19.87333	27.49341	1.000	-73.9076	113.6543
	C1M4	28.56667	27.49341	.999	-65.2143	122.3476
	C2M1	105.11000*	27.49341	.013	11.3291	198.8909
	C2M2	-1.78000	27.49341	1.000	-95.5609	92.0009
	C2M3	-3.00667	27.49341	1.000	-96.7876	90.7743
	C2M4	76.53333	27.49341	.260	-17.2476	170.3143
	C3M1	95.91333*	27.49341	.039	2.1324	189.6943
	C3M2	2.73667	27.49341	1.000	-91.0443	96.5176
	C3M3	29.27667	27.49341	.999	-64.5043	123.0576

*, The mean difference is significant at the 0.05 level.