Optimising the pool test method for COVID-19 using evolutionary algorithms

1st Lungu Cristian
Machine Intelligence Xetten SRL
Cluj-Napoca, Romania
lungu.cristian@gmail.com

2nd Prof. Dr. Ing. Rodica Potolea
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
rodica.potolea@cs.utcluj.ro

***Abstract*—The spread of the COVID-19 pandemic, quickly became a public health crisis which acted on many levels. The most challenging one of these was the sudden unavailability of protective gear and a complete lack of testing capacity.**

Although availability of masks and protective equipment has improved in the last few months, the testing capacity still remains a limited resource for most countries. One mitigation strategy for addressing the scarcity of tests is to pool biological samples in a single test, as demonstrated by the Frankfurt Goethe University.

In this paper we add to the body of knowledge on the problem of optimizing the pooled testing strategy by optimizing a multi-stage adaptive testing scenario using an evolutionary algorithm. We also propose a generic framework by which optimisations can be advanced even further and will help to massively increase the testing capacity for stopping the current pandemic.

***Index Terms*—optimisation, pandemic, genetic algorithm, machine learning**

I. Introduction

The current COVID-19 pandemic has been extensively written about and does not need any formal introduction. The initial stages of the outbreak was marked by a sudden unavailability of tests for detecting the infected patients. At the same time, the WHO Director-General famously exclaimed:

”We have a simple message for all countries: test, test, test.” [1]

The scope of this statement is to highlight that a crucial part of the public health response to this new threat is to rapidly diagnose and isolate infected individuals to prevent further spreading. Recent reports suggest that between 10% and 30% of SARS-CoV-2 infected patients are asymptomatic while both asymptomatic and presymptomatic subjects can spread the disease [2]. Therefore, increasing the testing capacity, is a key strategy for facing this public health emergency.

The main diagnostic test that has been implemented worldwide to confirm the infection by this novel coronavirus is the real-time reverse transcriptase-polymerase chain reaction (RT-PCR) from respiratory samples with satisfactory levels of sensibility and specificity [3].

A major bottleneck of managing the pandemic through diagnostic testing is the limited laboratory capabilities as well as limited access to genome-extraction and Polymerase Chain Reaction (PCR) reagents. Furthermore, these tests are primarily performed on symptomatic patients, hence, there is an urgent need to increase diagnostic testing capabilities in order to allow screening of asymptomatic populations which contribute to disease spread.

Recent experiments [4], [5] suggest that the group testing paradigm holds for SARSCoV-2, the virus that causes the disease COVID-19; that is, pools of samples with just one positive sample and many negative samples do indeed produce positive results.

In light of these, it becomes thus prudent to explore how to optimize the implementation of it in the healthcare setting. Therefore, the aim of this study is to provide an optimisation framework based on evolutionary algorithms for tuning *how* to best perform a pooled test.

II. Pooled testing

Research conducted at [4] and [5] resulted in the development of a procedure that makes it possible to increase worldwide testing capacities for detecting SARS-CoV-2 by combining multiple test samples in the same test.

It works by combining swab samples from mucous membranes of the throat or nose in a common buffer solution, and subsequently testing it using what is known as the PCR procedure (polymerase chain reaction procedure) [6].

When a test result is negative, all pooled samples have a reliable negative result. The pool testing has no influence on the detection limit, which means that (even) a single positive sample will always make the whole pool test positive as the result is independent on a detection threshold which might get diluted by using multiple samples.

Currently the maximum known number of samples in a single test seems to be 32 [7]–[9].

III. Types of strategies testing pooled testing

Before going into more details, we need to differentiate between two generic approaches regarding the way the problem is formulated. The first approach assumes all tests are designed in advance and can be carried out in parallel, named *nonadaptive testing* [10].

On the other hand, *adaptive testing* consists of approaches where each test result is examined before the next test pool is chosen (i.e. how you do the second stage of a test, depends on knowing the results on the first stage).

Note that for *nonadaptive testing*, a result of [11] suggests that the optimal strategy where we want zero-error probability is the individual testing one (where you test all patients one at a time). This implies that the path for making fewer tests than the overall number of patients lies in the *adaptive testing* domain.

At the same time, *adaptive testing* has the disadvantage that one must wait for the full chain of tests to finish in order to have a definite conclusion. This means that for a K-staged test, where the result of a stage is known in time t, the test results would be known after $K * t$ (i.e. the tests must be performed one at a time, and the results will be known after a long time). If the number K (which determines the total time for results to be known) is unbounded, the best known scheme is a generalized binary splitting scheme studied by [12].

Based on the above discussions, we have focused in this paper on optimising a *adaptive testing* strategy, with a small number of stages (K).

IV. Related work

Several approaches for optimizing the testing capacity through test pooling have been published recently.

The authors of [13] propose a mathematical model based on the binomial expression of pooling samples that have independent probabilities of the results. Thus, they calculate that on a *nonadaptive* setting, and using only a pooled step followed by individual test, the global optimum pool size, regardless of the number of subjects tested is 4, but only up to an infection rate of 0.3 after which their model breaks down.

In [14] the authors propose a two stage pooling strategy where the first stage consists of splitting the full population into r disjoint groups to be tested all at once, and then testing individually all the members of positive groups. This strategy is also known as *Dorfman's algorithm*.

Bernoulli selection is studied in [15], [16] where an individual is selected for inclusion in any given test of a K-staged strategy with a random uniform probability $\frac{1}{p*n}$. Note that when $K = 2$ they demonstrated that using this strategy the total number of tests required is

$$T \sim np(eln\frac{p}{1-p} + 1)$$

.

The *constant tests-per-item* strategy is studied in [17] where individuals are split in as many groups required so as each appears in exactly r number of groups. Each group is pool tested. They show that this strategy is optimal for *nonadaptive* settings where the number of infected individuals is fixed (but this, unfortunately, is not the case for COVID-19).

Other works such as [17]–[19] discuss the usage of *double constant* designs in the context of COVID-19 pooled testing, where each individual is tested r times in groups that are exactly s in size (i.e. designs having two constant parameters: the number of duplicated tests, the size of a group).

V. Methodology

As most of the relevant work in this area, we use a similar model for posing the pooling problem: the number of individuals n is large; the prevalence p is constant and represents the probability of an individual to be infected; we assume all infections are independent; we wish to reduce the average-case number of tests T; and we want to be certain that each individual is correctly classified (the 'zero-error' paradigm).

We will define a sequence of parametric *Commands* (discussed bellow) and define the term *Algo* to denote a fixed sequence of such *Commands* and their parameters.

Our aim is, given p and n, finding the best *Algo* that minimises T our target metric. We use a genetic algorithm [20] implementation for this purpose that is evaluated against 1000 Monte Carlo simulation [21] results.

A. Commands

Equal Split (ES) is a command that takes one parameter s and partitions all the individuals into disjoint groups of size s. Using only the **Equal Split** once, followed by the **Stop** command retrieves *Dorfman's algorithm*.

The **Shuffle** (SH) command takes one parameter, s, and denotes the operation of permutation of individuals still suspected to be positive at a certain time. Specifically it partitions the individuals into groups of size s and then permutes the groups themselves (when $s = 1$, this operation represents a plain permutation). If this command is positioned between two testing stages it provides a behavior similar to *constant tests-per item* strategy (although in a multi-step fashion) and allows one individual's results to be uncorrelated from its neighbours.

Stop (ST) is a command which marks the end of the *Algo* it is always interpreted as to execute a "test-all-remaining-individuals" that have not been excluded as negatives.

B. Genetic Algorithm

We optimize (minimize) the problem by means of a genetic algorithm (GA) [20] where the DNA is a binary encoding of the commands for a single *Algo*.

We have used a population size of 100 individuals in each generation. The fitness function of each individual is the estimated expected number of tests that such an individual will require, on average, to find all the positive individuals.

The evolution of the algorithm, from one generation to the next one uses the following strategies (we remind our readers that a GA has three main phases: *selection*, *crossover* and *mutation*):

For the *selection* phase we use:

- elitism selection [22] - where we select the top 10% of the individuals sorted by fitness score
- proportional selection [23] - where we select 20% of total required individuals by random sampling, and where each individual has a probability of being sampled proportional to its distance from the best fitness score from that generation
- tournament selection [24] - where we select the remaining 70% of individuals by building random pairs and taking the best individual from each

For the *crossover* step we we use a single-point crossover strategy [25].

For the *mutation* we use uniform mutation [26] with a mutation chance of 0.5% per bit.

C. Fitness function

A single individual is evaluated against 1000 Monte Carlo simulation [21] results.

The procedure is as follows:

- generate a *unit case* - a random n sized binary vector is generated with 1 elements (positive samples) randomly chosen with a p probability (and subsequently, 0 elements with $1-p$ probability).
- decode the *Algo* from the DNA of the current individual and execute it against the *unit case*
- compute the partial fitness as the number of tests needed to be done under the given context (unit case, *individual*)
- repeat this procedure 1000 times and aggregate all the fitness values by averaging them

D. On using brute-force

Since the total size of the DNA does not exceed 50 bits it might be conceivable that the best *Algo* can be found by mere brute-force. This is though unfeasible for the following reasons:

- The expected value (fitness value) of a single unique *Algo* (represented by a unique binary DNA encoding) requires 1000 (rather slow) simulations.
- We have limited this research to only three commands but we aim towards an extensible framework that can sustain in the future additional commands. This will make the DNA be larger than 50 bits.

VI. Results

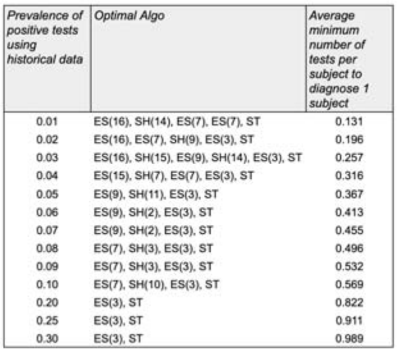
We have ran the optimisation on two settings, one *conservative* where the maximum pool size is 16 samples per group, and the other one *ambitious* where the maximum pool size is 32. As we have noted before [7]–[9] suggest that the maximum known number of samples in a single test seems to be 32 (thus the name *ambitious*).

In [17] the authors show that above

$$p \geq \frac{3-\sqrt{5}}{2} = 0.382$$

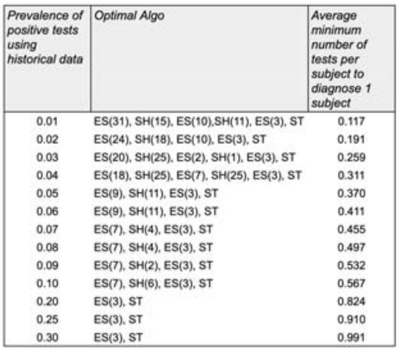
, the individual testing strategy is the best strategy possible, and this number is 0.307 for any two-stage strategy. As such, we have not tried to optimize above 0.3 given that all our *Algo*s will have at least two stages.

Figure 1 present the result of the optimisation for the *conservative* case. As stated before, the main usage of pooled testing strategies is mass testing for quickly finding and isolating the asymptomatic carriers (which are still infections). But since most patients exhibit symptoms in 3 to 5 days after their infection (at which point they can be identified clinically), we believe that the maximum feasible test-to-result interval is 24 hours. The *Shuffle* command is not bound to a PCR test so it can be considered instantaneous. Currently, a PCR result can be obtained in roughly six hours so we can afford a maximum of four testing stages (through the commands *Equal Splits* and *Stop*). As such, we have limited the number of *adaptive* testing steps (i.e. *Equal Splits*) to three. Note that all *Algo*s end in *Stop* which is a "test-all-remaining" individuals. This is required because we need to ensure that we have a "zero-error" result at the end.



Prevalence of positive tests using historical data	*Optimal Algo*	*Average minimum number of tests per subject to diagnose 1 subject*
0.01	ES(16), SH(14), ES(7), ES(7), ST	0.131
0.02	ES(16), ES(7), SH(9), ES(3), ST	0.196
0.03	ES(16), SH(15), ES(9), SH(14), ES(3), ST	0.257
0.04	ES(15), SH(7), ES(7), ES(3), ST	0.316
0.05	ES(9), SH(11), ES(3), ST	0.367
0.06	ES(9), SH(2), ES(3), ST	0.413
0.07	ES(9), SH(2), ES(3), ST	0.455
0.08	ES(7), SH(3), ES(3), ST	0.496
0.09	ES(7), SH(3), ES(3), ST	0.532
0.10	ES(7), SH(10), ES(3), ST	0.569
0.20	ES(3), ST	0.822
0.25	ES(3), ST	0.911
0.30	ES(3), ST	0.989

Fig. 1. Optimum *Algo* to use, given the prevalence p on a pool size of maximum **16** samples. The *Algo*s have been bound to a maximum of three adaptive test stages.



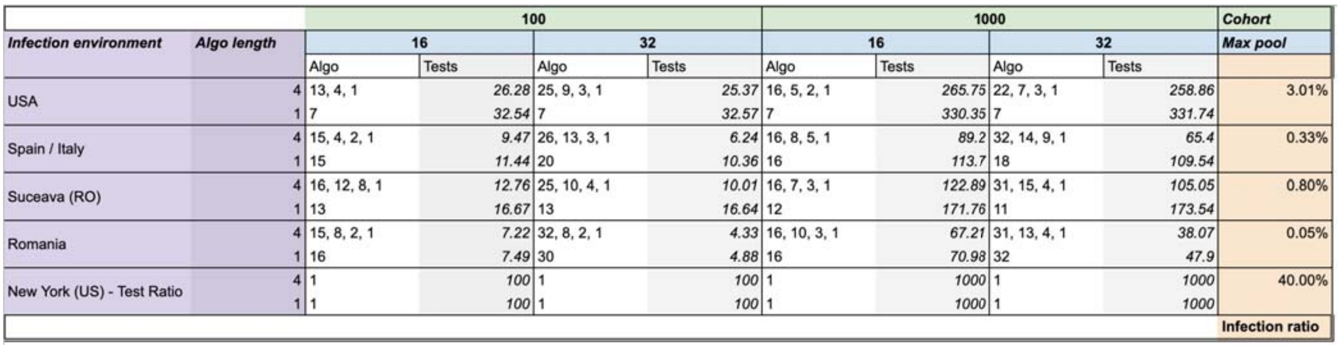
Prevalence of positive tests using historical data	*Optimal Algo*	*Average minimum number of tests per subject to diagnose 1 subject*
0.01	ES(31), SH(15), ES(10),SH(11), ES(3), ST	0.117
0.02	ES(24), SH(18), ES(10), ES(3), ST	0.191
0.03	ES(20), SH(25), ES(2), SH(1), ES(3), ST	0.259
0.04	ES(18), SH(25), ES(7), SH(25), ES(3), ST	0.311
0.05	ES(9), SH(11), ES(3), ST	0.370
0.06	ES(9), SH(11), ES(3), ST	0.411
0.07	ES(7), SH(4), ES(3), ST	0.455
0.08	ES(7), SH(4), ES(3), ST	0.497
0.09	ES(7), SH(2), ES(3), ST	0.532
0.10	ES(7), SH(6), ES(3), ST	0.567
0.20	ES(3), ST	0.824
0.25	ES(3), ST	0.910
0.30	ES(3), ST	0.991

Fig. 2. Optimum *Algo* to use, given the prevalence p on a pool size of maximum **32** samples. The *Algo*s have been bound to a maximum of test 3 adaptive test stages.

We can see that for an infection rate of 1% our strategy allows a 9x decrease in tests needed. Up to 10% infection rate all scenarios benefit from the *Shuffle* command which reinforces the claims of [17] on *constant test-per-item* being superior to *Dorfman's algorithm*. Between 20% and 30% infection rate all strategies are of type *Dorfman* although the gains are marginal. Our experiments also confirm the 30% infection limit above which individual testing should be the desired strategy to pursue.

Figure 2 present the result of the optimisation for the *ambitious* case. We note that the advantage of having a larger pool size is only beneficial up to an infection rate of $p \leq 2\%$ after which strategies on both settings largely converge to the same solutions.

In general, we also note that the *Shuffle* command should not be done on a per-group basis but at individual level, since strategies that only differ (between the two tables) by having different parameters of the *Shuffle* command, yield roughly



		100				1000				Cohort
Infection environment	***Algo length***	**16**		**32**		**16**		**32**		***Max pool***
		Algo	Tests	Algo	Tests	Algo	Tests	Algo	Tests	
USA	4	13, 4, 1	*26.28*	25, 9, 3, 1	*25.37*	16, 5, 2, 1	*265.75*	22, 7, 3, 1	*258.86*	3.01%
	1	7	*32.54*	7	*32.57*	7	*330.35*	7	*331.74*	
Spain / Italy	4	15, 4, 2, 1	*9.47*	26, 13, 3, 1	*6.24*	16, 8, 5, 1	*89.2*	32, 14, 9, 1	*65.4*	0.33%
	1	15	*11.44*	20	*10.36*	16	*113.7*	18	*109.54*	
Suceava (RO)	4	16, 12, 8, 1	*12.76*	25, 10, 4, 1	*10.01*	16, 7, 3, 1	*122.89*	31, 15, 4, 1	*105.05*	0.80%
	1	13	*16.67*	13	*16.64*	12	*171.76*	11	*173.54*	
Romania	4	15, 8, 2, 1	*7.22*	32, 8, 2, 1	*4.33*	16, 10, 3, 1	*67.21*	31, 13, 4, 1	*38.07*	0.05%
	1	16	*7.49*	30	*4.88*	16	*70.98*	32	*47.9*	
New York (US) - Test Ratio	4	1	*100*	1	*100*	1	*1000*	1	*1000*	40.00%
	1	1	*100*	1	*100*	1	*1000*	1	*1000*	
										Infection ratio

Fig. 3. Optimisation results when using only the *Equal split* command. The cell values show the size s of groups for each stage. Two max pools sizes are shown: 16, 32. The *Algo* has a limit on a maximum 4 testing stages. Infections rate date from May 2020.

the same outcome. This shows that the shuffling operation is important and not the size of the grouping before shuffling.

Figure 3 shows the results on real-world contexts where we have restricted (for simplicity) the *Shuffle* and *Stop* commands. It contains optimal strategies and expected outcomes for both 16 and 32 maximum pool sizes, as well as a 4 step strategy and a 2 step strategy for each context. Our 2 step strategies converge with the results presented by the authors of [13].

VII. Conclusions

In order to increase the capacity of COVID-19 testing we have published through this work, optimized *adaptive* testing strategies modeled as a sequence of parametric commands, specific to different known infection ratios. These commands were engineered as to allow modeling of most known strategies published up to this point with the hope of possibly finding better derivatives of these.

We have also optimized the problem on a few real-world contexts so that practitioners could adopt them immediately.

The results shown here (to the best of our knowledge) are the first to compute a (practically bounded) multi-stage pooled test scenario that goes beyond two testing stages while also providing solutions for the optimal length of the stages themselves. Previous research on *adaptive* pooled testing strategies in the COVID-19 context, was aimed at having a fixed number of either one or two stages. Fixing the number of stages leads to sub-optimal results when used on different infection ratios so optimising on this dimension as well is critical for obtaining better results. The biggest disadvantage of having multiple stages is a substantial increase in the time required for a testing strategy to be completed.

Our main contribution is modeling the problem as a sequence of steps that need to be optimized for the pooled testing problem and present an evolutionary framework by which this can be achieved. This opens the door to creating additional commands and allow for tuning the fitness function to also take into account other factors by which to optimize (e.g. total test time of strategy, complexity of operations, etc..). All of these should warrant further research.

References

[1] T. Adhanom, “Who director-general’s opening remarks at the media briefing on covid-19 - 16 march 2020,”

[2] R. Li, S. Pei, B. Chen, Y. Song, T. Zhang, W. Yang, and J. Shaman, “Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (sars-cov-2),” *Science*, vol. 368, no. 6490, pp. 489–493, 2020.

[3] V. M. Corman, O. Landt, M. Kaiser, R. Molenkamp, A. Meijer, D. K. Chu, T. Bleicker, S. Brünink, J. Schneider, M. L. Schmidt, *et al.*, “Detection of 2019 novel coronavirus (2019-ncov) by real-time rt-pcr,” *Eurosurveillance*, vol. 25, no. 3, p. 2000045, 2020.

[4] e. a. Professor Erhard Seifried, “The german red cross blood donor service in frankfurt,”

[5] e. a. Professor Sandra Ciesek, “The institute for medical virology at the university hospital frankfurt at goethe university,”

[6] R. K. Saiki, S. Scharf, F. Faloona, K. B. Mullis, G. T. Horn, H. A. Erlich, and N. Arnheim, “Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia,” *Science*, vol. 230, no. 4732, pp. 1350–1354, 1985.

[7] R. Ben-Ami, A. Klochendler, M. Seidel, T. Sido, O. Gurel-Gurevich, M. Yassour, E. Meshorer, G. Benedek, I. Fogel, E. Oiknine-Djian, A. Gertler, Z. Rotstein, B. Lavi, Y. Dor, D. G. Wolf, M. Salton, Y. Drier, and , “Large-scale implementation of pooled rna-extraction and rt-pcr for sars-cov-2 detection,” *medRxiv*, 2020.

[8] B. Abdalhamid, C. R. Bilder, E. L. McCutchen, S. H. Hinrichs, S. A. Koepsell, and P. C. Iwen, “Assessment of Specimen Pooling to Conserve SARS CoV-2 Testing Resources,” *American Journal of Clinical Pathology*, vol. 153, pp. 715–718, 04 2020.

[9] N. Shental, S. Levy, S. Skorniakov, V. Wuvshet, Y. Shemer-Avni, A. Porgador, and T. Hertz, “Efficient high throughput sars-cov-2 testing to detect asymptomatic carriers,” *medRxiv*, 2020.

[10] M. Aldridge, “Conservative two-stage group testing,” *arXiv preprint arXiv:2005.06617*, 2020.

[11] M. Aldridge, “Individual testing is optimal for nonadaptive group testing in the linear regime,” *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2058–2061, 2018.

[12] N. Zaman and N. Pippenger, “Asymptotic analysis of optimal nested group-testing procedures,” *Probability in the Engineering and Informational Sciences*, vol. 30, no. 4, pp. 547–552, 2016.

[13] D. Aragón-Caqueo, J. Fernández-Salinas, and D. Laroze, “Optimization of group size in pool testing strategy for sars-cov-2: A simple mathematical model,” *Journal of Medical Virology*, vol. n/a, no. n/a.

[14] R. Dorfman, “The detection of defective members of large populations,” *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[15] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama, “Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms,” in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1832–1839, IEEE, 2011.

[16] M. Aldridge, L. Baldassini, and O. Johnson, “Group testing algorithms: Bounds and simulations,” *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3671–3687, 2014.

[17] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Information-theoretic and algorithmic thresholds for group testing," *arXiv preprint arXiv:1902.02202*, 2019.

[18] R. Ben-Ami, A. Klochendler, M. Seidel, T. Sido, O. Gurel-Gurevich, M. Yassour, E. Meshorer, G. Benedek, I. Fogel, E. Oiknine-Djian, *et al.*, "Pooled rna extraction and pcr assay for efficient sars-cov-2 detection," *medRxiv*, 2020.

[19] A. Z. Broder and R. Kumar, "A note on double pooling tests," *arXiv preprint arXiv:2004.01684*, 2020.

[20] J. Sadeghi, S. Sadeghi, and S. T. A. Niaki, "Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: an improved particle swarm optimization algorithm," *Information Sciences*, vol. 272, pp. 126–144, 2014.

[21] H. Kahn and T. E. Harris, "Estimation of particle transmission by random sampling," *National Bureau of Standards applied mathematics series*, vol. 12, pp. 27–30, 1951.

[22] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, pp. 82–87, Ieee, 1994.

[23] J. Grefenstette, "Proportional selection and sampling algorithms," *Evolutionary computation*, vol. 1, pp. 172–180, 2000.

[24] B. L. Miller, D. E. Goldberg, *et al.*, "Genetic algorithms, tournament selection, and the effects of noise," *Complex systems*, vol. 9, no. 3, pp. 193–212, 1995.

[25] 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.

[26] R. Wardlaw and M. Sharif, "Evaluation of genetic algorithms for optimal reservoir system operation," *Journal of water resources planning and management*, vol. 125, no. 1, pp. 25–33, 1999.