# GPU Accelerated Genetic Algorithm with Sequence-based Clustering for Ordered Problems

# GPU Accelerated Genetic Algorithm with Sequence-based Clustering for Ordered Problems

Ryoma Ohira
*School of Information and Communication Technology*
*Griffith University*
Gold Coast, Australia
r.ohira@griffith.edu.au

Md. Saiful Islam
*School of Information and Communication Technology*
*Griffith University*
Gold Coast, Australia
saiful.islam@griffith.edu.au

***Abstract*—The island model allows genetic algorithms to effectively maintain diversity through migration between multiple independent populations. Due to its flexibility and modularity, it is commonly employed in distributed and parallel implementations, particularly in recent trends in leveraging the massively parallel cores in GPUs. However, the efficiency and effectiveness of the island model can be considered as its ability to manage its global and local search while minimising the overlap of islands searching in the same area of the solution space. This paper introduces a GPU accelerated island-model genetic algorithm that conducts global search by organising its populations into islands according to the similarity in genotype sequences. Local search is managed through adaptive mechanisms designed to maintain population diversity. The characteristics of the proposed genetic algorithm are investigated with encouraging results demonstrating its robustness and scalability when solving ordered optimisation problems.**

***Index Terms*—genetic algorithm, ordered problems, adaptive optimisation, GPU, island model genetic algorithm**

## I. Introduction

Optimisation can be considered as finding the best solutions for a given problem with a possible solution space. Since originally proposed by Holland [1] and DeJong [2], genetic algorithms (GAs) have demonstrated their ability as metaheuristic algorithms in obtaining near-optimal solutions to solving NP-hard problems within a finite time.

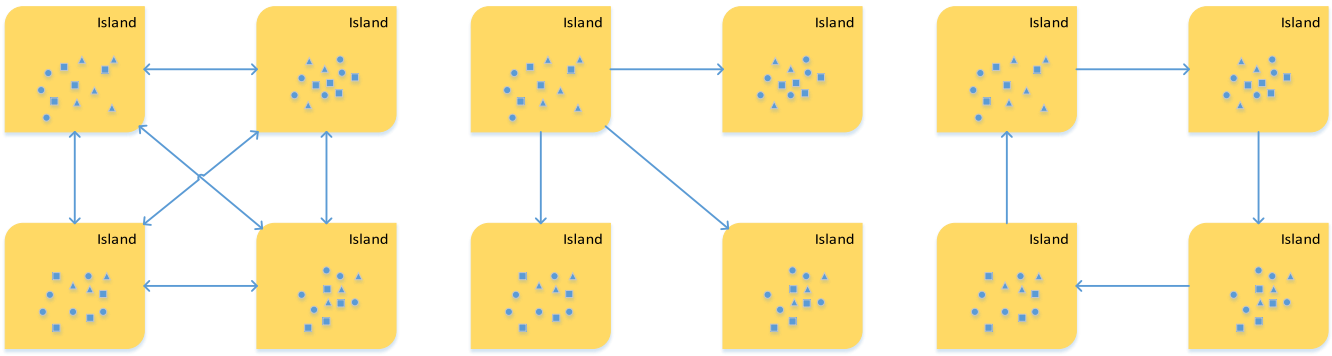
Recent trends in parallel implementations of GAs have demonstrated the scalability and flexibility of the island-model (IMGA) [3], [4] and its capabilities in effectively exploring the search space while delaying convergence. This method involves distributing a large global population to islands where each island's population undergoes the evolutionary process. Good performing individuals are selected to migrate between islands in order to maintain diversity and prevent premature convergence. While this method is an effective mechanism for preventing global convergence, its effectiveness is dependent on the number of islands. With the ability for an IMGA to maintain population diversity being linked to the number of islands it manages, it can become a computationally intensive process. As such, an effective and efficient IMGA must maintain a balance between the number of islands and the computational power required to process them.

Recent trends have focused on parallel implementations of GAs on both CPU and, more recently, the graphics processing unit (GPU) in order to accelerate GAs. The Compute Unified Device Architecture (CUDA) [5] toolkit released by NVIDIA allows for GPUs to be used for general purposes. With a single modern GPU containing thousands of cores, GPU processing has allowed for high-performance computation in many scientific applications. When used to accelerate genetic algorithms, GPUs are often used to implement master-slave, IMGA and hybrid configurations [3]. In a master-slave configuration, a host (CPU) manages the population through selection, crossover, and mutation while the GPU handles the more computationally expensive processes such as fitness and diversity evaluations.

While the massively parallel nature of GPU processing makes the platform desirable for deploying an IMGA, recent trends have identified several challenges in fully deploying an IMGA onto GPU architecture. These include the nature of the single instruction multiple threads (SIMT) architecture, the amount of shared memory available and the latency between global and shared memory. Furthermore, when considering the massive number of islands that a GPU can handle in an IMGA, there is a higher likelihood of two or more islands searching in overlapping areas of the solution space. In order to improve the effectiveness and efficiency of an IMGA, it is important to minimise the way in which these islands overlap by managing the search at a global and local level.

In this paper, we present an IMGA completely deployed on the GPU that manages many islands in a parallel manner. To improve the efficiency and effectiveness of the IMGA's search, we combine adaptive diversity maintenance mechanisms and spectral clustering in order to prevent islands from searching in the same areas as other islands. This allows for the proposed IMGA to direct the global search of the solution space to better navigate the complexities of the fitness landscape.

The remainder of this paper is organised as follows: Section II highlights the existing works in the literature for GPU accelerated IMGAs with Section III discussing the details of the IMGA and GPU architecture. In Section IV, we outline the proposed IMGA and its characteristics. Section V contains the experimental environments and results analysis for instances from the travelling salesman problem and the capacitated vehicle routing problem. Finally, a discussion and conclusions are drawn in Section VI.




(a) Fully-connected (global) migration (b) Stepping-stone migration (c) Ring migration

Fig. 1. Examples of island model topologies with each method of migration contributing to different characteristics of an IMGA

## II. Related Work

A GA's ability to maintain diversity is critical to its capabilities in preventing premature convergence. In an IMGA, diversity is maintained through migration between populations. However, the method of migration has significant effects on the GA's search capacity as well as its ability to maintain diversity. While many methods for distributing a GA for parallel computation have been proposed, the IMGA is one of the most commonly used techniques due to its performance and flexibility [3].

Another active research area is that of adaptive genetic algorithms. These aim to introduce mechanisms that monitor and maintain the diversity of a single population [6]. This includes adaptively changing operators [7] or their parameters [8]. Other methods include creating sub-populations to maintain a balance between exploration and exploitation according to the state of the population [9] or creating candidate individuals that introduces population diversity while improving its fitness [10]. These works have found effective ways of managing a single population's diversity and search in order to both prevent premature convergence and improve the GA's search capabilities. While these adaptive features are not often used in maintaining population diversity in distributed GA models, some of these mechanisms have been introduced to migration strategies.

Studies into understanding the characteristics of the fitness landscape of combinatorial optimisation problems, like the Travelling Salesman Problem (TSP), is an active and on-going research topic [11]. Where it was previously theorised as a hill-valley landscape, recent works highlighted how communities of local optima can be found as clusters in the solution space as funnels [12], [13]. Further works [14], [15] identified how the number of clusters and the size of the cluster that the global optimum resides in correlates to the search difficulty. By better understanding the relationship between the many local optima and the fitness landscape, more efficient and effective methods of searching the solution space can be developed. This has led to recent works incorporating clustering techniques into GAs in numerous ways. These include using clustering to measure the diversity of the population [16], part of the evolution process [17] or during migration for IMGAs [18], [19].

Where IMGAs distribute sub-populations around the global solution space, ensuring that the islands are not searching the same areas is important to the IMGA's effectiveness. By clustering similar individuals together, an IMGA can encourage each island to conduct local search around an optima but it is necessary to ensure that the sub-population itself does not converge prematurely. By combining techniques from adaptive GAs and clustering based migration strategies, each island will be able to maintain a healthy level of population diversity while allowing the IMGA to help direct each island's search in order to minimise overlap in the solution space.

## III. Preliminaries

One of the open research problems in the topic of GPU accelerated GAs is the challenge of deploying an IMGA completely onto the GPU. In this section, we discuss the technical challenges and compromises in designing IMGAs for GPU deployment.

### A. Island Model based Genetic Algorithms

Island model based genetic algorithms (IMGAs) are a popular and efficient model for implementing GAs in a parallel manner [3], [4], [20]. The main premise of an IMGA is to parallelise the search process by managing multiple islands of populations that are run concurrently and independently to one another. In order to maintain diversity, individuals are allowed to migrate from one island to another. The main migration methods are demonstrated in Fig. 1. Whitley et al [20] demonstrated how a global migration method (Fig. 1a) results in the IMGA acting as a single global population. This can result in an overall high level of genetic similarity between populations. A stepping-stone migration method (Fig. 1b) allows for individuals to migrate to the next island only [21]. This reduces the rate of convergence and allows islands further away from the globally fittest individuals to explore more of the solution space while introducing diversity to

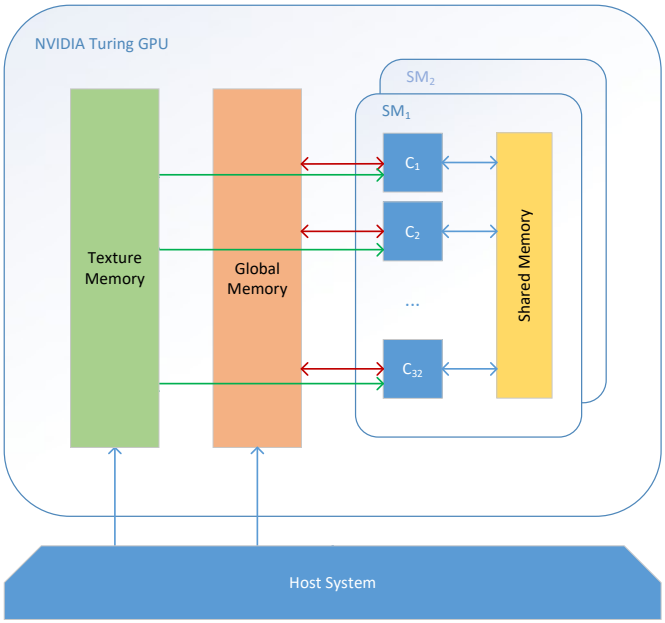



Fig. 2. NVIDIA Turing memory hierarchy

nearby islands. Ursem [22] proposed a novel migration policy which focused on classifying individuals into islands using a hill-valley detection algorithm. While this method is successful in maintaining different islands as they merged, it is inefficient and the moving speed of islands is generally slow. However, its important contribution to the topic was how the relationship between individuals and the fitness landscape should be taken into consideration when deciding which island an individual should belong to.

There are several challenges in developing an IMGA for the GPU. Many approaches implement a hybrid Master-Slave IMGA where islands are managed on a master node (CPU) with slave nodes (GPU) being responsible for introducing parallel processing for computationally expensive tasks such as fitness calculations [4]. However, the communication between CPU and GPU can negatively affect its performance due to the latency in synchronising the memory between the two.

By deploying each island as a thread-block on a CUDA based GPU, Melab et al [23] demonstrated how an IMGA can be deployed on the GPU. By storing the local population on the shared memory of the thread block, the authors were able to minimise the communication cost between the shared and global memory. However, due to the GPU architecture, several limitations must be considered when deploying the island model onto the GPU.

### B. GPU Architecture

The Compute Unified Device Architecture (CUDA) toolkit developed for GPUs have enabled software to make use of General Purpose GPUs (GPGPUs) to massively parallelise computing tasks. NVIDIA introduced some changes with their Turing architecture. Turing based GPUs consists of Graphics Processing Clusters (GPCs) that hosts six Texture Processing clusters (TPCs). Unlike the previous architectures, each TPC contains two Streaming Multiprocessors (SMs) that each contain four warps. Each SM contains 64KB of shared memory capacity.

One of the challenges involved with deploying an IMGA to the GPU is the constraints on the shared memory [24]. While subsequent generations has provided greater flexibility in accessing and managing the amount of memory available, NVIDIA's architecture has largely limited the amount of shared memory available to 48-64KB while global memory has increased at a steady rate. This is important to the performance of an GPU based IMGA due to the cost in communicating between the cores and the different levels of memory available. While the global memory available to the GPU can range between 6GB to 24GB, accessing global memory can take between 200 and 600 memory clock cycles [5]. However, local and shared memory can be accessed much faster for 2-8 clock cycles. As over-utilisation of the global memory can result in significant costs to computation time, data should be partitioned or encoded to fit inside the shared-memory. However, due to the complexity and the size of larger combinatorial optimisation problems, it is an ongoing and active research topic [23]–[26].

Unlike multi-tasking on the CPU, the GPU architecture is implemented as single instruction multiple threads (SIMT) where the GPU is only able to run a single task across multiple threads at any given time. An IMGA deployed on the GPU should aim to minimise the thread divergence by ensuring all threads within a warp follow the same execution path.

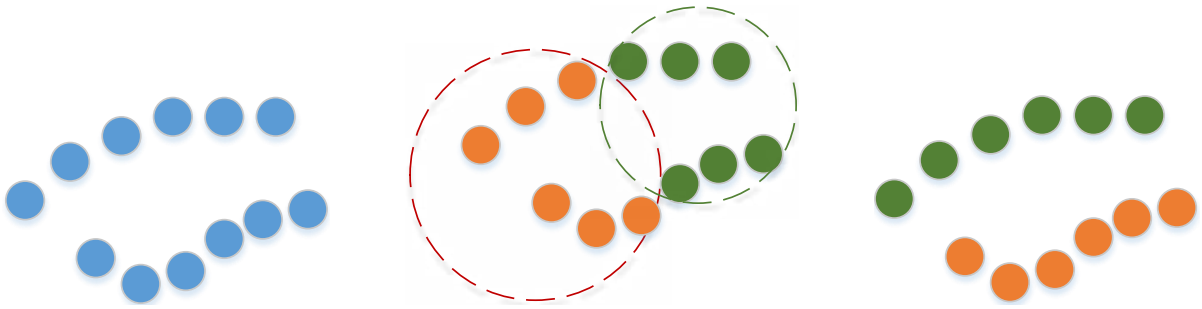
These technical constraints contributes to the challenge of fully deploying an IMGA onto the GPU in an optimal manner.

## IV. Our Approach

In our proposed approach, each island is responsible for conducting local search. This is achieved by migrating similar individuals to the same island. This allows for each island to focus on searching within a specific area of the solution space. Global search is conducted by managing the islands to ensure that they do not overlap by encouraging them to search in different areas of the solution space. Inspired by the Galapagos Islands, this creates a series of islands where the populations evolve to become highly niche and adapted to its local solution space. By distributing this process across the many cores available on GPUs, the Galapagos Island Model Genetic Algorithm (GIMGA) is designed to manage and direct the search of its islands. This allows GIMGA to minimise the overlaps in its search efforts thus improving its efficiency and effectiveness in searching large solution spaces.

### A. Sequence-wise Similarity

In order to adaptively maintain population diversity, it is necessary for a GA to measure and monitor the similarity of genotypes. However, there are a number of ways to measure the similarity or difference between individuals. Many adaptive GAs use euclidean or Hamming distances [6]. When considered the sequential nature of ordered problems, sequence-wise



(a) 2D representation of individuals in the solution space

(b) $k$-means clustering

(c) Spectral clustering

Fig. 3. Comparison between $k$-means and spectral clustering of individuals where $k = 2$

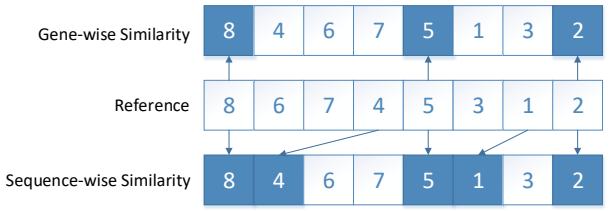



Fig. 4. Demonstration of a gene-wise and sequence-wise approach to measuring genotypic similarity

measurements have demonstrated their suitability and effectiveness for the TSP and VRP [27]. Similar to the Hamming distance, the longest common subsequence (LCS) distance is the number of nodes that separate the distance between the sequences of two genotypes. This is demonstrated in Fig. 4 where the gene-wise approach considers two genes to be similar if they share the same gene values at the same positions in the genotype sequence. Meanwhile, a sequence-wise approach considers the similarity in gene-sequence. When considering that a genotype represents a single or a set of routes in the TSP and CVRP, the sequence of nodes contributes to the fitness of the genotype more so than the positions of each node. For this reason, GIMGA uses the LCS length of genotypes to determine the diversity and similarity of individuals and islands.

### *B. Sequence Based Clustering*

In IMGAs, good solutions from an island are propagated to other islands through migration. There are numerous migration strategies, each with their unique characteristics. A global migration strategy allows for the best individual from the global population to migrate to all islands. Whitley et al [20] demonstrated how this strategy results in IMGAs with similar characteristics to a single, global population. A stepping-stone strategy allows the best individual from an island to only migrate to an adjacent island [4]. This slows the propagation of the best individuals and reducing their dominance on the global population. Variants of this strategy include the ring and star migration strategies.

By reducing the dominance of strong individuals over the global population, IMGAs are better able to explore the solution space through its numerous islands. When islands begin to converge on the same solutions, the search space of each island begin to overlap, reducing the IMGA's ability to effectively conduct global search of the solution space. In order to improve the IMGA's capabilities in maintaining a balance between exploration (global search) and exploitation (local search), our proposed IMGA uses spectral clustering to organise the global population into islands according to their genotypic similarity.

As GIMGA evolves across generations, each island will conduct its own exploration and exploitation resulting in a sub-population with a healthy degree of diversity while focused on an area of the solution space. The global population is clustered into $k$ sub-populations according to their sequence-wise genotypic similarity. Spectral clustering was selected due to it ability for parallel processing on the GPU [28].

Given a global population of $p$, in order to group the population into $k$ islands using spectral clustering, we must first construct an affinity matrix. With a sequence-based approach, we can define the affinity matrix as shown in Eq. 1.

$$A_{ij} = \begin{cases} 1 & \text{if } i = j \\ \frac{LCS(i,j)}{N} & \text{if } i \neq j \end{cases} \tag{1}$$

The affinity matrix ($A$) is created using the LCS length between each genotypes $i$ and $j$. This LCS length is normalised against the length of the genotype ($N$). $A_{ij} \simeq 1$ indicates that the two are similar while $A_{ij} \to 0$ means that the two genotypes are far apart. The diagonal matrix ($D$) and Lapalacian matrix ($L$) are defined in Eq. 2 and Eq. 3 below.

$$D_{ij} = \sum_{j} A_{ij} \tag{2}$$

$$L_{ij} = Dij - Aij \tag{3}$$

The first $k$ smallest eigenvalues are calculated and mapped to clusters. This $k$ value is dependent on the number of islands or thread blocks available on the GPU. This spectral clustering migration process can be seen in Fig. 5. During the evolution

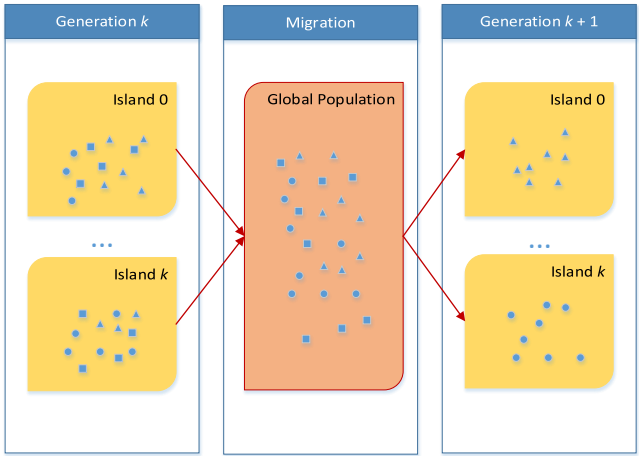



Fig. 5. Spectral clustering migration

stages, each island acts as an adaptive genetic algorithm that is responsible for maintaining its own population diversity. From adaptively balancing its focus between exploration and exploitation, each island will have contributed to local search and global search. These individuals are then stored in global memory for clustering into new islands.

### C. Adaptive Islands

Each island in the IMGA is capable of acting as an independent GA that is responsible for maintaining its balance between exploration and exploitation. The architecture for balancing exploration and exploitation sub-populations within an island is demonstrated in Fig 6. Each individual is processed on a CUDA core $(C_1, \ldots, C_n)$ according to the number of cores available on the GPU device. During the diversity evaluation stage of the evolution process, the island measures the LCS distance $(N - LCS(i, j))$ of its population in order to determine its diversity. By measuring the diversity between the fittest individual and the rest of the island population, GIMGA can effectively monitor its population diversity in order to manage its exploration and exploitation island sub-populations. While measuring the difference between all individuals provides a much thorough measure of population diversity, previous work has demonstrated how measuring the difference between a population and its fittest individual can provide a relatively accurate measurement at a much lower computational cost [29]. The exploitation sub-population selects two parents from the shared memory to conduct crossover, and mutation. Individuals in the exploration sub-population selects a single parent and undergoes mutation at a rate dependent on the degree of diversity of the population. The coefficient of variation in LCS distances between individuals is used in the population diversity ratio of the population. This is then averaged with the individual's fitness contribution to establish a mutation value that reflects the individual's contribution to the population's fitness and diversity [27].

By combining the spectral clustering with adaptive parameter controls, GIMGA aims to allow each island to actively maintain its own population diversity and focus on different areas of the solution space. By distributing the islands in such a manner, GIMGA aims to improve on the ability to search for good solutions in an effective and efficient manner.

## V. Experiments

All algorithms were implemented and tested on an AMD Ryzen 5 2600x CPU with 32GB of main memory running Windows 10. The GPU implementations were run using an NVIDIA GTX 2080. Each implementation was written in C++ with the CUDA 10.1 toolkit. For benchmark purposes, the CPU-based IMGA was run on a single core.

In order to test GIMGA's ability to solve ordered problems, a number of benchmark instances were selected from the TSP and CVRP benchmark libraries. The TSP is a well known combinatorial optimisation problem where the sequence of cities a salesperson is to travel must be optimised to reduce the total distance travelled. The CVRP is a generalised TSP where $N$ parcels with known weights must be made using $K$ vehicles with known capacities. Both are ordered problems where each node can only be visited once. The instances were selected for their variety in dimensions that allow for testing the robustness of the GAs. Results presented in this study represent the average values over 50 independent runs and 95% confidence intervals. In order to determine the statistical significance of the results, two-sample Z-tests were conducted between the GIMGA results and those of the IMGA and a parallel implementation of the LCSB-AGA (P-LCSB-AGA).

For all GAs, the Modified Ordered Crossover (MOX) operator and Partially Shuffled Mutation (PSM) operator were implemented. These were selected due to their performance and adherence to the constraints of ordered problems as reported in [30].

### A. Efficiency

Fig. 7 compares the execution time in seconds of a single CPU core against the NVIDIA GTX 2080 across different number of islands and their sizes. This speedup is measured as the time required to complete 1,000 generations of the problem instance and represents the average times after 50 runs. Both CPU and GPU implementations run the same GIMGA framework to ensure an accurate comparison. As expected, as the global population increases in size, a single CPU core takes longer to compute each generation. However, the GPU is much more capable of processing the increasing number of islands and population size. In Table I, we present the relationship between the number of islands, computation time and solution quality between a single CPU core and the multi-core GPU in running GIMGA. The island population has been kept at a constant size of 128. As the number of islands increases, the solution quality for both CPU and GPU implementations improve. However, while there is minimal increase in computation time for the GPU implementation, a single-core CPU experiences a significant increase in computation time.

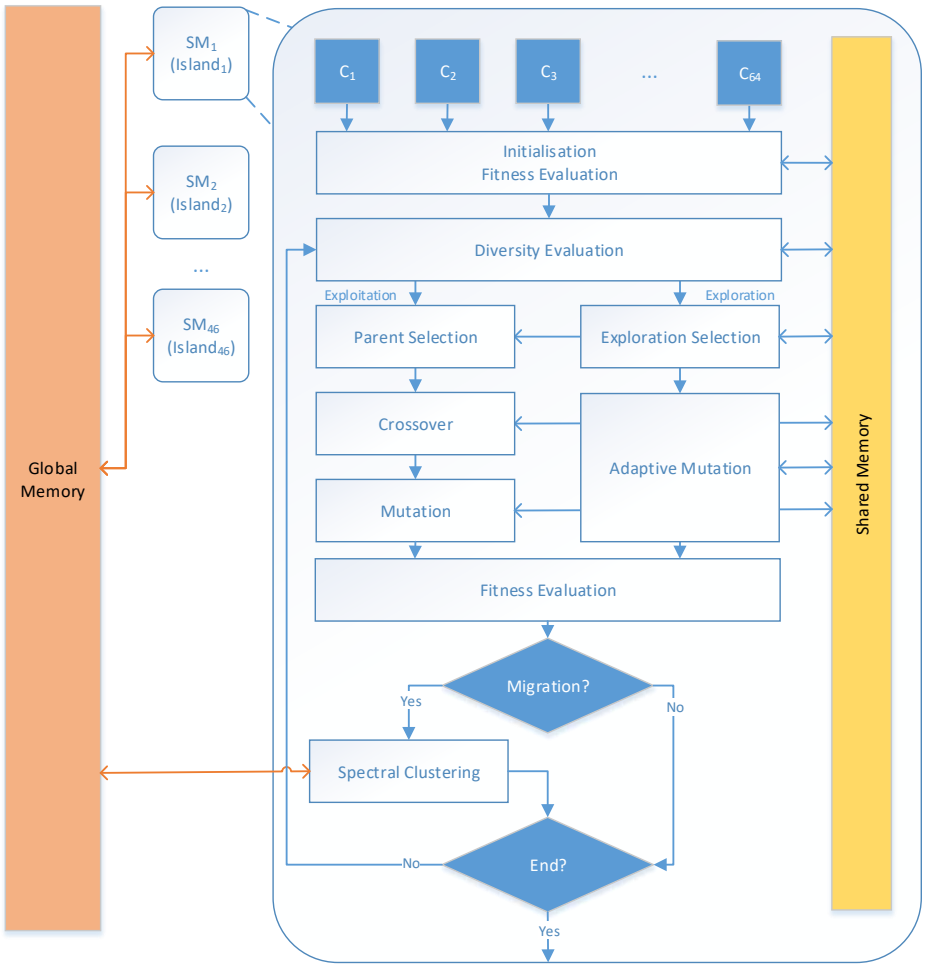



Fig. 6. Adaptive islands architecture

TABLE I
GIMGA SPEEDUP FOR F-N135-K7

| Islands | CPU | | GPU | | Speed Up |
|---|---|---|---|---|---|
| | Qlty. | Time | Qlty. | Time | |
| 32 | 1,535 | 350 | 1,534 | 8 | 46x |
| 48 | 1,488 | 1,174 | 1,486 | 8 | 152x |
| 64 | 1,495 | 2,735 | 1,493 | 8 | 339x |
| 80 | 1,462 | 5,270 | 1,363 | 8 | 631x |
| 96 | 1,361 | 9,155 | 1,370 | 9 | 1,059x |
| 112 | 1,354 | 14,367 | 1,340 | 9 | 1,589x |
| 128 | 1,325 | 15,361 | 1,298 | 9 | 1,640x |

TABLE II
DIVERSITY MAINTENANCE FOR F-N135-K7

| Islands | IMGA | | P-LCSB-AGA | | GIMGA | |
|---|---|---|---|---|---|---|
| | Qlty. | Div. | Qlty. | Div. | Qlty. | Div. |
| 32 | 2,167 | 0.12 | 1,652 | 0.77 | 1,533 | 0.78 |
| 48 | 2,153 | 0.10 | 1,567 | 0.71 | 1,486 | 0.76 |
| 64 | 2,132 | 0.11 | 1,532 | 0.74 | 1,493 | 0.77 |
| 80 | 2,142 | 0.10 | 1,482 | 0.71 | 1,363 | 0.79 |
| 96 | 2,113 | 0.12 | 1,460 | 0.67 | 1,369 | 0.78 |
| 112 | 2,105 | 0.09 | 1,455 | 0.65 | 1,360 | 0.73 |
| 128 | 2,023 | 0.11 | 1,412 | 0.68 | 1,298 | 0.79 |

TABLE III
BENCHMARK RESULTS

| Instance | Optimum | IMGA | | P-LCSB-AGA | | GIMGA | |
|---|---|---|---|---|---|---|---|
| | | Qlty. | Div. | Qlty. | Div. | Qlty. | Div. |
| Berlin52 | 7,542 | 7,583 | 0.25 | 7,542 | 0.55 | 7,542 | 0.65 |
| Pr107 | 44,303 | 51,524 | 0.20 | 44,303 | 0.63 | 44,303 | 0.76 |
| Pr152 | 73,682 | 103,841 | 0.12 | 73,770 | 0.64 | 73,720 | 0.78 |
| Rat195 | 2,323 | 6,894 | 0.14 | 2,897 | 0.63 | 2,564 | 0.79 |
| Pr264 | 49,135 | 65,348 | 0.15 | 61,441 | 0.67 | 57,123 | 0.81 |
| A-n32-k5 | 784 | 785 | 0.20 | 784 | 0.52 | 784 | 0.65 |
| A-n65-k9 | 1,174 | 1,181 | 0.10 | 1,174 | 0.58 | 1,174 | 0.73 |
| A-n80-k10 | 1,763 | 2,551 | 0.07 | 1,773 | 0.60 | 1,770 | 0.75 |
| P-n101-k4 | 681 | 1,198 | 0.08 | 821 | 0.61 | 763 | 0.76 |
| F-n135-k7 | 1,162 | 2,023 | 0.07 | 1,370 | 0.61 | 1,298 | 0.76 |

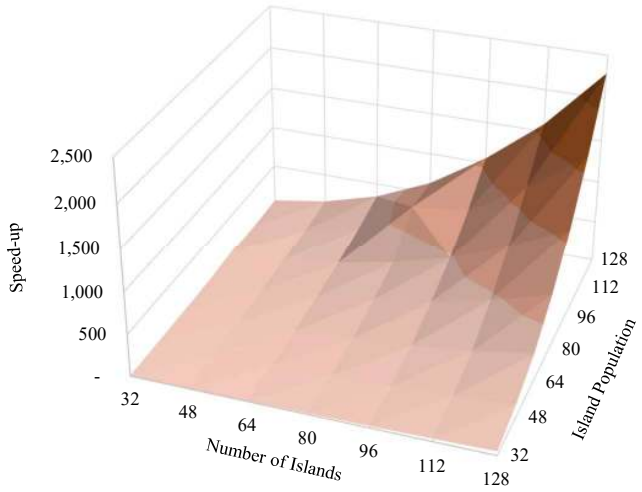



Fig. 7. Speed-up of GPU implementation of GIMGA for CVRP instance F-n135-k7

TABLE IV
STATISTICAL SIGNIFICANCE (Z-TESTS)

| Problem Instance | IMGA | P-LCSB-AGA |
|---|---|---|
| Berlin52 | ++ | * |
| Pr107 | ++ | * |
| Pr152 | ++ | + |
| Rat195 | ++ | ++ |
| Pr264 | ++ | ++ |
| A-n32-k5 | * | * |
| A-n65-k9 | ++ | * |
| A-n80-k10 | ++ | ++ |
| P-n101-k4 | ++ | ++ |
| F-n135-k7 | ++ | ++ |

### B. Diversity Maintenance

While speed-up is often considered as the main component of scalability, it is important to also consider the ability of an IMGA to maintain population diversity as the number of islands increases. In Table II, we measure the average quality of the solutions found as well as the average global population diversity for each IMGA across a range in number of islands after 1,000 generations. In order to measure the global population diversity, we measure the LCS distance between the fittest individuals of each island. This distance is then normalised against the problem's dimensions ($N$) where islands that are highly converged will approach a diversity value of 0 and highly diverse islands will result in a diversity value reaching 1.

The benchmark IMGA with a simple global migration strategy manages to find good quality solutions for the F-N135-K7 instance of the CVRP. However, it struggles to maintain a healthy level of sequence-wise diversity. This can be seen in the way that the IMGA maintains a low level of island diversity even when the number of islands is increased. This suggests that as the IMGA evolves and migrates, the islands begin to converge on the same solutions with the individuals of each island also becoming similar to one another.

On the other hand, P-LCSB-AGA manages to maintain a high level of population diversity with a smaller number of islands but this begins to diminish as more islands are introduced. The manner in which the quality of the solution found increases with the increased number of islands suggests that a number of islands begin to overlap with one another while each island attempts to maintain its adaptive diversity maintenance. This highlights the strengths of introducing adaptive mechanisms to IMGAs as well as how islands can become redundant if there is no oversight into their global search patterns.

The GIMGA configuration incorporates both adaptive mechanisms as well as spectral clustering in order to direct the search of each island into different areas of the solution space. By clustering individuals into islands according to their similarity, each island is influenced by the highest performing individuals of its cluster. As the island balances its exploration and exploitation, these influential individuals are likely to direct the local search while maintaining enough diversity to prevent convergence. The relatively high and stable degree of population diversity indicates that GIMGA is successful in directing the global search of its islands even as the number of islands increases. This can then be seen to contribute to its ability to find better quality solutions.

### C. Effectiveness

Where the efficiency and diversity maintenance demonstrate the IMGAs' scalability and capacity to search broadly through the solution space, these ultimately contribute to how effective the IMGAs are in terms of finding good quality solutions within a finite amount of time. Table III presents each IMGA's effectiveness in searching for the global optimum. The quality (Qlty.) and diversity (Div.) are average values from the 50 independent runs for each IMGA configuration. Each IMGA was set to 128 islands with 64 individuals per island for 1000 generations or until the known optimum was found.

The results indicate how each IMGA performs as the problem's dimensions increases. Given the fixed population and island parameters, the IMGA manages to achieve high quality solutions but with greater errors as the problem's dimensions increases. Furthermore, with no diversity maintenance, the degree of diversity decreases with the increase in dimensions. This suggests that there is a higher likelihood of convergence as the problem's complexity increases due to the clustered proximity of the local optima. As the IMGA evolves, the islands begin to converge towards the local optima.

When diversity mechanisms are introduced, P-LCSB-AGA manages to maintain a higher level of population diversity. In fact, as the problem dimensions increase, P-LCSB-AGA manages to slightly increase the amount of diversity between the islands. This indicates that diversity mechanisms are working as designed. However, when considering the performance of GIMGA with its spectral clustering migration strategy, it appears as though P-LCSB-AGA still is not able to sufficiently search the solution space in its attempts at global search despite its islands having significantly less overlap than the basic IMGA. GIMGA improves on this by being more

effective in its global search as indicated by its higher level of island diversity. By maintaining more diverse islands, this indicates that each island is less likely to be searching in the same areas as another island and thus giving more opportunity for the GA to find other clusters of local optima.

While these results demonstrate what GIMGA is capable of achieving, the two-sample z-tests shown in Table IV highlight their significance. In the z-test results, we compare the GIMGA with the IMGA and P-LCSB-AGA with a significant and a very significant improvement in performance is indicated with a "$+$" and "$++$" respectively. A "$*$" indicates that there is no significant difference in results between the IMGAs. In small dimension problems, all three IMGAs perform very well and are able to consistently find the optimal solution. However, as the problem size increases, GIMGA quickly gains significant improvements in solution quality over bot IMGA and P-LCSB-AGA.

## VI. Conclusion

In this paper, we propose an IMGA that focuses on maintaining unique islands of solutions that are capable of searching independently to one another. Unlike previous works, we introduce adaptive diversity maintenance mechanisms to each island population. Combining sequence-based adaptive diversity maintenance with spectral clustering, GIMGA is able to distribute its local search efforts across the global solution space with minimal overlap. Furthermore, by distributing the computation across the massively parallel NVIDIA GTX 2080, we demonstrate how the solution and its speedup can provide significant improvements to GIMGA's ability to find good quality solutions without significantly increasing the search duration.

Our findings suggest that when considering significantly large number of islands, IMGAs need to be designed with awareness of inefficiencies can exist when islands begin to overlap in their search. By reducing the likelihood of multiple islands searching in the same area, islands can be directed to search in new areas of the solution space, thus improving the effectiveness of the search. However, while this approach improves the broadness of the IMGA's search capabilities, the overlapping islands can be considered as a means to improve the depth of local search in an area dense with local optima. Future research in balancing global and local search should consider the optimal balance between the breadth of global search and the depth of local search between islands, ideally in an adaptive and online manner.

## References


[1] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[2] K. A. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," 1975.

[3] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.-J. Li, "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," *Applied Soft Computing*, vol. 34, pp. 286–300, 2015.

[4] J. R. Cheng and M. Gen, "Accelerating genetic algorithms with gpu computing: A selective overview," *Computers & Industrial Engineering*, vol. 128, pp. 514–525, 2019.

[5] "Nvidia turing gpu architecture: Graphics reinvented," "NVIDIA", Tech. Rep. WP-09183-001-v01, 2018.

[6] A. Aleti and I. Moser, "A systematic literature review of adaptive parameter control methods for evolutionary algorithms," *ACM Computing Surveys*, vol. 49, no. 3, pp. 1–35, 2016.

[7] A. F. Cruz-Salinas and J. G. Perdomo, "Self-adaptation of genetic operators through genetic programming techniques," in *GECCO*. ACM, 2017, pp. 913–920.

[8] F. Vafaee and P. C. Nelson, "An explorative and exploitative mutation scheme," in *CEC*. IEEE, 2010, pp. 1–8.

[9] B. Mc Ginley, J. Maher, C. O'Riordan, and F. Morgan, "Maintaining healthy population diversity using adaptive crossover, mutation, and selection," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 692–714, 2011.

[10] H. Zhang, Y. Liu, and J. Zhou, "Balanced-evolution genetic algorithm for combinatorial optimization problems: the general outline and implementation of balanced-evolution strategy based on linear diversity index," *Natural Computing*, vol. 17, no. 3, pp. 611–639, 2018.

[11] G. Ochoa and N. Veerapen, "Mapping the global structure of tsp fitness landscapes," *Journal of Heuristics*, vol. 24, no. 3, pp. 265–294, 2018.

[12] G. Ochoa, N. Veerapen, D. Whitley, and E. K. Burke, "The multi-funnel structure of tsp fitness landscapes: a visual exploration," in *Evolution Artificielle*. Springer, 2015, pp. 1–13.

[13] G. Ochoa and N. Veerapen, "Deconstructing the big valley search space hypothesis," in *Evolutionary Computation in Combinatorial Optimization*. Springer, 2016, pp. 58–73.

[14] S. Herrmann, G. Ochoa, and F. Rothlauf, "Communities of local optima as funnels in fitness landscapes," in *GECCO*. ACM, 2016, pp. 325–331.

[15] S. L. Thomson, F. Daolio, and G. Ochoa, "Comparing communities of optima with funnels in combinatorial fitness landscapes," in *GECCO*. ACM, 2017, pp. 377–384.

[16] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[17] A. Chehouri, R. Younes, J. Khoder, J. Perron, and A. Ilinca, "A selection process for genetic algorithm using clustering analysis," *Algorithms*, vol. 10, no. 4, p. 123, 2017.

[18] Q. Meng, J. Wu, J. Ellis, and P. J. Kennedy, "Dynamic island model based on spectral clustering in genetic algorithm," in *IJCNN*. IEEE, 2017, pp. 1724–1731.

[19] H. Wang, J. Chen, and K. Guo, "A genetic spectral clustering algorithm," *Journal of Computational Information Systems*, vol. 7, no. 9, pp. 3245–3252, 2011.

[20] D. Whitley, S. Rana, and R. B. Heckendorn, "The island model genetic algorithm: On separability, population size and convergence," *Journal of computing and information technology*, vol. 7, no. 1, pp. 33–47, 1999.

[21] I. BoussaïD, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information sciences*, vol. 237, pp. 82–117, 2013.

[22] R. K. Ursem *et al.*, "Multinational gas: Multimodal optimization techniques in dynamic environments." in *GECCO*. ACM, 2000, pp. 19–26.

[23] N. Melab, E.-G. Talbi *et al.*, "Gpu-based island model for evolutionary algorithms," in *GECCO*. ACM, 2010, pp. 1089–1096.

[24] J. Jaros, "Multi-gpu island-based genetic algorithm for solving the knapsack problem," in *CEC*. IEEE, 2012, pp. 1–8.

[25] C.-C. Li, J.-C. Liu, C.-H. Lin, and W. Lo, "On the accelerated convergence of genetic algorithm using gpu parallel operations," in *Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2017, pp. 1115–1130.

[26] C.-C. Li, C.-H. Lin, and J.-C. Liu, "Parallel genetic algorithms on the graphics processing units using island model and simulated annealing," *Advances in Mechanical Engineering*, vol. 9, no. 7, 2017.

[27] R. Ohira, S. Islam, J. Jo, and B. Stantic, "Lcs based diversity maintenance in adaptive genetic algorithms," in *AusDM*, 2018, pp. 56–68.

[28] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[29] R. Ohira and S. Islam, "A distributed genetic algorithm with adaptive diversity maintenance for ordered problems," in *PDCAT*, 2019.

[30] R. Ohira, M. S. Islam, J. Jo, and B. Stantic, "Amga: An adaptive and modular genetic algorithm for the traveling salesman problem," in *ISDA*, 2018, pp. 1096–1109.