

Abstract

PARTHEEPAN, RAJASOORIYAR. Hybrid Genetic Algorithms. (Under the direction of Dr. S. Ranji Ranjithan)

Genetic algorithms (GAs), a class of evolutionary algorithms, emerging to be a promising procedure for solving engineering optimization problems. As GAs are able to conduct global search with minimal simplifying assumptions about the problem as well as the corresponding decision space, they offer a good alternative to the many gradient-based nonlinear local search procedures. While the underlying operators of a typical GA are designed for global search, their ability to search locally by exploiting information in the vicinity of apparently good solutions is relatively weak. This results in rapid convergence to a relatively good solution followed by slow improvements to that good solution, making GA computationally inefficient. To alleviate this deficiency, GAs can be integrated with local-search procedures such that the strengths of both global and local search approaches are embedded into a hybrid search procedure. One approach is to couple sequentially a GA-based global-search with a local-search procedure. Alternatively, local-search steps can be integrated within the GA operators to potentially refine the solutions throughout the global-search steps in the GA. This research investigates four hybrid search procedures, one based on a sequential approach and the others as local-search-based operators within a GA. These methods and a simple GA are evaluated using a set of test problems, and their performance (in terms of solution quality and computation time) is compared. These performance comparisons are conducted for multiple random trials. These methods are also applied and tested on a realistic urban runoff control problem. Compared to a simple GA, all methods perform well in terms of solution quality and number of fitness evaluations (used as a surrogate for computational resource needs). One of the local-search-based operator methods outperforms others consistently and exhibits a robust performance, indicating a promising hybrid search approach to solving real engineering optimization problems.

HYBRID GENETIC ALGORITHMS

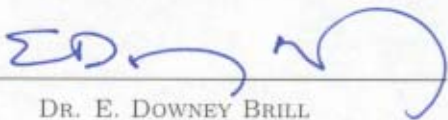
BY
RAJASOORIYAR PARTHEEPAN

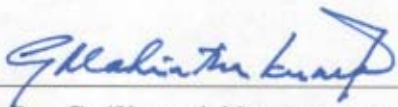
A THESIS SUBMITTED TO THE GRADUATE FACULTY OF
NORTH CAROLINA STATE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE


DEPARTMENT OF CIVIL, CONSTRUCTION, AND ENVIRONMENTAL ENGINEERING

RALEIGH
AUGUST 2003

APPROVED BY:


DR. E. DOWNEY BRILL


DR. G. (KUMAR) MAHINTHAKUMAR


DR. S. RANJI RANJITHAN

CHAIR OF ADVISORY COMMITTEE

Biography

Rajasooriyar Partheepan was born in Jaffna, Srilanka on December 21, 1974. He was raised with his two brothers and two sisters. He graduated from University of Peradeniya in July 2000 with BSc. (Hons) in Civil Engineering. He started his masters program in August 2001 at North Carolina State University. He worked as a research assistant under Dr S. Ranji Ranjithan during his masters program. His research interests are developing decision support tools, design and development of evolutionary algorithms for environmental engineering problems and software development.

Acknowledgement

I express my sincere gratitude to Dr Ranji S. Ranjithan at North Carolina State University for investing so much of his time and energy to attain my educational goals. This thesis would not have been possible without his help. I would also like to thank Dr. Downey Brill and Dr. G. Mahinthakumar for agreeing to be on my committee.

I am particularly grateful to Dr. G. Mahinthakumar for his valuable guidance throughout my masters program.

In addition, thanks to my graduate student friends Emily, Sunil and Siva and my roommates Satta, Bala and Vijay for the great time both in school and out.

My family, mother, father, brothers and sisters have always supported me, for that I am grateful.

Table of Contents

| | |
|---|-----------|
| List of Figures | v |
| List of Tables | vi |
| 1. Introduction | 1 |
| 2. Background | 2 |
| 2.1 Sequential Hybrid Procedures | 2 |
| 2.2 Local-search-based GA operators | 3 |
| 3. Methodology | 4 |
| 3.1 Hybrid Method 1: Sequential GA-NM Procedure | 5 |
| 3.2 Hybrid Method 2: Local-search-based Operator: ElitePatternMove Procedure | 6 |
| 3.3 Hybrid Method 3: Local-search-based Operator: Simplex GA Procedure | 7 |
| 3.4 Hybrid Method 4: Local-search-based Operator: Simplex Crossover | 9 |
| 4. Evaluation using test problems | 10 |
| 4.1 Comparison of Results: Sequential GA-NM vs. Simple GA | 10 |
| 4.2 Comparison of Results: Local-search-based Operators vs. Simple GA | 11 |
| 5. Application to an urban runoff control problem | 18 |
| 6. Summary and Final remarks | 24 |
| Reference | 25 |

List of Tables

| | | |
|---------|---|----|
| Table 1 | : Test problems and known optimal solutions | 10 |
| Table 2 | : Comparison of Sequential GA-NM method and Simple GA for Ackley function (for 30 random trials) | 11 |
| Table 3 | : Comparison of Sequential GA-NM method and Simple GA for Rosenbrock function (for 30 random trials) | 11 |
| Table 4 | : Comparison of Sequential GA-NM with Simple GA based on 25 trials (for 25 random trials) | 21 |

List of Figures

| | |
|---|----|
| Figure 1: Flow Chart for the Sequential GA-NM hybrid procedure | 5 |
| Figure 2: Pseudo code for the Elite Pattern Move hybrid procedure | 6 |
| Figure 3: Formation of a new population using the Simplex GA hybrid procedure | 7 |
| Figure 4: Weighted Centroid & Reflection Operation (solution c is better than solution b, which is better than solution a) | 8 |
| Figure 5: Pseudo code for the Simplex GA hybrid procedure | 8 |
| Figure 6: Pseudo code for the Simplex Crossover hybrid procedure | 9 |
| Figure 7: Comparison of local-search-based operators with Simple GA for Rastrigin function | 12 |
| Figure 8: Comparison of local-search-based operators with Simple GA for Ackley function | 13 |
| Figure 9: Comparison of local-search-based operators with Simple GA for Schwefel function | 14 |
| Figure 10: Comparison of convergence (for a single random seed) of Simplex GA and of Simple GA for Rastrigin function. | 15 |
| Figure 11: Comparison of convergence (based on 30 random trials) of Simplex GA and of Simple GA for Rastrigin function | 16 |
| Figure 12: Comparison of convergence (for a single random seed) of ElitePatternMove and of Simple GA for Rastrigin Function | 16 |
| Figure 13: Comparison of convergence (based on 30 random trials) of ElitePatternMove GA and of Simple GA for Rastrigin function | 17 |
| Figure 14: Comparison of convergence (for a single random seed) of SimplexCrossover and of Simple GA for Rastrigin function | 17 |
| Figure 15: Comparison of convergence (based on 30 random trials) of SimplexCrossover GA and of Simple GA for Rastrigin function | 18 |
| Figure 16: The catchment areas ((a) existing condition, and (b) after improvement) and sewer flows ((c) existing conditions, and (d) after improvement) corresponding to the hypothetical low development (LID) case study. | 20 |
| Figure 17: Comparison of local-search-based operators with Simple GA for low impact development problem | 22 |
| Figure 18: Comparison of convergence (based on 25 random trials) of Simplex GA and of Simple GA for LID Problem | 22 |
| Figure 19: Comparison of convergence (based on 25 random trials) of ElitePatternMove GA and of Simple GA for LID Problem | 23 |
| Figure 20: Comparison of convergence (based on 25 random trials) of Simplex Crossover and of Simple GA for LID Problem | 23 |

Chapter 1

Introduction

Genetic algorithms (GAs), a class of evolutionary algorithms, emerging to be a promising procedure for solving search problems associated with engineering systems analysis. Real-world applications of GAs in an array of engineering problems are frequently reported in the literature. As GAs are able to conduct global search with minimal simplifying assumptions about the problem as well as the corresponding decision space, they offer a good alternative to the many gradient-based nonlinear local search procedures. While the operators (such as selection and recombination) in a typical GA are designed for global search, their ability to search locally by exploiting information in the vicinity of apparently good solutions is relatively weak. Thus GAs are generally good at globally searching the decision space to quickly identify relatively better solutions, but are not efficient at making local refinements to improve upon the good solutions. This is characterized by typical GA performance where the algorithm rapidly converges to a relatively good solution but becomes slow to make any significant improvement to that good solution. This behavior, especially during the latter stage after the initial rapid convergence, makes GAs highly compute intensive as the models that represent the problem space need to be executed multiple instances over the iterations of the algorithm.

Recognizing these convergence properties of GA-based search procedures, studies have been conducted to integrate GAs with local-search procedures such that the strengths of both global and local search approaches are embedded into a hybrid search procedure. This expectedly improves not only the quality of the solution, but also the convergence time. Most studies examined hybrid approaches that employ sequentially a GA-based global-search with a local-search procedure (Yen et al. 1995). Among the several existing sequential hybrid procedures, a typical approach uses a GA-based search initially to conduct a global search, and then improves that solution by using a local-search technique. Alternatively, a few hybrid search procedures investigated ways to integrate the local-search steps within the GA operators to potentially refine the solutions throughout the global-search steps in the GA (Yen et al 1995, Renders and Bersini 1994). While these preliminary studies show promising results, more investigations are needed to explore newer approaches that integrate the local-search steps into the GA operators to enhance

the overall search capabilities of a GA. Further, additional study is needed to conduct a more comprehensive comparison of performance of these hybrid procedures.

This paper reports several new hybrid search procedures and their evaluations using several test problems. These hybrid procedures use a GA for the global search, and Nelder-Mead (NM) simplex technique as well as the steps from the Hooke and Jeeves method for local-search. The applicability of these new hybrid procedures is also demonstrated for a realistic urban runoff control problem. The following section provides a background to the existing hybrid methods, followed by a section that describes the new hybrid search procedures. The subsequent section reports and compares the results associated with the test problems. Then the description of the urban runoff control problem and the application of the new hybrid procedures to that problem are presented. The last section presents a summary of the findings and final remarks.

Chapter 2

Background

This section provides a brief background to the main categories of hybrid procedures that are reported in the literature. The two main categories are: sequential hybrid procedures, and local-search-based GA operators.

2. 1 Sequential Hybrid Procedures

The sequential procedures are the most commonly investigated GA-based hybrid techniques. They employ a global-search procedure and a local-search procedure in tandem or sequence to conduct the search. These hybrid techniques can be further categorized into two main types:

1. local-search procedure preceded by a global-search procedure
2. local-search procedure followed by a global-search procedure.

Procedures in type 1 typically use a local-search procedure to generate the initial population for the global-search method. For example, Okamoto et al. (1998) apply repeatedly a local-search method to a random initial GA population to locally improve each solution. The locally refined set of solutions is then used as the starting population for the GA-based global-search procedure. Alternatively, the local-search procedure can be applied in each generation of the GA to locally

improve the population of solutions. For example, Mathias et al. (1994) apply a local-search procedure at the beginning of each generation to make local refinements to the population of solutions. The GA-operators are then applied as in a typical GA to the set of refined solutions. While these procedures may hold some promise for some class of problems, the rationale for generally conducting local search prior to global search appears to be weak and potentially inefficient. Since a global search step makes large moves to explore the decision space, resulting in macro changes in the set of solutions, refinements made a priori via a local-search procedure are likely to be ineffective during the macro changes.

Alternatively, the second type of procedures use the global-search methods to first narrow the search space down to a few good regions, and then apply a local-search procedure to make refinements based on information local to those few good regions. As this results in the macro moves being conducted prior to the local search, the benefits of the subsequent local refinements can potentially be effective unlike in type 1 procedures. For example, Lobo and Goldberg (1996) and Chelouah and Siarry (2003) report hybrid approaches where the solution found by a GA is then used as the starting point for a local-search method.

As a GA-based search procedure results in a population of solutions, it is possible to use not just the best solution but also a set of nearly good solutions from the final population as starting points for a local search. One potential approach is to take a set of the good solutions from the final population of the GA and use them as separate starting points for the local search. This result in multiple instances of the local-search algorithm being implemented simultaneously, which increases the computational burden associated with conducting multiple local searches. Alternatively, the final GA population (or a subset of good solutions) that characterizes the good regions of the decision space can be used collectively to represent the starting conditions for a local search. One such approach that is investigated in this paper uses the Nelder-Mean simplex local-search procedure that uses a subset of nearly good solutions from the GA population to form the starting simplex for local search.

2.2 Local-search-based GA operators

Instead of applying the local search in sequence with a GA, the local search steps can be embedded within the GA-operators such that the local refinements are applied continually

throughout the global search. The primary goal of this category of procedures is to conduct local search starting from intermediate solutions as a GA continues to search the decisions space. An example of this is a method proposed by Xu et al. (2001) where a Hooke & Jeeve's method-based local-search procedure is applied to a small set of the population within the context of a micro GA. Starting from each solution in the small population of the micro GA, local refinements are made. As the population is small in a micro GA, the computational burden associated with the local search is limited. A new approach developed in this paper presents an alternative in which the Hooke and Jeeve's local-search step is applied to the best solution (or the elite solution) at any time during the GA-based search. This approach not only instills into the GA population local improvement to the best GA solution at every generation, but also keeps the computational needs manageable since the local search is limited to just one solution.

An alternative approach is to apply the local refinement concurrently with the GA operators, resulting in some solutions undergoing a local-search step while the others are subjected to the typical GA steps. For example, Yen et al. (1995) applied the Nelder-Mead simplex-like local search to a small number of solutions in the GA population to generate new solutions during the GA crossover operation. The rest of the solutions underwent the normal GA crossover operator.

Another approach is to conduct the crossover operator in a GA using a local-search-based operator to generate a locally modified new solution. Using such a local-search-based operator, the parent solutions are used to generate offspring solutions that replace them in the new generation. In contrast to the previous approach where the local-search step is applied in a greedy manner (i.e., accept only improved solutions), the local modification is not checked for improvement in the solution before adding it to the new population. For example, Render and Bersini (1994) proposed a simplex-crossover operator that takes a set of solutions from the GA population to generate a fraction of the offspring solutions.

Chapter 3

Methodology

The following four different hybrid search methods were developed and investigated in this paper. This section describes the key steps of these four methods.

3. 1 Hybrid Method 1: Sequential GA-NM Procedure

The first step in this method is a global search using a GA. The population of solutions at convergence is then used to construct the starting condition for a Nelder-Mead-based local search procedure. The convergence is defined based on the degree of improvement in the best solution during the GA-based global search. For example, the GA search is terminated when the improvement in the fitness of the best solution is less than a prespecified value. While other convergence criteria (e.g., population diversity, maximum number of generation, fitness distribution of the GA population, etc.) could be used, the non-improvement criterion was found to be simpler to set up and less dependent on the problem characteristics. At convergence, instead of using only the best solution, a set of good solutions from the converged GA population is used to form the initial simplex. The key steps of this procedure are shown in Figure 1.

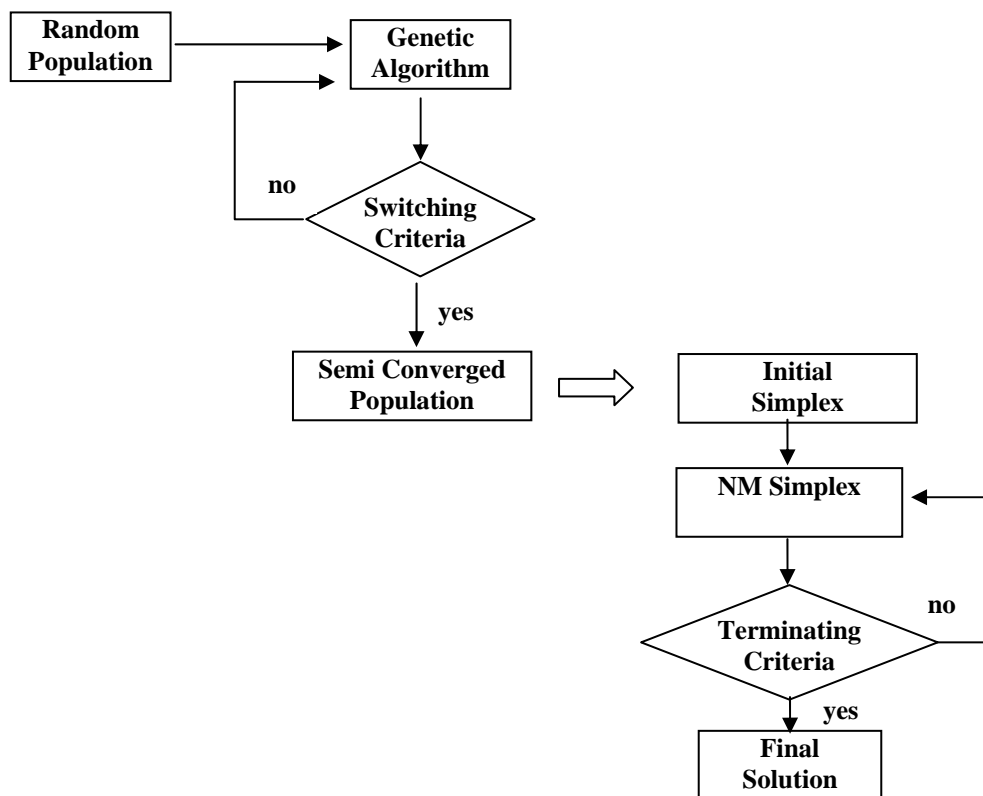


Figure 1– Flow chart for the Sequential GA-NM hybrid procedure

3.2 Hybrid Method 2: Local-search-based Operator – ElitePatternMove Procedure

In this procedure, a local-search step is applied within the GA as an additional operator. Similar to the approach suggested by Xu et al. (2001), this operator combines the “pattern move” in the Hooke and Jeeve’s local-search method with the “explorative moves” conducted by the GA. Typically the best (or the elite) solution found by the GA is stored at each intermediate generation. This set of elite solutions represents the outcome of the explorative steps of a GA. In a standard Hooke and Jeeve’s local-search procedure, an exploration move is combined with a pattern move to search the decision space. Instead, in this hybrid procedure a pattern move is applied to the elite solutions as the GA continues to explore the decisions space. The pattern move is applied based on the last elite solution and the current best solution to generate a new solution. If X_e is the elite solution and X_c is the current best solution, then a new solution $X_n = X_c + \alpha (X_c - X_e)$ is generated for an arbitrary value of the α parameter that represents the size of the local step. The results presented in this paper are based on the values 0.5 and 1.0 for this parameter. This pattern move represents a line search in the direction of local improvements. The resulting new solution is incorporated into the subsequent GA population. If the resulting new solution violates a constraint, then it is set to a point on the constraint boundary. Figure 2 shows the pseudo code for this procedure.

```
performElitism () {  
    if(currentBest( $X_c$ ) > previousBest( $X_p$ )){  
         $X_1 = X_c + 1.0 * (X_c - X_e)$   
         $X_2 = X_c + 0.5 * (X_c - X_e)$   
        ConstraintHandling:  
            If genes of  $X_1$  and  $X_2$  > upper bound or < lower bound then  
            set those genes to respective bound values  
  
        if( $X_1$  > population.worst1)  
            replace worst1 with  $X_1$   
        if( $X_2$  > population.worst2)  
            replace worst2 with  $X_2$   
    }  
}
```

Figure 2 - Pseude code for the Elite Pattern Move hybrid procedure

3.3 Hybrid Method 3: Local-search-based Operator – Simplex GA Procedure

This procedure introduces an alternative crossover operator that is applicable with the standard GA crossover operators. Similar to the GA crossover operators, this operator is applied to a subset of parent solutions to generate offspring solutions. This alternative operator is designed based upon a Nelder-Mead simplex method to perform local search. For example, $(n+1)$ parent solutions, where n is an arbitrary value ≥ 2 , are selected from the GA population, and the best among these solutions is taken through the typical Nelder-Mead simplex steps (e.g., reflection, expansion, and contraction) for local improvement based upon the simplex formed by the other n solutions (Figure 3).

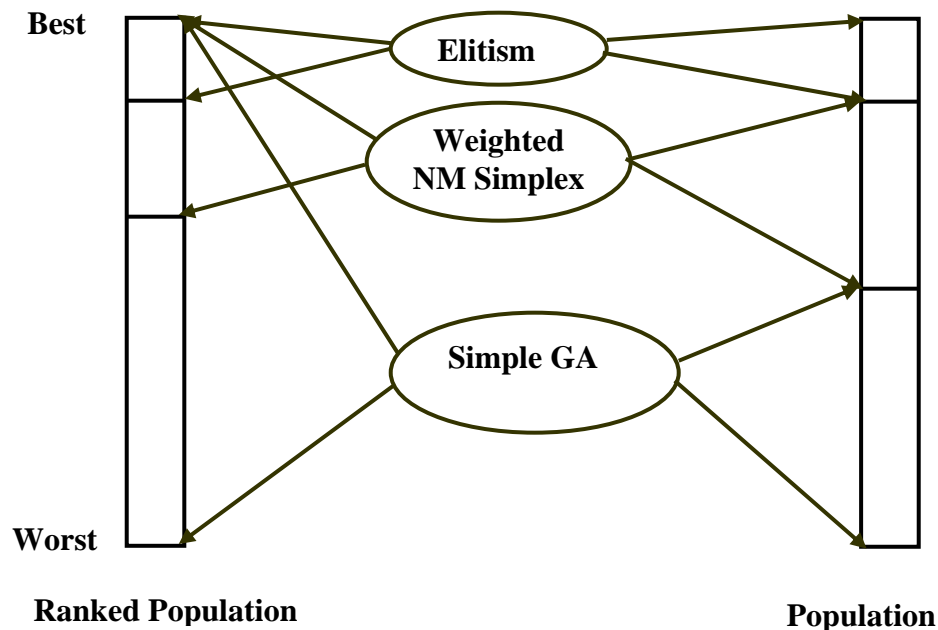


Figure 3 – Formation of a new population using the Simplex GA hybrid procedure

Unlike in the standard Nelder-Mead simplex method, the new operator defines the centroid of simplex by weighting each corner of the simplex by its fitness value. This results in a fitness-weighted centroid that is then used in the reflection, expansion and contraction steps (Figure 4)

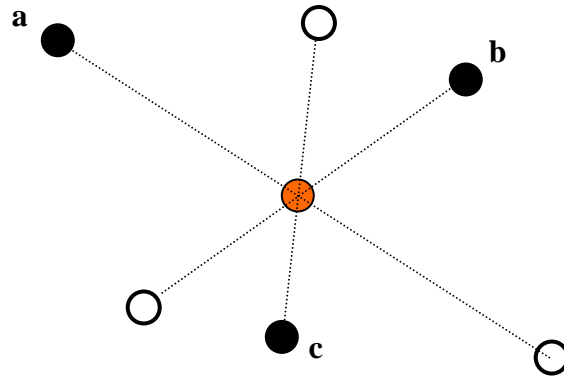


Figure 4--Weighted Centroid & Reflection Operation (solution c is better than solution b, which is better than solution a)

The proposed Simplex GA procedure employs this local-search-based crossover operator on a randomly selected portion of the GA population, and the rest of the population undergoes the typical GA crossover operators that are designed for global exploration of the decision space. The fraction that defines the portion of the population to which the new operator is applied is treated as a new algorithmic parameter. Figure 5 shows the pseudo code for Simplex GA.

```
Simplex GA {
    While (termination criteria is not true){
        Evaluate the population ()
        Sort population ();
        Elitism()
        Create two populations:
            Pop1: 0.1 * population size: top solution from the size of the simplex
            Pop2: 0.9 * population size : size of the simple GA , arbitrarily picked.
        Pop1: Simplex search
        Pop2: Simple GA search
    }
}
```

Figure 5 - Pseudo code for the Simplex GA hybrid procedure

3.4 Hybrid Method 4: Local-search-based Operator – Simplex Crossover

Similar to the previous method, this procedure (Figure 6) introduces a Nelder-Mead simplex method-based crossover operator. The primary difference is that only the reflection step is applied. Unlike the standard Nelder-Mead simplex method, the new solution found after the reflection step is not evaluated to determine if the expansion or contraction steps are to be applied. Instead, the solution found after this local move is included into the GA population, and the subsequent selection step in the GA will evaluate and appropriately propagate that solution. This is analogous to the typical GA crossover operator where the offspring solutions are incorporated into the population, which then eventually undergoes evaluation and selection to determine the survivability of those solutions in the next generation. As in the previous method, this simplex crossover operator is applied to a randomly selected portion of the population, and the rest is subjected to the standard GA crossover operator. While this new approach eliminates the need for additional fitness evaluations, it is relatively less “greedy” in making local refinements.

```
Perform Crossover(){  
    Sort the population  
    Create two sub populations:  
        Pop1: 0.2* population size : top solution after sorting  
        Pop2: 0.8* population size : randomly picked  
    Pop1: Simplex Crossover  
        Calculate the weighted centroid  
        Generate offspring using reflection operation using the weighted centroid'  
    Pop2: Blend crossover  
        Use traditional blend crossover  
  
    Shuffle the population  
}
```

Figure 6- Pseudo code for the Simplex Crossover hybrid procedure

Chapter 4

Evaluation using test problems

The four hybrid procedures described above were evaluated using several test problems for which the optimal solutions are known. These problems and the corresponding optimal solutions are shown in Table 1. The performance of each new method is compared with that of a simple GA with only the standard GA operators. To examine the robustness of these methods, 30 random trials were conducted for each method and for each instance of the problem.

Table 1 – Test problems and known optimal solutions

| Problem | Mathematical Formulation | Optimal Solution |
|------------|--|--|
| Ackley | $\text{Min } f(x) = 20 + e + 20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{10} x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^{10} \cos(2\pi x_i) \right)$ $-32.768 \leq x_i \leq 32.768$ | $x_i = 0 \quad \forall i$ $f(x) = 0$ |
| Rosenbrock | $\text{Min } f(x) = \sum_{i=1}^{10} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right)$ $2.048 \leq x_i \leq 2.048$ | $x_i = 1 \quad \forall i$ $f(x) = 0$ |
| Rastrigin | $\text{Min } f(x) = 100 + \sum_{i=1}^{10} \left(x_i^2 - 20\pi x_i \right)$ $-5.12 \leq x_i \leq 5.12$ | $x_i = 0 \quad \forall i$ $f(x) = 0$ |
| Schwefel | $\text{Min } f(x) = \sum_{i=1}^{10} -x_i \sin \left(\sqrt{ x_i } \right)$ $-500 \leq x_i \leq 500$ | $x_i = 420.9687 \quad \forall i$ $f(x) = 4189.829$ |

4.1 Comparison of Results: Sequential GA-NM vs. Simple GA

Tables 2 and 3 compare the performance (i.e., the average and standard deviation of fitness, which is being minimized, as well as the number of function evaluations based on 30 random trials) of the sequential GA-NM method with that of a simple GA for the Ackley function and the Rosenbrock function, respectively.

Table 2 – Comparison of Sequential GA-NM method and Simple GA for Ackley function (based on 30 random trials)

| Method | Best Fitness | | No. of Evaluations | | Error (Final Fitness – Global Optimum) | | |
|------------------|--------------|--------------------|--------------------|--------------------|---|---------|-------|
| | average | standard deviation | average | standard deviation | = 0 | 0 - 0.1 | > 0.1 |
| Sequential GA-NM | 0.5140 | 0.874 | 23081 | 8840 | 16 | 5 | 9 |
| Simple GA | 0.0001 | 0.0059 | 120000 | - | - | - | - |

Table 3 – Comparison of Sequential GA-NM method and Simple GA for Rosenbrock function (based on 30 random trials)

| Method | Best Fitness | | No. of Evaluations | | Error (Final Fitness – Global Optimum) | | |
|------------------|--------------|--------------------|--------------------|--------------------|---|---------|-------|
| | average | standard deviation | average | standard deviation | = 0 | 0 - 0.1 | > 0.1 |
| Sequential GA-NM | 1.3826 | 2.5417 | 35032 | 10671 | 22 | 0 | 8 |
| Simple GA | 0.0010 | 0.5957 | 120000 | - | - | - | - |

4.2 Comparison of Results: Local-search-based Operators vs. Simple GA

The hybrid methods 2-4 were applied to the Rastrigin, Schwefel, and the Ackley functions for 30 random seeds, and their results are compared with each other and as well as with the results obtained using the simple GA. Figures 2, 3, and 4 compare the average and +/- one standard deviation of the fitness, which is being minimized, and number of fitness evaluations at different degrees of convergence for each method.

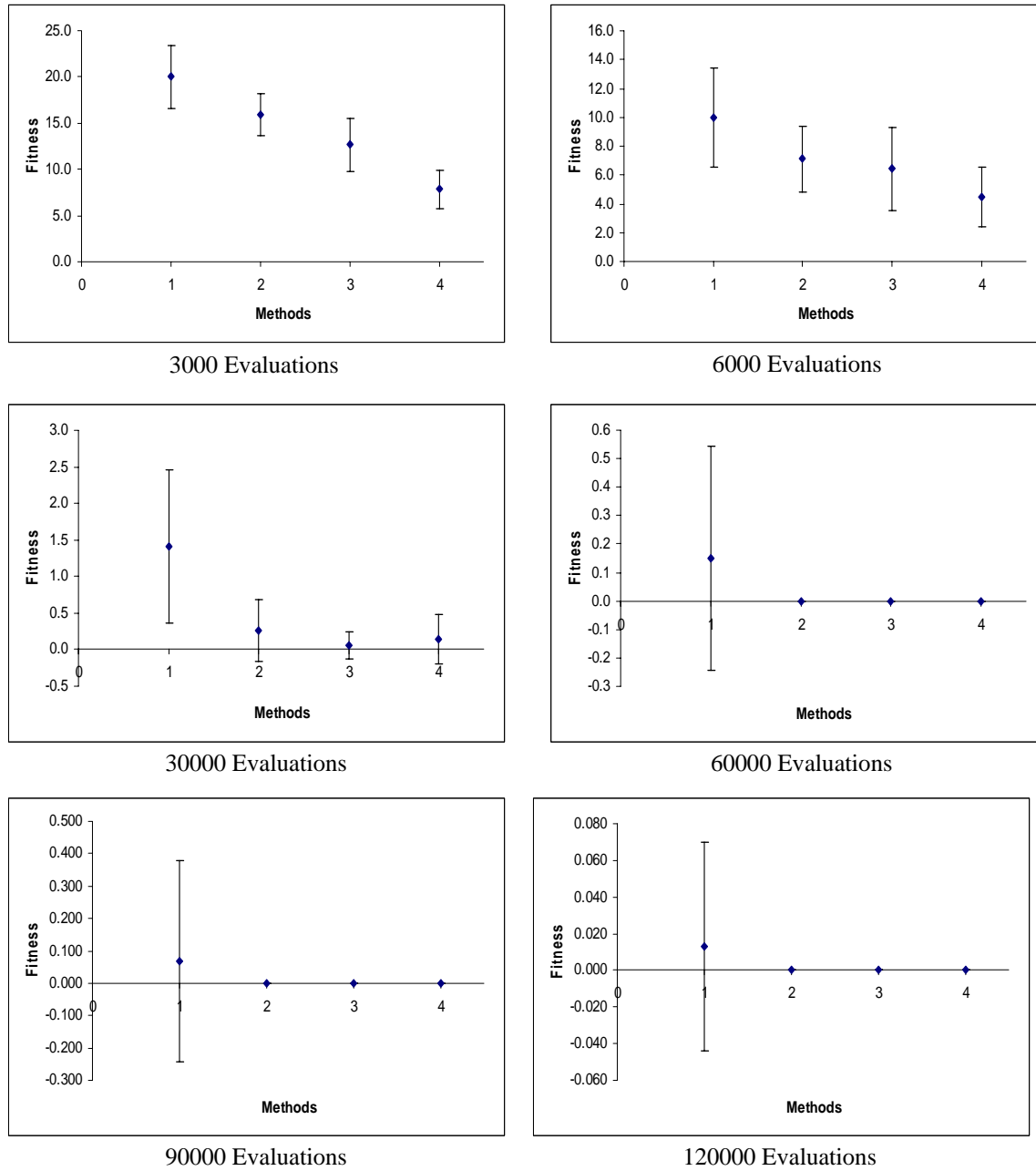
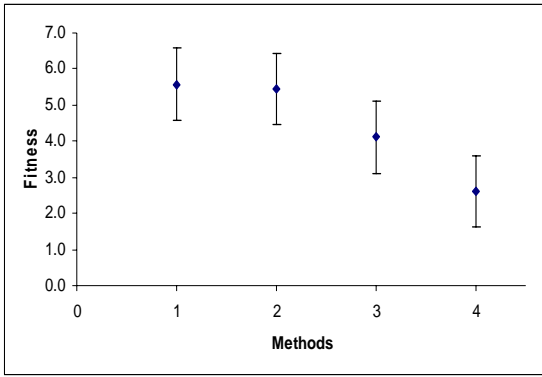
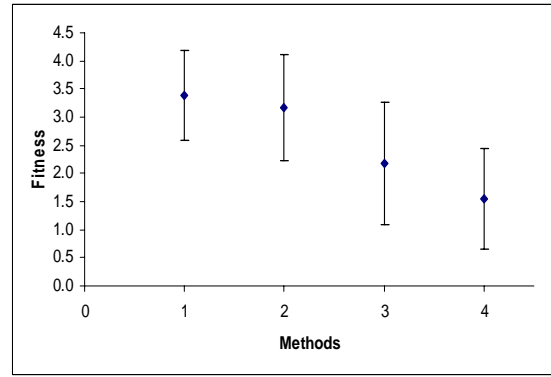


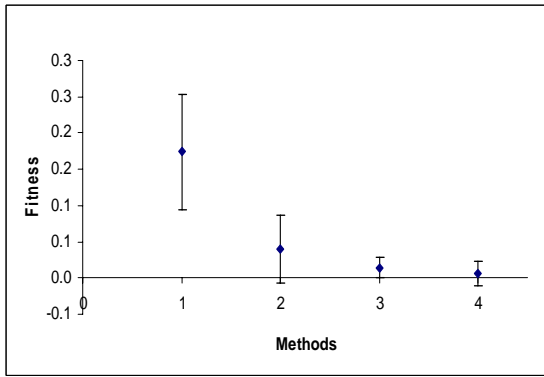
Figure 7 – Comparison of local-search-based operators with Simple GA for Rastrigin function (1 – Simple GA, 2 – Simplex Crossover, 3 – ElitePatternMove, 4 – Simplex GA)



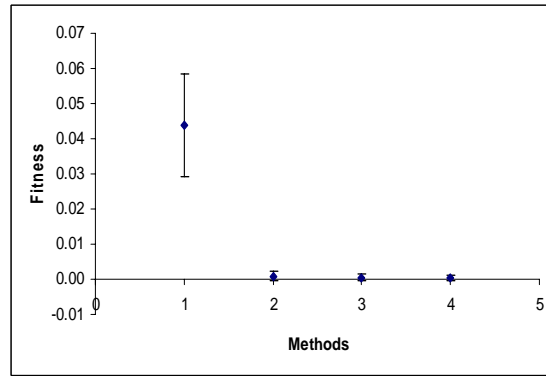
3000 Evaluations



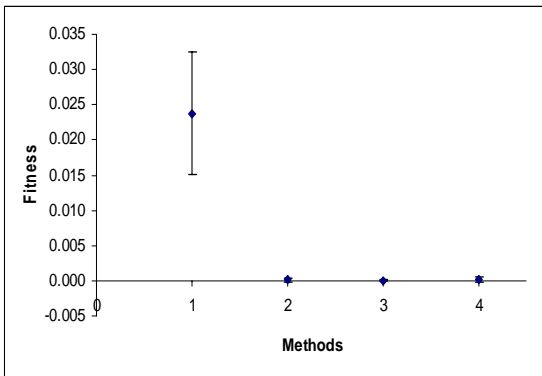
6000 Evaluations



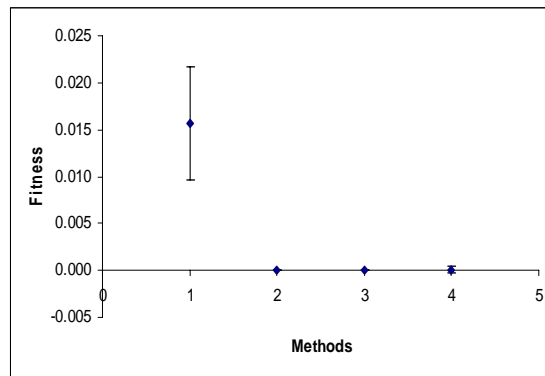
30000 Evaluations



60000 Evaluations



90000 Evaluations



120000 Evaluations

Figure 8 – Comparison of local-search-based operators with Simple GA for Ackley function (1 – Simple GA, 2 – Simplex Crossover, 3 – ElitePatternMove, 4 – Simplex GA)

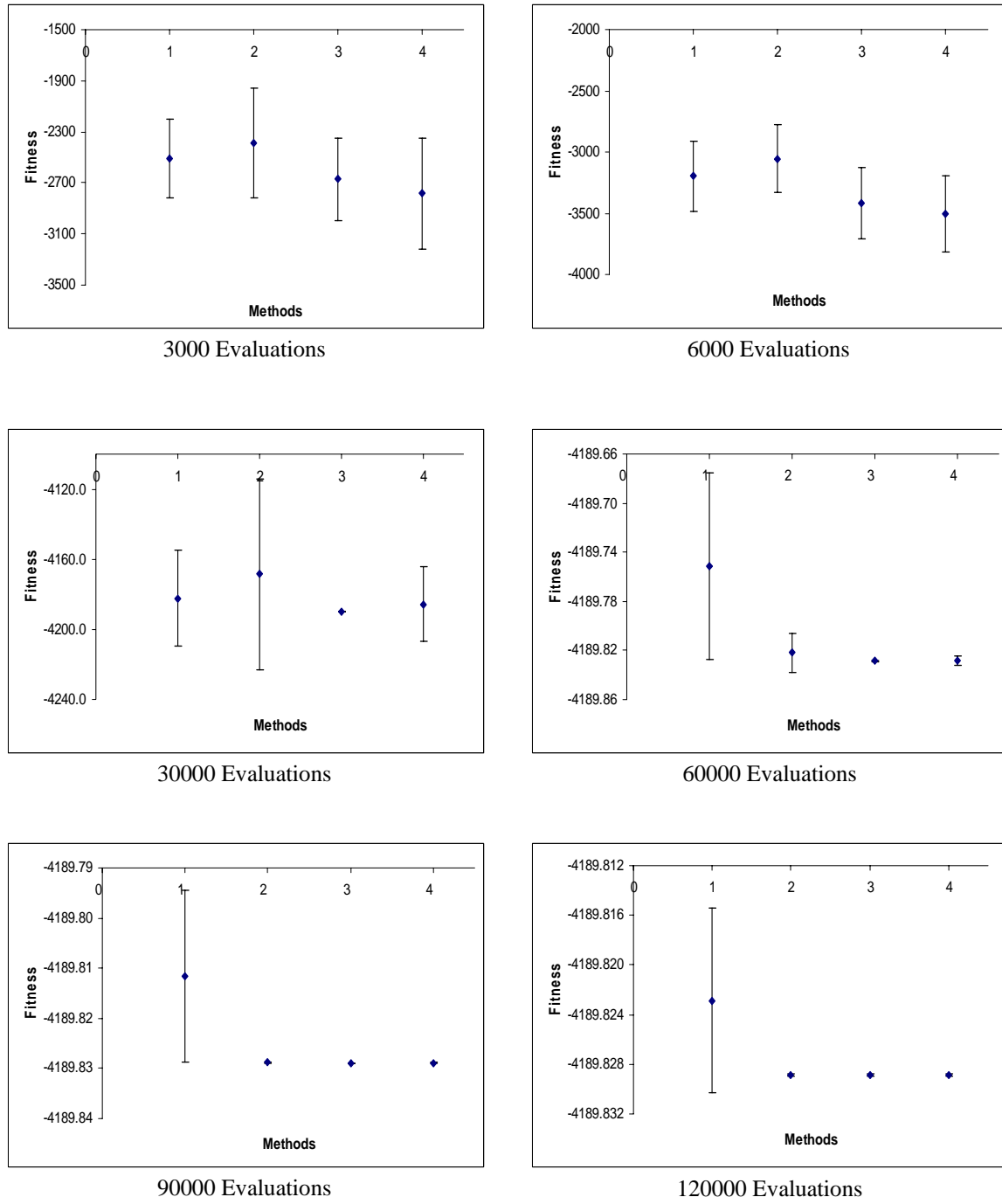


Figure 9 – Comparison of local-search-based operators with Simple GA for Schwefel function (1 – Simple GA, 2 – Simplex Crossover, 3 – ElitePatternMove, 4 – Simplex GA)

The typical convergence behavior of the hybrid method Simplex GA is represented in Figures 5 and 6 that show the convergence for a typical random trial and an aggregate representation of all 30 random trials, respectively, for the Rastrigin function. Similar comparisons are made for the other hybrid methods in Figures 7-10

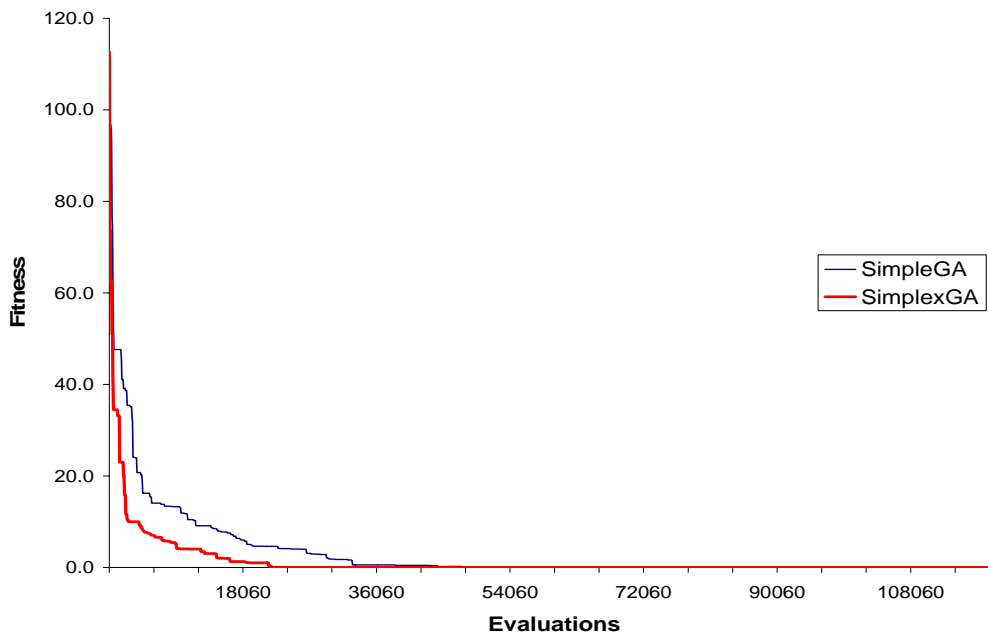


Figure 10 – Comparison of convergence (for a single random seed) of Simplex GA and of Simple GA for Rastrigin function

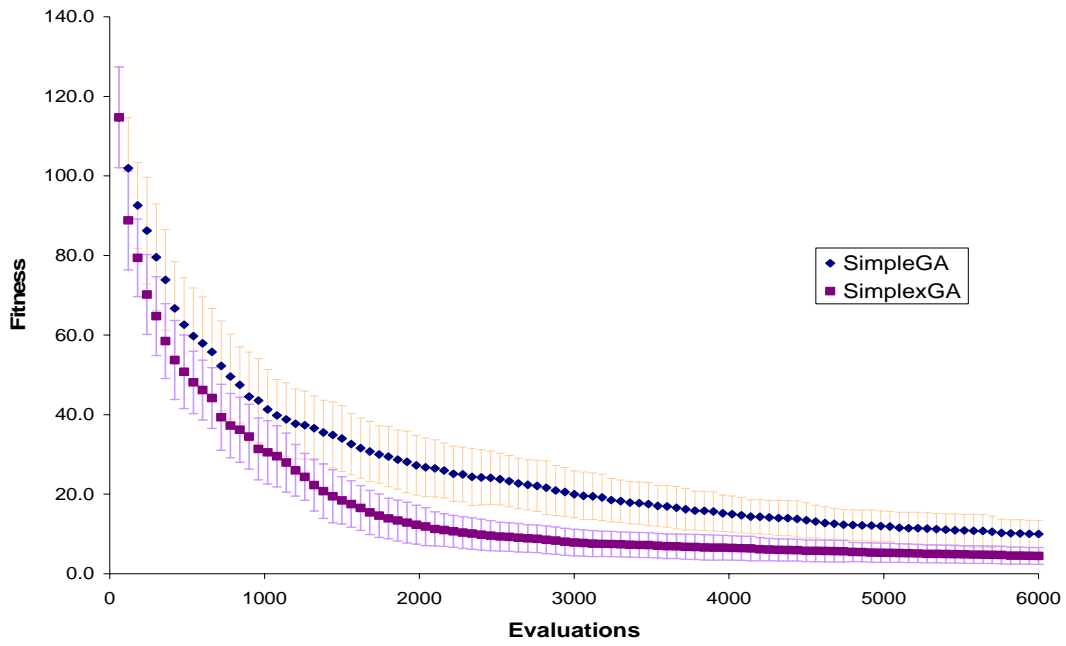


Figure 11 – Comparison of convergence (based on 30 random trials) of Simplex GA and of Simple GA for Rastrigin function

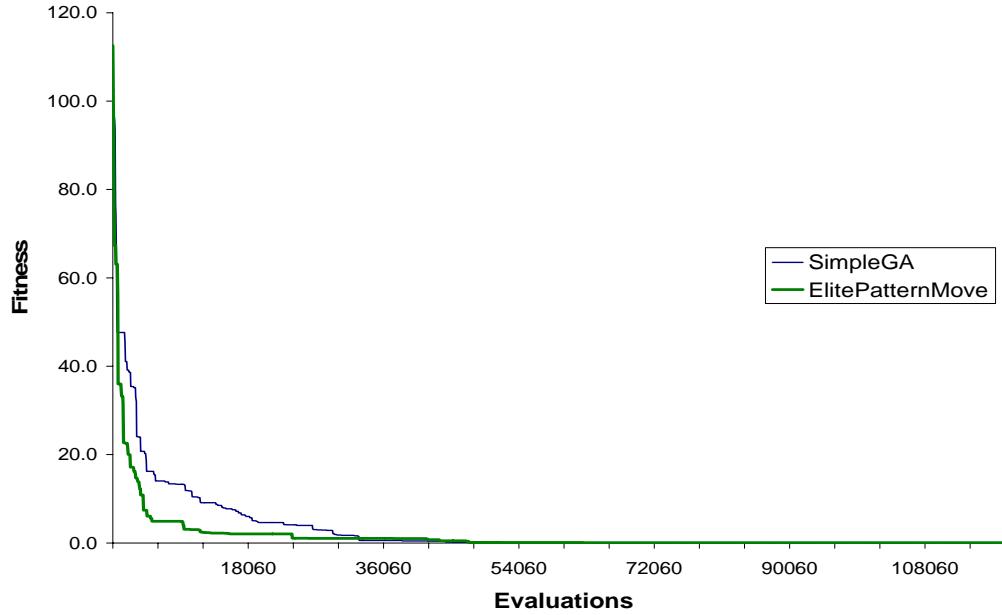


Figure 12 – Comparison of convergence (for a single random seed) of ElitePatternMove Method and of Simple GA for Rastrigin function

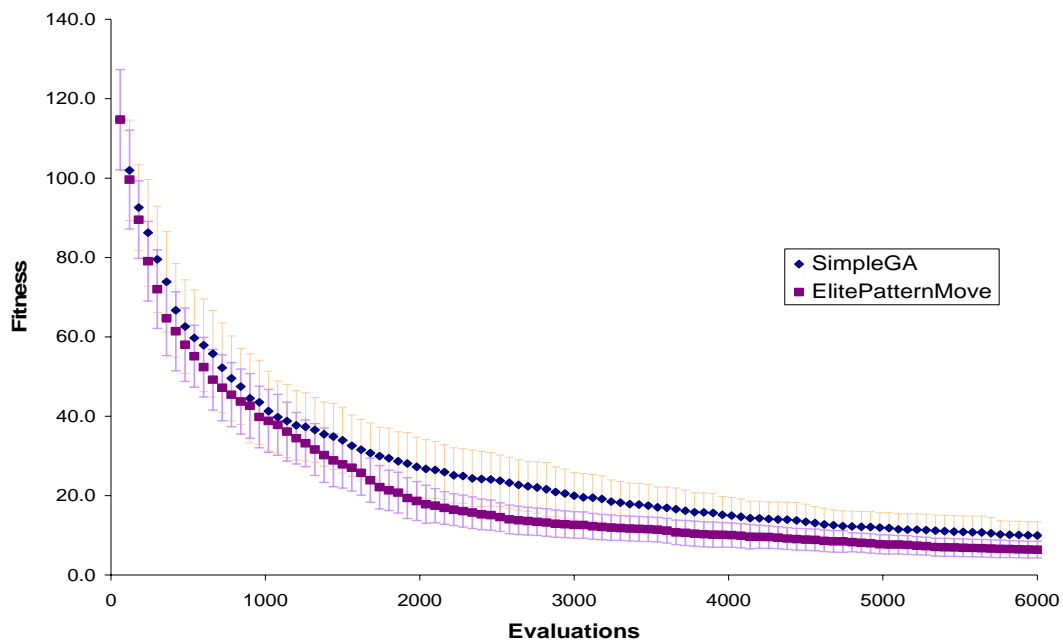


Figure 13 – Comparison of convergence (based on 30 random trials) of ElitePatternMove Method and of Simple GA for Rastrigin function.

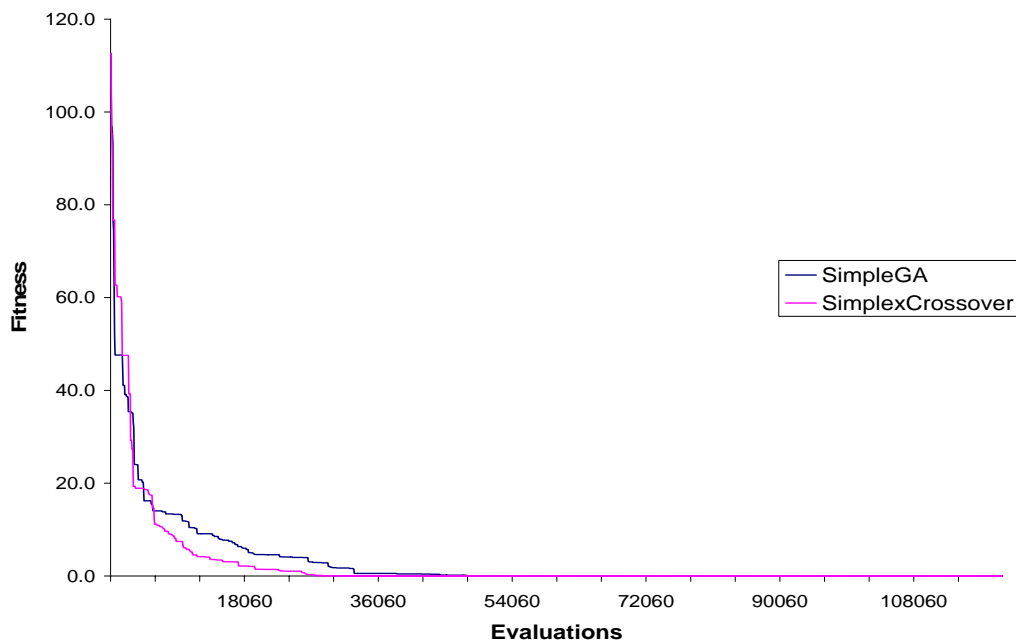


Figure 14 – Comparison of convergence (for a single random seed) of Simplex Crossover Method and of Simple GA for Rastrigin function.

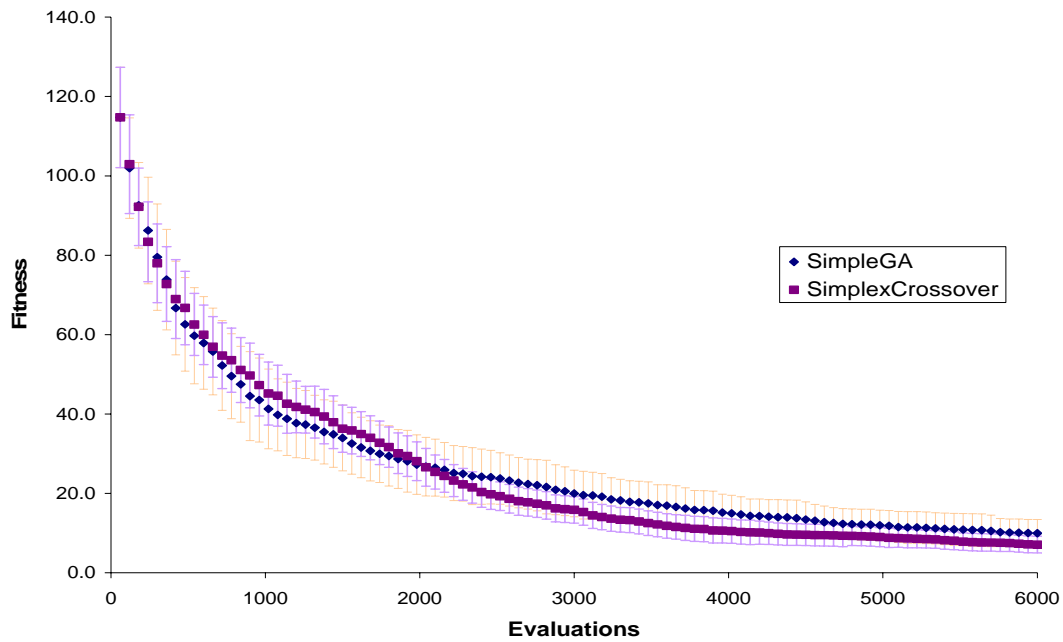


Figure 15 – Comparison of convergence (based on 30 random trials) of Simplex Crossover Method and of Simple GA for Rastrigin function.

Chapter 5

Application to an urban runoff control problem

The conversion of land from its natural state to the various residential and commercial uses changes the landscape dramatically, effectively reducing the pervious area. These land-use changes impact the hydrologic cycle of the watershed or the drainage basin. In general, these modifications to the landscape result in increased rainfall runoff relative to its undisturbed state. The excess runoff is conveyed via drainage networks to prevent localized flooding during rain events. Often this drainage network consists of a dendritic conduit-node network consisting of buried pipes and manholes.

The increased rainfall runoff due to unmanaged land use development often has a deleterious impact on downstream receiving water bodies. For example, increased stream velocities erode natural channel banks, smothering aquatic organisms with eroded sediment. Low Impact Development (LID) is a guiding principle to be applied during the land use development process where one seeks to mimic predevelopment site hydrology as much as possible. LID involves designing techniques that promote storage, infiltration, and evaporation to reduce the amount of runoff from a watershed area. Using the Storm Water Management Model (SWMM), one may simulate practices associated with LID by treating the watershed pervious and impervious detention storage parameters. The following scenario seeks to identify the optimal spatial allocation of additional depression storage (through LID that affects the pervious & impervious areas) necessary in a small watershed (Figure 11) consisting of eight sub-catchments and 16 conduits, such that the conduits design capacity is not exceeded.

This optimization problem is represented mathematically as follows:

$$\begin{aligned} \text{Min } \sum_{i=1}^8 ((x_i - x_0) * a_i + (y_i - y_0) * b_i) & \quad (1) \\ \text{subject to } v_j / V_j \leq 0.75 \quad j = 1, 2, \dots, 16 & \quad (2) \end{aligned}$$

- x_i = Pervious destore coefficient
- x_0 = Initial pervious destore coefficient
- y_i = Impervious destore coefficient
- y_0 = Initial impervious destore coefficient
- a_i = Pervious area of the catchment.
- b_i = Impervious area of the catchment
- V_j = Maximum capacity of the pipe
- v_j = Flow in the pipe

The flow in pipe v_j is estimated by running the Storm Water Management Model (SWMM) that simulates these values for a design rainfall event. The objective (Eqn. 1) of this optimization model is to minimize the additional storage that is required to meet the primary set of constraints (Eqn. 2) on the flow in the pipes. Constraint violations are handled through penalty functions.

The fitness of a solution is calculated using normalized values of the objective function and the penalty functions.

All four hybrid techniques were applied to this problem and the results are compared based on 25 random trials. Figure 11(b) shows the condition of the watershed after improvement via LID corresponding to a typical optimal solution that represents the least increase in depression storage. Also Figure 11(c) and (d) shows the energy profiles during the design peak runoff for the existing and improved conditions. The improved conditions result in flows in pipes to stay below full to avoid flooding.

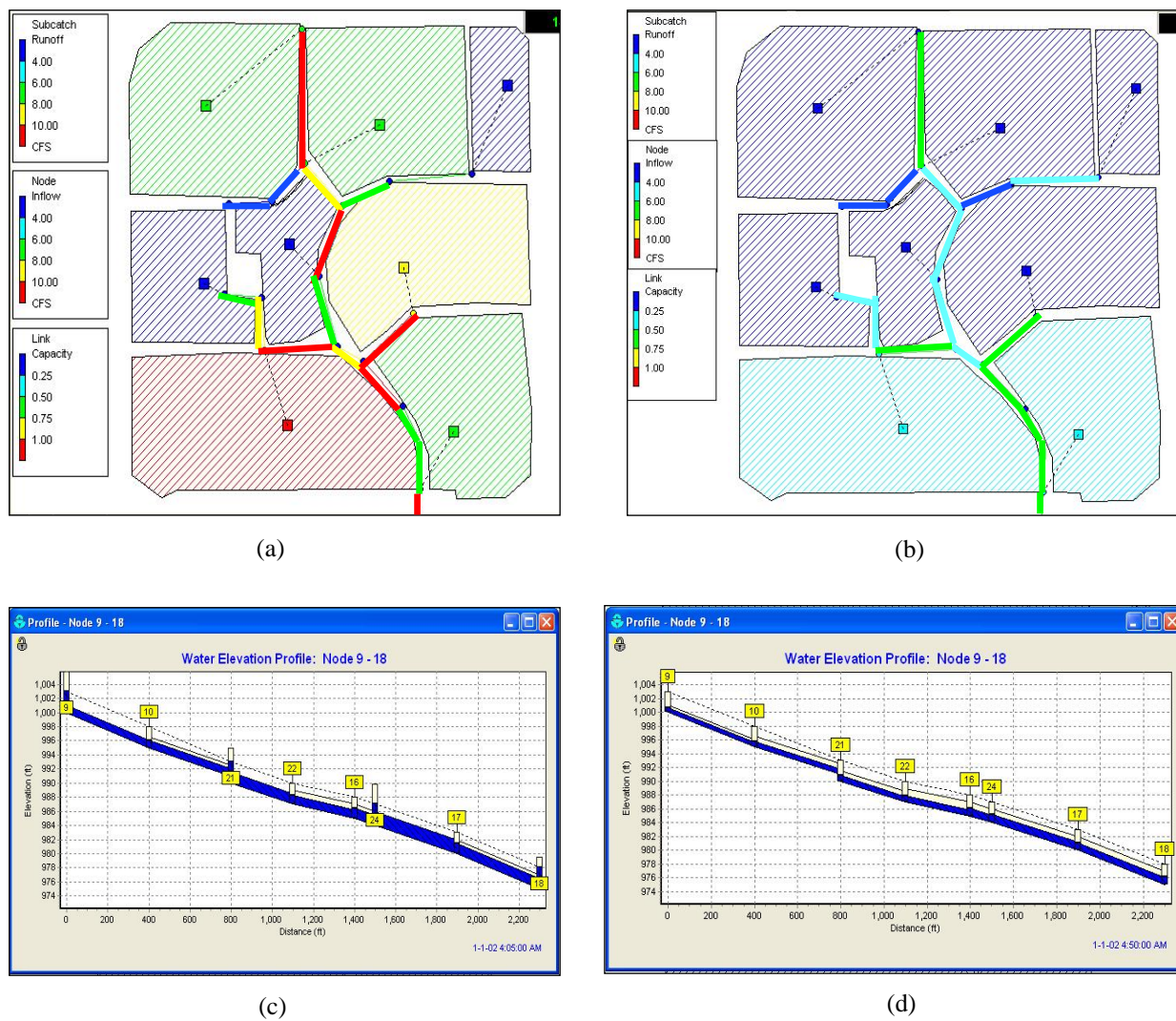


Figure 16– The catchment areas ((a) – existing condition, and (b) – after improvement) and sewer flows ((c) – existing conditions, and (d) – after improvement) corresponding to the hypothetical low development (LID) case study.

The Table 4 compares the performance (in terms of the best objective function value and the number of fitness evaluations needed to achieve the best solution) of the Sequential GA-NM method with that of the Simple GA. The results are reported in terms of the average and standard deviation obtained based on the 25 random trials.

Table 4 – Comparison of Sequential GA-NM method and Simple GA for the LID Problem (based on 25 trials)

| Method | Minimum Additional Storage | | No. of Evaluations | |
|------------------|-----------------------------------|--------------------|---------------------------|--------------------|
| | average | standard deviation | average | standard deviation |
| Sequential GA-NM | 0.4256 | 0.0172 | 16111 | 5444 |
| Simple GA | 0.4231 | 0.0186 | 30296 | 8912 |

The results corresponding to the local search base operators are summarized and compared in Figure 12. As presented for the test function results, this figure compares the average and the standard deviation of the fitness (i.e., minimum additional storage) at different stages of the search (as represented by the different numbers of fitness evaluations). The convergence behaviors for these methods are also compared in Figures 18-20.

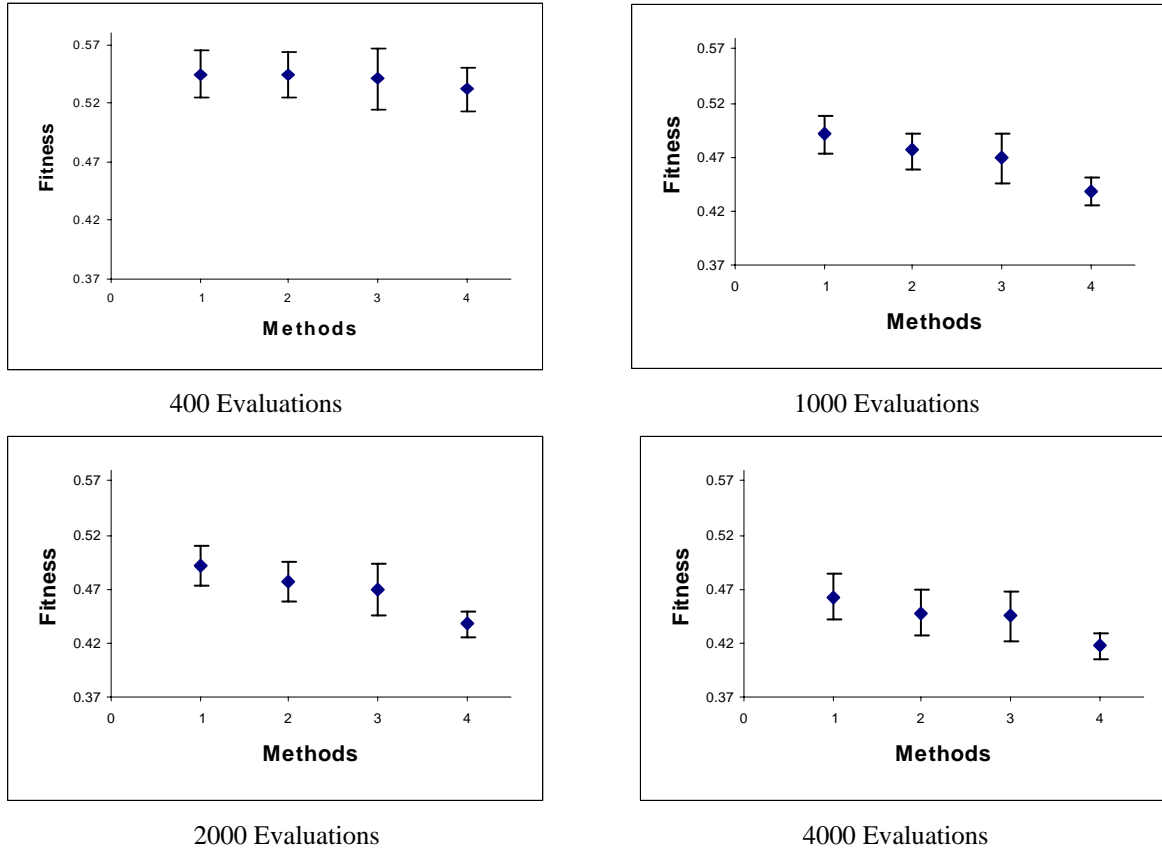


Figure 17 – Comparison of local-search-based operators with Simple GA for the LID problem (1 – Simple GA, 2 – Simplex Crossover, 3 – ElitePatternMove, 4 – Simplex GA)

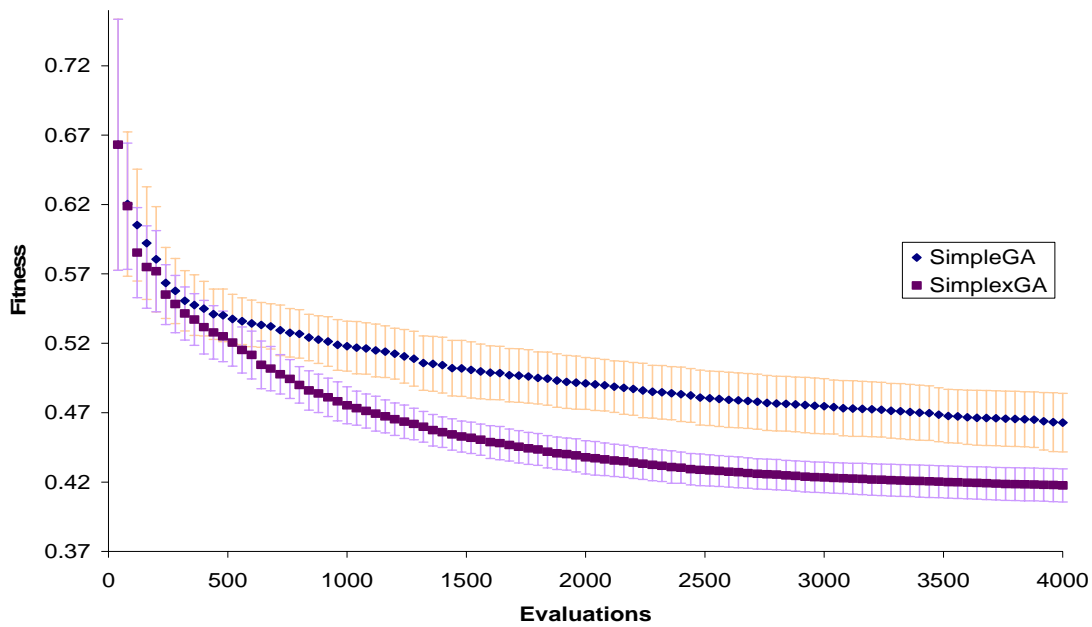


Figure 18 – Comparison of convergence (for 25 random trials) of Simplex GA and of Simple GA for the LID problem

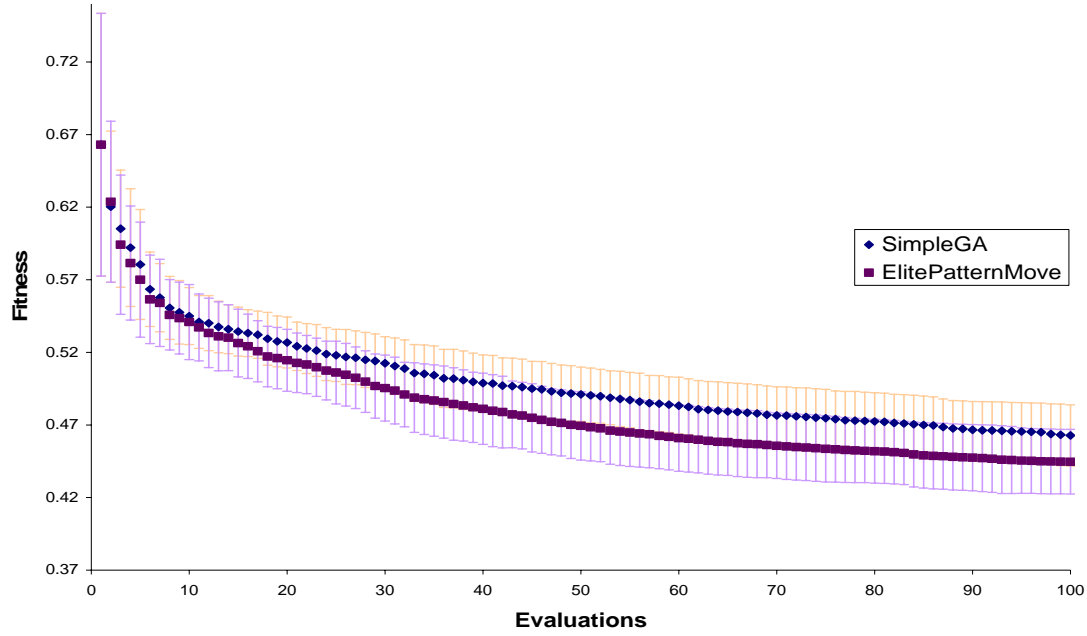


Figure 19 – Comparison of convergence (for 25 random trials) of ElitePatternMove Method and of Simple GA for the LID problem

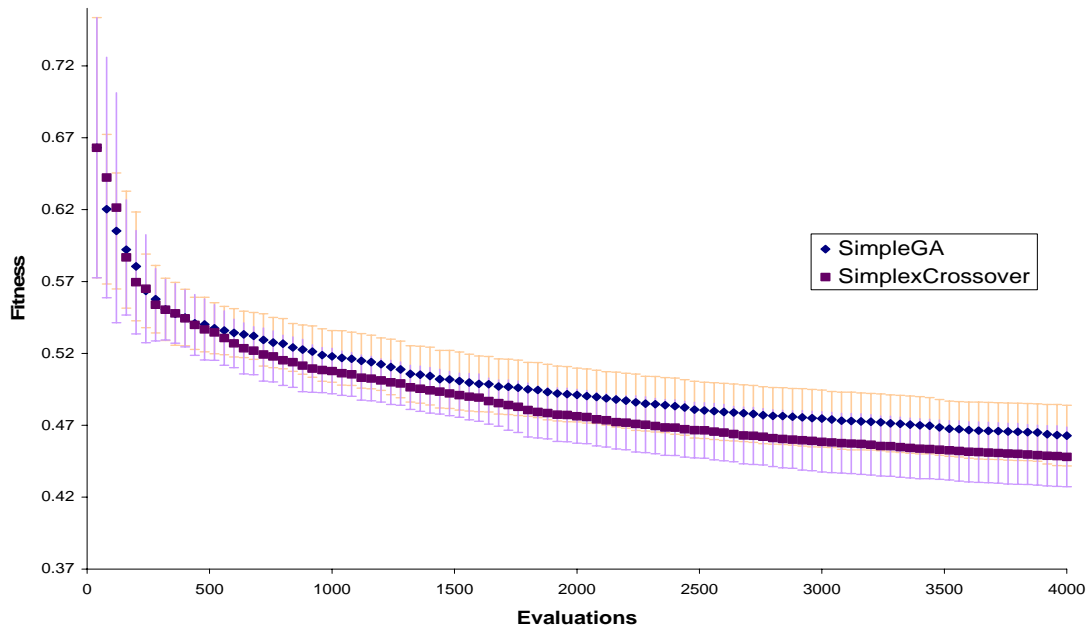


Figure 20 – Comparison of convergence (for 25 random trials) of Simplex Crossover Method and of Simple GA for the LID problem

Chapter 6

Summary and Final remarks

Several hybrid search methods that combine the power of genetic algorithms and local-search procedures were presented. While one method is built on the standard approach of applying a global search and local procedure sequentially, the others innovatively combine the key algorithmic steps in global and local search methods to explore efficiently the decision space. Using several test functions, the performance of these hybrid methods was compared with each other as well as with that of a simple GA. This comparison was conducted based on the quality of the solutions and the number of fitness evaluations (used as a surrogate for computational need). While all hybrid procedures show improvement over the simple GA-based search, the Simplex GA hybrid method shows the best overall performance. The ElitePatternMove operator method requires relatively a smaller number of additional fitness evaluations and yields good, but not necessarily the best, quality of solutions. The Simplex Crossover operator method shows the least improvement in the quality of solutions although it requires the least increase in the number of fitness evaluations. The sequential hybrid method quickly converges to a good solution but not necessarily to the best solution compared to that obtained using the Simple GA. In comparison, the sequential hybrid method typically required about half the number of evaluations to get a solution of similar quality.

These hybrid methods were also applied to an illustrative urban watershed management problem involving a hypothetical, but realistic, optimal runoff control scenario. The flow conditions in the watershed were simulated using SWMM, which was coupled to the hybrid search methods in evaluating the fitness of each solution. The performance of the hybrid methods as well as a simple GA was compared. The trends in the performance were similar to those found for the test problems—the Simplex GA showed the most robust performance.

While consistent observations regarding the relative performance of these hybrid methods were made in this study, more testing is required. Such a testing should include a broader class of test problems, including different levels of constraints, with varying degree of difficulty. Additional testing with more realistic problems is also needed. While ongoing work is investigating these

hybrid methods for more complex scenarios of the watershed management problem, application to different types of real problems is required to make a more comprehensive conclusion about the effectiveness of each of these hybrid methods.

Reference

1. Yen, J., Liao, J.C., Randolph, D., Lee, B., A hybrid approach to modeling metabolic systems using genetic algorithm and simplex method, In Proceedings of the 11th IEEE Conference on Artificial Intelligence for Applications, pp. 277—283, 1995.
2. Renders, R., Bersini, H., Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways, In Proceedings of the First IEEE Conference on Evolutionary Computation, pp312-317, 1994
3. Xu, Y.G., Li, G.R., Wu, Z.P., A Novel Hybrid Genetic Algorithm using local optimizer based on heuristic pattern move. Applied Artificial Intelligence vol. 15, pp. 601-631, 2001
4. Chelouah, R., Patrick, S., Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multi-minima functions, European Journal of Operational Research, vol. 148, pp. 335-345, 2003.
5. Whitley, L., D., Winter, G., P'eriaux, J., Gal'an, M., Cuesta, P., Modeling hybrid genetic algorithms. Genetic Algorithms in Engineering and Computer Science, pp. 191--201, 1995.
6. Lobo, F., G., Goldberg, D., E., Decision making in a hybrid genetic algorithm, Illegal report No. 96009, 1996.
7. Tsutsui, S., Goldberg, D., E., Simplex Crossover and Linkage Identification: Single-Stage Evolution Vs Multi Stage Evolution, Illegal Report No. 2002001. 2002.
8. Mathias, K., E., Whitley, L., D., Stork, C., Kusuma, T., Staged hybrid genetic search for seismic data imaging, In Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 356-361, 1994.
9. Lim, M., H., Yuan, Y., Omatu, S., Extensive testing of a hybrid genetic algorithm for solving quadratic assignment problems, Computational Optimization and Applications, vol. 23(10), pp. 47-64, 2002.
10. Li, M, Tam, H., Y., Hybrid evolutionary search method based on clusters, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23 (8), pp. 786-799, 2001.

11. Fleurent, C., Ferland, J., A., Hybrid genetic algorithms for combinatory optimization, *Rairo-Recherche Operationnelle-Operations Research*, vol. 30 (4), pp. 373-398, 1996.
12. Richard, Y., K., Jiafu, T., Dingwei, W., Extension of a hybrid Genetic algorithm for nonlinear programming problems with equality and inequality constraints, *Computers & Operations Research*, vol. 29, pp. 261-274, 2002.
13. Duan, Q., Gupta, V., Sorooshian, S., Shuffled complex evolution approach for effective and efficient global minimization, *Journal of Optimization. Theory Application*, vol. 76 (3), pp. 501-521, 1993
14. Michalewicz, Z., *Genetic Algorithms + Data structures = Evolutionary Programs*, Springer, Berlin. 1992.
15. Okamoto, M., Nonaka, T., Ochiai, S., Tominaga, D., Nonlinear numerical optimization with the use of a hybrid Genetic Algorithm incorporating the Modified Powel, *Applied Mathematics and Computation*, vol. 91, pp. 63-72, 1998.