CSE 848 Project Report
Extending Applicability of Groundwater Flow Algorithm through Algorithmic Equivalence

Submitted by: Ritam Guha (MSU ID: guharita)

Date: March 26, 2021

1 Abstract

2 Introduction

Algorithmic Equivalence can be defined as the process of finding *connections* between two different algorithms and representing them in a common framework. The purpose of finding this common framework is that if two algorithms are found equivalent, it is possible to borrow useful operators from one algorithm to the other to improve their individual performances. Inpsired from the success achieved in [1] to improve the performance of Particle Swarm Optimization (PSO), it was interesting to extend the concept of algorithmic equivalence for other algorithms in varying problem settings. For this purpose, a recently proposed meta-heuristic algorithm named Groundwater Flow Algorithm (GWFA) [2] has been selected as the underlying optimization algorithm. The goal of the project is to improve the performance of GWFA over multiple Unimodal and Multimodal problems using algorithmic equivalence with different algorithms.

In statistics, Unimodal problems are defined as problems where objective functions are having a single peak in its distribution. On the other hand, Multimodal Problems may have several peaks in their objective distributions. The goals for these two types of problem settings are often considered to be different. In most of the case, the goal of unimodal problems is to get to the best solution in the objective space as fast as possible. On the other hand, in 1987, Goldberg et. al formulated the goal of multimodal problems different from unimodal problems. In [3], they argued that if the goal of multimodal problems still remain to find the best solution, the algorithms may run into some trouble when small initial population may allow some sampling error. Then the algorithms may start over estimating some schemata and finally they may reach the wrong optima. For this reason, the goal of the multimodal problems was further shifted to find all the different peaks at once. The later literature [4] have followed this definition and kept on conceiving multimodal problems using this objective.

From the previous discussion, it is clear that the same algorithm cannot work efficiently for both types of problems because different procedures are needed to satisfy these different (somewhat contradicting) goals or objectives. For this reason, these two problems are suitable for testing out the power of algorithmic equivalence. There are several advantages of doing algorithmic equivlence in different problem settings.

- First of all, different programmers may be comfortable using separate algorithms. According to the *No Free Lunch* theorem, there is no algorithm which is better than all the other algorithms for the entire set of problems. So, if programmers want to extend the applicability of the preferred algorithm over different problem domains where the algorithm does not work well, algorithmic equivalence is a great way to do that. The goal then becomes to make it algorithmically equivalent to another algorithm which works well for the destination problem, if possible.

- Algorithmic Equivalence is a way to discover strengths and weaknesses of a particular algorithm with the help of some other algorithms in different problem settings.

In this project, GWFA is used as the index algorithm which is modified with the help of algorithmic equivalence over the two sets of problems. GWFA is a nature-inspired optimization algorithm proposed in 2020 based on the flow of groundwater from one place to another. At the final stage of water cycle, i.e. precipitation, the water precipitated in the form of hail, rain, snow etc. recharges the water beds present undergorund. These areas are known as Recharge Areas (RAs) while the places where groundwater gets exposed to the surface of the earth (like streams, rivers etc.) are known as Discharge Areas (DAs). The flow of groundwater from RAs to DAs is known as groundwater flow and French mathematician Darcy was the first person to mathematically describe this flow. GWFA uses this concept for optimization by considering the best solutions as DAs and the other solutions as RAs where the goal for the RAs is to move towards the DAs (the better solutions found by the algorithm) with an expectation to improve their own solutions and explore the space in the process.

A careful observation at the GWFA algorithm will illustrate the fact that some of its features are similar to Evolutionary Algorithms (EAs) in many aspects. For example, the position update procedure for any RA in GWFA involves three candidate solutions (One DA, local average of all the better RAs and the RA under consideration). Although the equation for combining these three candidate solutions involve terms like hydraulic gradient and velocities, it is ultimately a way of recombination which is a very important operator in the domain of EA. Some other concepts like Mutation, Parent-Child Comparison are also present in GWFA which makes it a great candidate to perform algorithmic equivalence with EAs.

The rest of the report is divided into 3 sections. Section 3 describes the project in detail explaining the problems, the gradual progress with algorithmic equivalence, experimental settings and metrics. In the following section, the experimental outcomes and detailed analysis are provided. FInally, the project is conluded in the 5^{th} section.

3 Project Description

In this section, the overall structure of the project, the goals and the gradual progress has been described in detail. The final objective of the project is to explore the prowess of algortihmic equivalence using GWFA as the underlying test algorithm and applying it over unimodal and multimodal problems. In order to successfully perform GWFA's algorithmic equivalence with other EAs, the first process is to represent GWFA in a common framework as other EAs. After careful observation of GWFA's inner operations, it has been observed that GWFA can be represented as the sketch shown in Figure 1.

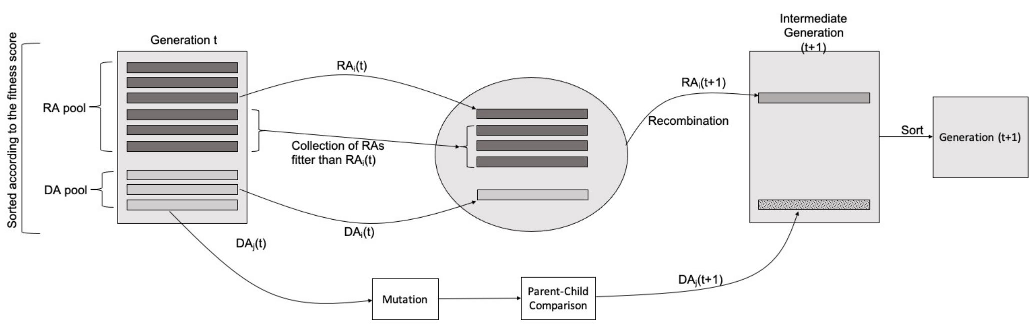

Figure 1: Representation of GWFA in an Evolutionary Algorithm Framework

In the next part, the two types of problems are discussed one after the other.

3.1 Unimodal Problems

As a part of this project, GWFA and its variants are applied over three popular unimodal problems: Ellipsoidal, Schewefel's and Rosenbrock's function. EAs like G3-PCX [5] are known to produce great results for these problems. The goal of this algorithmic equivalence is to check how GWFA's performance over these problems can be improved by importing different operators from G3-PCX. As it works great over these problems, it can be assumed that the operators in G3-PCX are well-suited for unimodal problems and they should improve GWFA's performance. Please note that the objective of this project is not to discuss the superiority of one algorithm over the other. The target is to improve GWFA's performance and in the process, learn how one algorithm's inner operations can help another algorithm to grow.

3.1.1 Experimental Setting

The goal of the experiment is to make the algorithm converge as fast as possible for the three functions. The convergence is defined as finiding a value ≤ 0.1 (S_1 criterion) or $\leq 10^{-20}$ (S_2 criterion). Every variant of GWFA is run 50 times. If all the runs converge, the variant is tested for quality based on the number of function evaluations it required. If some of the runs did not converge, then the number of times it converged is mentioned in brackets. If the variant did not converge at all for any run, the best values found by the runs are recorded. The outcome of the experimentation is provided in Table 1.

3.1.2 Order of Operator Imports

At the beginning, only the S_1 criterion was considered for the GWFA variants. The order in which different operators of G3-PCX are included to the standard GWFA is explained below:

- **Standard GWFA:** At first, the standard version of GWFA was applied over the three unimodal problems mentioned before. For Ellipsoidal function, it was able to find a convergence for all the independent runs, but for Schewefel's function, it converged only 4 times and for Rosenbrock's, it did not converge at all.

- **Adding Mutation:** In the next stage, Mutation was added to GWFA. But, from the experimenation, it can be seen that Mutation worsens the solutions for all the three cases. So, Mutation was never used after this point.

- **Adding SSU:** Then the Steady State Update (SSU) procedure was used in an attempt to modify the solutions. It made a huge difference in the performance of standard GWFA for all the three cases.

- **Adding Selection:** Getting inspired from the improvement in GWFA, in the next variant of GWFA, Tournament Selection was added to increase the selection pressure for the elite candidates. In GWFA, elite solutions are called DAs and they guide the RAs towards a better direction. Every RA randomly chooses a DA to follow. But this variant of GWFA selects a DA for each RA through Tournament Selection. From the results, it can be seen that Selection also helped the algorithm improve further.

- **Adding Parent Child Comparison:** Previously GWFA replaced the childern created from the parents directly without comparing their fitness values. Parent Child Comparison (PCC) preserves the children only if they are able surpass their parents in terms of fitness. But after adding PCC to GWFA, the quality of the solutionss degraded. So, the use of PCC was discarded for these problems.

- **Adding Parent Centric Crossover:** The next intuition was that converting the position update equation in GWFA with some other method may help the algorithm. So, Parent Centric Crossover (PCX) operator was used in place of the position update procedure. PCX uses three parents for each RA, namely: the RA itself, a selected DA and the average of all the better RAs than the present RA (known as LA in GWFA). The DA serves as the index parent and finally PCX returns a recombination of these three parents. PCX helped the algorithm to a large extent and finally this version of GWFA was able to get convergences for the Rosenbrock's function.

- **Using Single DA:** The next realization was that using a single global best solutions (DA) instead of multiple DAs maybe great for unimodal problems. This further improved the quality of the solutions.

- **Random LA Selection:** GWFA uses the average of all the better RAs for an RA as a source of guidance. But the problem with this approach is that the better RAs are **known** to have more fitness than the RA under consideration but the quality of their avergae is **uncertain**. So, the next modification of GWFA was to use a randomly selected RA (as LA) from the pool of better RAs, instead of taking their average as the guiding factor. It worked better than the previous approach.

3.2 Multimodal Problems

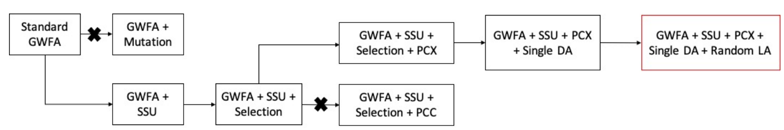

Figure 2: Process of Importing operations from G3-PCX to GWFA in an incremental fashion

3.3 Comparison with G3-PCX

The final variant of GWFA was also tested for the S_2 criterion and it has been compared with G3-PCX in Table 1. From the results, it can be observed that GWFA is able to achieve performance comparable to G3-PCX except for Rosenbrock's function which means that borrowing operators from G3-PCX has made GWFA somewhat algorithmically equivalent to G3-PCX and at the core, these two algorithms have become very similar in nature.

Criteria	$\mathbf{F_{elp}}$			$\mathbf{F_{sch}}$			$\mathbf{F_{ros}}$		
	Best	**Mean**	**Worst**	**Best**	**Mean**	**Worst**	**Best**	**Mean**	**Worst**
	Standard GWFA								
$\mathbf{S_1}$	78480	264011.2	469880	29940 (4)	171300	898620	15.63071727	17.68300715	18.75319299
	GWFA + Mutation								
$\mathbf{S_1}$	147640	294700	548920	31080 (1)	31080	31080	15.43421954	17.89707144	18.79622757
	GWFA + SSU								
$\mathbf{S_1}$	20440	48371.2	246760	29280 (29)	43080	879360	15.68396041	17.74399581	18.39445204
	GWFA + SSU + Selection								
$\mathbf{S_1}$	16760	20900	25040	26520 (47)	40320	818640	14.38492712	18.78694878	68.24623085
	GWFA + SSU + Selection + PCC								
$\mathbf{S_1}$	20720	25040	36920	34320 (35)	55380	956100	12.92479599	17.73948282	71.36594349
	GWFA + SSU + Selection + PCX								
$\mathbf{S_1}$	12160	17680	29640	33420	43080	59640	619720	697230	846960
	GWFA + SSU + Selection + PCX + Single DA								
$\mathbf{S_1}$	1832	2648	3056	3948	5164	5772	609500	733360	903970
	GWFA + SSU + Selection + PCX + Single DA + Random LA								
$\mathbf{S_1}$	1040	1460	1880	2160	2470	3090	19790	48170	735380
$\mathbf{S_2}$	6920	7550	8180	15180	16730	17970	609500 (47)	733360	903970
	G3-PCX								
$\mathbf{S_2}$	5744	6624	7372	14643	16326	17712	14847 (38)	22368	25797

Table 1: Experimental Outcomes for the incremental additions of EA operators to GWFA

4 Conclusion and Future Plan

In conclusion, it can be stated that starting from the standard version of GWFA till the end version, it has gone thorugh a lot of improvement by borrowing operators from several EA concepts of G3-PCX. In this way, we are able to learn some disadvantages and advantages of GWFA in comparison to the EAs. As an extension to this set of experimentation, the area of the applications can be extended to multi-modal functions as well.

References

[1] Kalyanmoy Deb and Nikhil Padhye. Improving a particle swarm optimization algorithm using an evolutionary algorithm framework. *KanGAL report*, 2010:003, 2010.

[2] Ritam Guha, Soulib Ghosh, Kushal Kanti Ghosh, and Ram Sarkar. Groundwater flow algorithm: A novel hydro-geology based optimization algorithm. 2020.

[3] David E Goldberg, Jon Richardson, et al. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Hillsdale, NJ: Lawrence Erlbaum, 1987.

[4] Xinjie Yu and Mitsuo Gen. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[5] Kalyanmoy Deb, Ashish Anand, and Dhiraj Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary computation*, 10(4):371–395, 2002.