ISSN: 2831-6789

Research Article

Journal of Robotics and Automation Research

An Improved Artificial Rabbit's Optimization (ARO) Algorithm for Identifying Mutated Driver Genes in Cancer

Lionel Alangeh Ngobesing and Yılmaz Atay*

Department of Computer Engineering, Engineering Faculty, Gazi University, Çankaya, Ankara, Türkiye

[*]Corresponding Author
Yılmaz Atay, Department of Computer Engineering, Engineering Faculty, Gazi University, Çankaya, Ankara, Türkiye.

Submitted: 2024, Jan 31; **Accepted**: 2024, Mar 26; **Published**: 2024, May 21

Citation: Ngobesing, L. A., Atay, Y. (2024). An Improved Artificial Rabbit's Optimization (ARO) Algorithm for Identifying Mutated Driver Genes in Cancer. *J Robot Auto Res, 5*(2), 01-12.

Abstract

The struggle of medics, computational biologist and experts in bioinformatics to find a cure for cancer is one of the most difficult problems in the world today. Given the large amounts of genomic data that is generated on a daily basis, it is becoming increasing difficult to evaluate and investigate this data. This is due to the fact that cancer data is heterogeneous, consisting of passenger genes which do not contribute to oncogenesis as well as driver genes which are directly led to oncogenesis. Hence identifying these driver genes from passenger genes in these large chunks of data increasingly becomes a difficult task. Considering the previous methods that have been developed to solve this problem, in this research we propose a bio-inspired method called Artificial Rabbit's Optimization (ARO) that integrates a mutation phase to be used to solve this problem of identifying cancer driver genes. This method merges the survival behavior of rabbits through exploration and exploitation to handle both global and local search respectively, with a gene interaction network to improve the accuracy of discovering cancer driver genes. The model is applied to 4 different types of cancers: breast cancer, brain cancer, prostate cancer and ovarian cancer. The results demonstrate that the proposed model can identify well-known labelled canonical driver genes while prioritizing them over unknown cancer driver genes. GBM found 9 genes, BRCA found 25 genes, OV found 4 genes and PRAD found 12 genes in the top 30 ranked genes as recognized by the NCG7.0.

Keywords: Cancer, Driver Genes, Artificial Rabbits Optimization, Metaheuristic, Biological Interactions

1. Introduction

The development of cancer is driven by the alteration or evolution in a cell's genetic make-up, otherwise known as mutation [1]. Oncogenesis occurs when tumor suppressor genes are set to inactive while activating oncogenes and Copy Number Aberrations (CNA) tend to be a great contributor to oncogenesis [2]. In the field of medicine, there are two major challenges involved in the study of oncogenesis. The first challenge is faced in the identification of molecular subtypes whereby patients are stratified clinically, with the aim of improving patient treatment as well as prognosis. The second challenge is in the discovery of cancer driver genes and mutations that are effective in cancer development. This is a challenge as cancer driver genes are shuffled up in passenger mutations that do not directly contribute to the development of cancer and also happen to exist in much larger numbers [3]. According to a study by The Cancer Genome Atlas (TCGA), a single cancer patient can have up to 100 different cancer mutations in their DNA and amongst these, only up to 6 actually are revealed to be cancer driver mutations while the rest are just passenger mutations with no effect on oncogenesis [4].

Approaches to identify cancer driver genes have been developed which are based on classification techniques, for example, Random Forests [5]. Unfortunately, it is difficult for these techniques to provide information about the interaction between the different regulatory systems involved when working with different datatypes. Computational and statistical approaches which are based on identifying patterns in driver gene groups or communities across a number of patients have also been developed for the purpose of identifying driver genes from passenger genes. Some of the methods discovered over the years include Dendrix, MEMo, RME and QuaDMutEx [6-9]. Using interaction networks for the identification of cancer driver genes has been a promising area of study for bio-scientists and computational biologists.

The reason for this success is that cancer mutations that carry out a particular function exists in groups that share similar biological properties [10]. For example, HotNet and HotNet2 were one of the first network-based cancer driver gene identification approaches to be developed whereby a propagation process is applied to diffuse mutation frequency score across the biological interaction network (such as a gene-gene interaction network) in order to discover significantly mutated cancer subnetworks. Another network-based method called NBS which is closely similar to HotNet, identifies cancer mutation subnetworks found different patients separately and then uses a consensus clustering framework to combine all of these subnetworks [11-13]. MUFFIN unlike the other methods, takes the impact of neighbors of mutated genes into consideration in order to prioritize cancer driver genes in the functional biological network [14].

Junrong Song et al. in 2018 proposed a method known as DyTidriver which aims at discovering cancer driver mutations through the use of variation frequency, tissue-specific expression and gene dysregulated expression on a human functional interaction network [15]. In this method, mutation genes were first of all selected with respect to the effect they have on their downstream genes. This is then followed by weighing the gene interactive network via its gene-to-gene co-expression and its inter-mutated gene relations. Mutated genes were then ranked as a result of merging variation frequency and the weighted graph. In 2019, Junrong Song, Wei Peng and Feng Wang discovered a novel method for driver gene identification which was based on random walk [16]. In this work, a bipartite graph, subcellular localization and mutation frequency we all integrated to improve driver mutation prediction performance.

The random walk algorithm was then implemented in order to efficiently combine the above-mentioned biological features. In this work, the following hypothesis was used: the assumption that driver mutations are identified through their appearance frequency, dysregulated genes and reliable relationships found between dysregulated genes and mutated genes in a range of patients. The results showed that driver mutations are the mutations which are more liable to affect more dysregulated genes while having a higher variation frequency in important clusters. Zexian Zeng et al. used deep learning for driver gene discovery in 2021 [17].

In this study, a Convolutional Neural Network (CNN) model was used for raw sequencing of tumor DNA. A deep learning model was used in this study due to the fact that these models were discovered to be more efficient in learning intricate patterns gotten from raw data as compared to the conventional models. [18-20]. Here, the CNNs shares parameters between regions in order to compute convolution on these regions. Hence, permitting smooth model training on large sequences of DNA. Applications of this model we also done in DanQ, DeepBind, DeepCpG and DeepSEA [21-24].

It is fact that meta-heuristic algorithms have become quite popular in solving optimization problems. The reason for this is that meta-heuristic algorithms are less expensive and also more efficient than the normally-used numerical methods. Meta-heuristic algorithms have a random nature which gives them an added advantage in successfully escaping local minima and exploring the entire search space. In cancer driver gene identification, some of the common meta-heuristic algorithms such as the Genetic Algorithm and Differential Evolution have been implemented [25, 26]. Even though there are have been many proposed algorithms, more algorithms are still being developed to solve optimization problems. The reason to this is because there doesn't exist an algorithm that performs best in solving all optimization problems, most algorithms are problem specific and developed to solve specific optimization problems.

Therefore, in this study, recently proposed metaheuristic algorithm known as the Artificial Rabbits Optimization algorithm (ARO) was implemented for the discovery of cancer driver genes [27]. This algorithm was further improved by implementing mutation such as that implemented in the Genetic Algorithm at every iteration. The mutation step was used to help generate more optimized results. The optimization process proposed here is generally divided into the exploration step and exploitation step. Exploration allows the algorithm to search for a new solution in the solution space found far away from the current solution while keeping the search extensive and global. Exploitation on the other hand aims to improve the current solution in its local neighborhood intensively. ARO is based on the mathematical modelling of the survival nature of rabbits. This nature is based on three search strategies which will be implemented here. These are: random hiding, detour foraging and energy shrinking strategy. In this research we first of all generate a bipartite graph as that implemented by DriverNet and BetweenNet [28, 29].

The ARO algorithm is then implemented on the generated bipartite graph in order to identify cancer driver mutations in accordance with the Network of Cancer Genes and Healthy Drivers (NCG) [30]. The experiments were carried out on 4 benchmarking cancer datasets from TCGA: Glioblastoma Multiforme (GBM) brain cancer dataset, Prostate Adenocarcinoma (PRAD) prostate cancer dataset, Breast Cancer (BRCA) dataset and Ovarian Cancer (OV) dataset. Gene Ontology (GO) analysis was then performed on the results in order to outline the biological significance of the discovered genes in biological processes in humans. The rest of this research is divided as follows. Section 2 discusses the materials and methods used in this study. This includes the mathematical model of the Artificial Rabbits Optimization algorithm, the datasets used, and the proposed model. Section 3 discusses the experimental analysis carried out and discusses the results. The results are compared with other state-of-the-art methods for detecting cancer driver genes developed in recent years. Section 4 is the discussion section which give a full summary of the entire study and perspectives for future challenges. Finally, Section 5 summaries and concludes the study.

2. Materials and Methods

2.1 Background

The use of an evolutionary method for cancer driver gene identification is crucial as evolutionary methods are better in maximizing the data search space while avoiding local maxima and/or minima and attaining global maxima and/or minima respectively. This is due to the fact that day by day the amount of available cancer data increasingly becomes overwhelming. Therefore, efficiently picking out this data is quite important. The ARO algorithm enhances search by accurately alternating between two important phases later on explained in this chapter. These phases are: exploration (detour foraging) and exploitation (random hiding).

2.2 Artificial Rabbits Optimization (ARO) algorithm General Idea

ARO is a bio-inspired algorithm derived from the strategies rabbits use in order to survive in nature. Rabbits are animals that feed on greens such as leafy weeds, grass and forbs (herbivores), and just like other evolutionary animals, they have to evolve with survival [31]. The survival strategy of rabbits is to eat grass which are far away from their own nests, hence preventing their nests from being discovered by predators. Given their wide arial vision, they are able to scan wide areas for food resources [32-33]. This strategy is known as “detour foraging” and will be regarded as the exploration mechanism in this study. Another strategy for survival used by rabbits is the random hiding strategy. Rabbits escape predators through the use of burrows. Rabbits dig many burrows and randomly choose one during a chase and uses it for shelter [34]. Because rabbits can easily stop and change direction while being chased at high speeds, this technique has been a very important strategy for survival. This random hiding technique is known here as exploitation. Rabbits being on the lower level of the food chain have a vast range of predators, meaning they have to be able to run really fast to escape danger. This affects the rabbit energy-wise, meaning that they have to change between random hiding and detour foraging adaptatively with respect to their energy levels.

2.3 Model and Algorithm

In ARO, detour foraging is implemented as the exploration strategy while random hiding is implemented as the exploitation strategy. Finally, as an energy shrinking strategy, rotation between random hiding and detour foraging is applied. The model is described as follows.

2.3.1 Detour Foraging (Exploration)

Detour foraging as mentioned earlier is a technique where the rabbits feed on food far away from their own nests. This is done by randomly selecting a location which is far away from the home location. Rabbits assume that every rabbit in the population has a nest with a given number of burrows, *d*. So, in detour foraging, the rabbits randomly update their position close to the nest of other rabbits in the population. This is demonstrated mathematically as follows:

$$\vec{v}_i\,(t+1) = \vec{x}_j\,(t) + R \cdot (\vec{x}_{\mathrm{i}}\,(t) - \vec{x}_j\,(t)) + round\,(0.5 \cdot (0.05 + r_1)) \cdot n_1 \qquad i, j = 1, \ldots, n \text{ and } j \neq i \tag{1}$$

$$R = \mathrm{L} * \mathrm{c} \tag{2}$$

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2}\right) * \sin(2\pi \mathrm{r}_2) \tag{3}$$

$$c(k) = \begin{cases} 1\ if\ \ k == g(l) \\ 0\ else \end{cases} \quad \mathrm{k} = 1, \ldots, d \text{ and } l = 1, \ldots, \lceil r3 \cdot d \rceil \tag{4}$$

$$g = randperm(d) \tag{5}$$

$$n1 \sim N(0, 1) \tag{6}$$

In the above equations, $v\vec{}i\ (t+1)$ represents the candidate position at time *t+1* of the ith rabbit, n is the population size, $x\vec{}j\ (t)$ represents the position of the rabbit at time *t*, d is the problem dimension, *T* is the number of iterations, *roundperm*(*d*) is a function that returns a random integer between 1 and *d, L* is the running length when carrying out detour foraging, r_1, r_2 and r_3 are random numbers between 0 and 1, n_1 is the subject to standard normal deviation.

Equations 1-6 are implemented to achieve detour foraging. **Equation 1** is used to demonstrate the random search of individuals in finding a food source. This significantly contributes to exploration and gives the ARO algorithm the ability to perform global search. **Equation 3** represents the running length, *L*, generated during iterations, which is typically longer during initial iterations and shorter during later iterations.

2.3.2 Random Hiding (Exploitation)

This is the situation whereby the rabbit has to dig a couple of burrows around its nest, in which it randomly chooses and hides when it is being chased by a predator. ARO implements random hiding by generating across the search space d number of burrows in each iteration. At every iteration, one of the burrows is chosen for hiding in order to increase probability of survival. In mathematically expressing exploitation, a hiding parameter, H, linearly drops from 1 to $1/T$, where T is the total number of iterations [35]. This implies that as the start of the iterations, the search space across which burrows are generated is quite large, and reduces as the iteration number increases. The mathematical implementation of random hiding is demonstrated below using **Equations 7-11.**

$$\mathrm{H} = \frac{T-t+1}{T} * \mathrm{r}_4 \tag{7}$$

$$\vec{v}(t+1) = \vec{x}_i(t) + R * (\mathrm{r}_4 * \vec{b}_{i,r}(t) - \vec{x}_i(t)) \text{ , i=1,…, n} \tag{8}$$

$$g_r(k) = \begin{cases} 1 & ifk == |r_5 \cdot d| \\ 0 & else \end{cases} \quad \mathrm{k} = 1, \text{….}, \mathrm{n} \tag{9}$$

$$\vec{b}_{i,r}(t) = \vec{x}_i(t) + H * g_r * \vec{x}_i(t) \tag{10}$$

$$\vec{x}_i(t+1) = \begin{cases} \vec{x}_i(t) & f(\vec{x}_i(t)) \leq f(\vec{v}(t+1)) \\ \vec{v}(t+1) & f(\vec{x}_i(t)) > f(\vec{v}(t+1)) \end{cases} \tag{11}$$

Equations 1 and 8 denote the new candidate positions generated for each rabbit. Here, $b^{\rightarrow}_{i,r}$ is a burrow selected at random from the total d number of burrows. r_4 and r_5 are two randomly selected number between 0 and 1. If the calculated fitness score of these positions is better than that of the current position, the rabbit's position will be updated with respect to this new candidate positions. Updating the position of the rabbit is shown by **Equation 11.**

2.3.3 Energy Shrink (Alternating between exploitation and exploration)

Rabbits tend to perform detour foraging when they have high energy levels. Later when energy levels drop, they will switch to random hiding. This application is the same in ARO. At the start of the iterations, exploration is performed and then switches to exploitation at later stages in the iteration. This brings about an energy factor in the current rabbit that is used to model the alternation between random hiding and detour foraging. This energy factor is given mathematically as follows on Equation 12. Where r is a random number between 0 and 1.

$$\mathrm{A(t)} = 4(1 - \frac{t}{T})\ln\frac{1}{r} \tag{12}$$

A large value of A(t) indicates that the rabbit has enough energy to perform exploration. On the other hand, a small value of A(t) indicates that the rabbit has less energy and has to go into hiding (exploitation). This is indicated in ARO by: exploration occurs when A(t) > 1 and exploitation when A(t) ≤ 1. Wang L et al. [27] demonstrate the behavior of the energy factor over 1000 iterations as shown on **Figure 1** below. This shows how the value of the energy factor decreases as the number of iterations increase.

2.3.4 Mutation

ARO in this study is improved by implementing mutation just before detour foraging and random hiding. Mutation here is implemented in order to variate the search space and foster evolution, hence prevent the algorithm from converging and eventually coming to a halt. To implement mutation, at every iteration, individuals in the current solution are randomly chosen and given new randomly selected positions. Exploration and exploitation are then carried out on the output solution gotten from mutation.

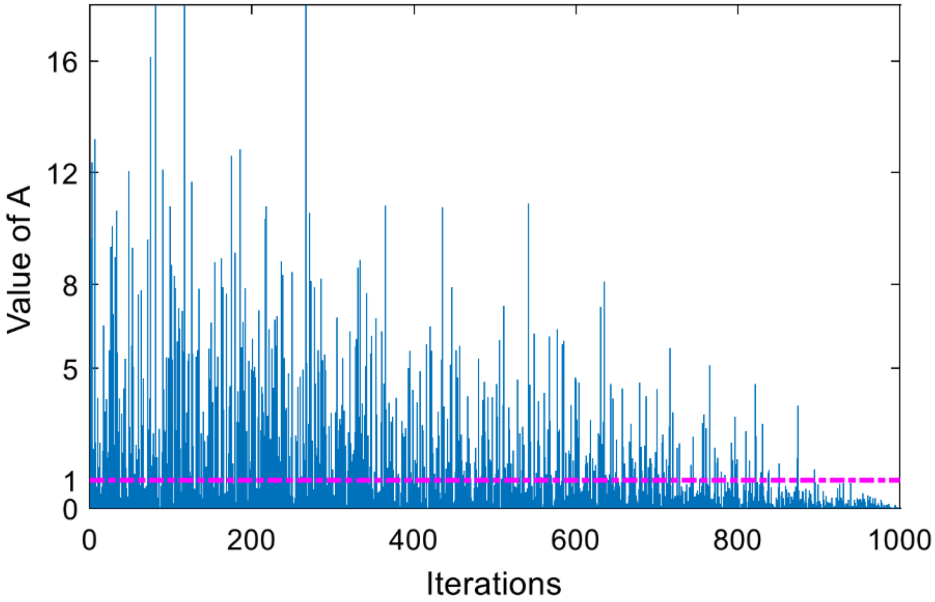

Figure 1: Behavior of the energy factor, A, over 1000 iterations.

2.3.5 The Algorithm

The improved ARO algorithm implemented in this research is as follows: First, a random solution of X rabbits is initialized. The fitness of this initial rabbit is calculated and labelled as the best solution X_{best}. The iteration parameter is then set as the terminating criteria. In the iteration loop, mutation is first of all applied to the current selected population of X rabbits. For each individual in this population, the energy factor of the rabbit is calculated. If the energy factor, A, is greater than 1, then a new rabbit is randomly chosen from the neighborhood population, it fitness calculated and position updated using **Equation 11.** If the energy factor is less than or equal to 1, a number of burrows, d, is generated, random hiding implemented and the position of the current selected rabbit is updated using Equation 11. These steps are repeated until the stopping criteria for both the inner and outer loops are met, after which the update best solution is taken as the optimized solution. The flowchart for the ARO algorithm is demonstrated on **Figure 2.** The algorithm is simplified in the following pseudocode in **Table 1**

Pseudocode for ARO

```
➢ Begin
➢ Initialize a random solution of rabbits, X
➢ Calculate their fitness, $X_{best}$
➢ Start while ($t <$ iteration count)
    • Implement mutation on the current solution
    • Begin for loop
        ○ For each solution, X, calculate the energy factor, A, using Equation 12
        ○ If ($A > 1$),
            ✓ Randomly choose new rabbit from neighborhood
            ✓ Calculate R using Equations 2-6, perform detour foraging using equation 1,
            ✓ Calculate the fitness and update the position of the rabbit using Equation 11
        ○ Else If ($A \leq 1$),
            ✓ Generate d number of burrows and choose one of the burrows at random
              which will be used for hiding (Equation 10)
            ✓ Use Equation 8 to perform random hiding then calculate the fitness of the
              solution.
            ✓ Finally, update the position of the rabbit using Equation 11
        ○ End if
    • End for loop.
➢ End while loop
➢ Update the solution to get the new best solution $X_{best}$
➢ End
```

Table 1: Pseudocode for Artificial Rabbits Optimization.

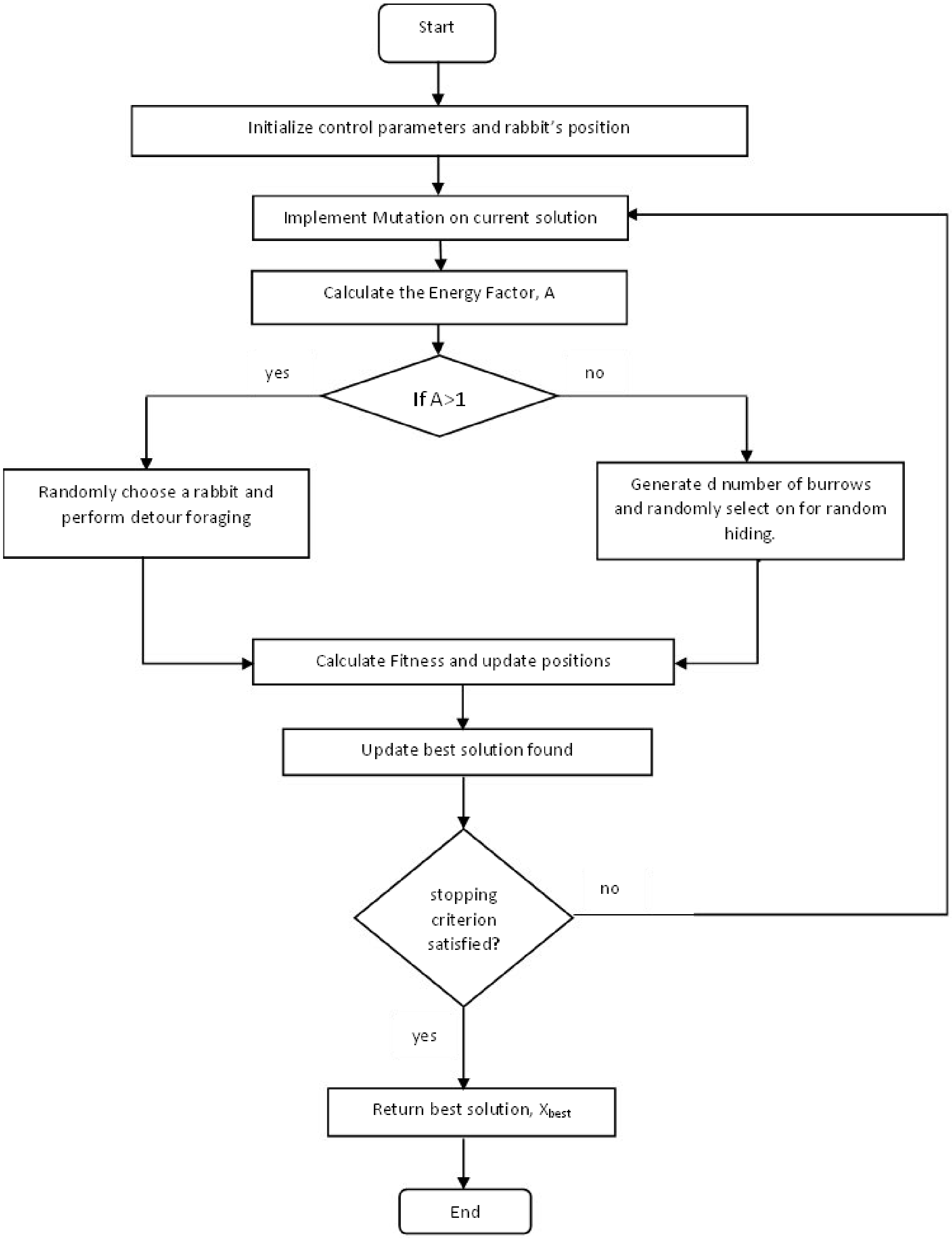
Start
Initialize control parameters and rabbit's position
Implement Mutation on current solution
Calculate the Energy Factor, A
yes
If A>1
no
Randomly choose a rabbit and perform detour foraging
Generate d number of burrows and randomly select on for random hiding.
Calculate Fitness and update positions
Update best solution found
stopping criterion satisfied?
no
yes
Return best solution, Xbest
End

Figure 2: Flow chart of Artificial Rabbits Optimization (ARO) algorithm.

3. Results

3.1 Datasets

In this research, 4 different datasets were used to carry out experimental analysis. These datasets were gotten from of The Cancer Genome Atlas (TCGA) datasets gotten from the cBioPortal for Cancer genomics [36]. **Table 2** below shows the demographics of the different datasets with respect to the number of patients involved, the number of genes, number of mutations, CNA and RNA-Seq. The first dataset used for experiments in this research was made up of copy number alteration (CNA), mRNA sequencing expression and gene mutation data for 585 patients with brain cancer. This dataset is called the Glioblastoma Multiforme (GBM). The next dataset was for patients with ovarian cancer. This dataset was made up of copy number alterations, gene mutations and mRNA sequencing expression data for 585 patients. It is called the TCGA Ovarian Serous Cystadenocarcinoma (OV). The third dataset was the TCGA Prostate Adenocarcinoma (PRAD) dataset. This dataset was made up of gene expression data, gene mutations and copy number alteration data for 334 patients. The fourth dataset was the TCGA Breast Cancer (BRCA) dataset which was made up of mRNA-seq expression data, gene expression data and CNA data for 112 patients. The biological interaction network used here was a gene-gene interaction influence graph gotten from REACTOME database, consisting of over of 518,302 genes.

In this study, experiments were carried out on 4 benchmarking datasets from TCGA as mentioned above in section 2. These were GBM, BRCA, OV and PRAD datasets. The implementation was carried out in a python environment with number of iterations being the input parameter. The fitness is calculated using the *Schwefel 2.22* function, which is an unconstraint test benchmark function [37]. The Top 30 ranked genes discovered by this algorithm for each dataset are represented on **Table 3.** Benchmarking performance analysis was then carried out on the results and compared with the results of 4 state-of-the-art models: HotNet2 Dendrix, DriverNet and QuaDMutNetEx [6, 9, 28, 38]. The analysis was done using information of genes labelled as canonical cancer driver genes by the Network of Cancer Genes and Healthy Drivers (NCG7.0) data repository [30].

Dataset	No of Patients	No of genes	Mutations	CNA	RNA-Seq
PRAD	334	34,192	333	333	290
OV	585	53,204	523	572	300
GBM	585	68,802	397	575	160
BRCA	112	17,272	112	112	112

Table 2: TCGA datasets from the cBioPortal

3.2 Performance Benchmarking Analysis

In the benchmarking performance analysis of the glioblastoma multiforme brain cancer dataset (GBM) which consist of 68,802 genes, a total of 15 canonical genes were identified out of the total 24 canonical genes presented by NCG7.0. of these 15 genes, 9 were ranked amongst the top 30 genes by the ARO algorithm. These genes include: TP53, PTEN, PIK3R1, PIK3CA, RB1, NF1, PDGFRA, PTPN11 and STAG2. For the triple negative breast cancer dataset (BRCA), out of the total 110 canonical driver genes, a total of 55 BRCA driver genes were identified in this study. From the 55 identified driver genes, 25 were ranked in the top 30 genes by ARO. For ovarian cancer (OV), out of the total of 11 canonical cancer genes, a total of 6 were identified in this study, with 4 gene (TP53, BRCA1, RB1, FAT3) ranking in the top 30. Finally for the prostate adenocarcinoma (PRAD) dataset, out of the total 43 canonical driver genes, 19 were identified in experiments with ARO, where 12 of these genes rank in the top 30 genes. **Table 4** shows the cancer driver genes identified by ARO in this study.

Rank	GBM	BRCA	OV	PRAD
1	TP53	TP53	TP53	TP53
2	PTEN	PIK3CA	EP300	CTNNB1
3	PIK3R1	EP300	UBC	EP300
4	PIK3CA	AKT1	TTN	FOXA1
5	RB1	CDH1	DYNC1H1	PIK3CA
6	EB300	KMT2C	EGFR	ATM
7	NF1	GATA3	PIK3CA	PTEN
8	PLCG2	NCOR2	BRCA1	TTN
9	PRKACA	PIK3R1	RB1	STAT3
10	UBC	MAP3K1	PRKCB	CREBBP
11	PDGFRA	NOTCH1	PIK3CB	HSPA8
12	PTPN11	ERBB3	NCOA3	DYNC1H1
13	PIK3CB	NCOR1	DCTN1	SPOP
14	CREBBP	ERBB2	TAF1	SPTA1
15	SP1	NF1	LRRK2	HRAS
16	SPTA1	CBFB	SRC	PRKACA
17	LRP2	EGFR	PRKACB	KMT20
18	APOB	ARID1A	LRP2	FAT3
19	PIK3CG	BRCA1	STAT3	NCOR1
20	DYNC1I1	CTCF	SP1	EGFR
21	KDR	SMAD4	FAT3	PIK3CD

22	MUC16	NCOA3	HTT	RELA
23	ITGB2	JAK1	NCOR2	MUC16
24	JUN	RYNX1	PDGFRA	APC
25	SRC	ATR	ATM	POLR2B
26	PRKCB	AKAP9	POLR2A	HTT
27	STAG2	ERBB4	NFKB1	KMT2C
28	ARRB1	TAF1	ALMS1	PLCB4
29	ESR1	SMARCC2	GNAL	SMAD4
30	DYNC1H1	HERC2	PCDH15	ANK2

Table 3: Top 30 ranked genes discovered by ARO from experiments on GBM, BRCA, PRAD and OV.

Dataset	Canonical Driver Genes
BRCA	TP53 PIK3CA EP300 AKT1 CDH1 KMT2C GATA3 PIK3R1 MAP3K1 NOTCH1 ERBB3 NCOR1 ERBB2 NF1 CBFB EGFR ARID1A BRCA1 CTCF SMAD4 ATR RB1 RUNX1 PTEN
OV	TP53 BRCA1 RB1 FAT3
GBM	TP53 PTEN PIK3R1 PIK3CA RB1 NF1 PDGFRA PTPN11 STAG2
PRAD	TP53 CTNNB1 FOXA1 PIK3CA ATM PTEN SPOP HRAS KMT2D NCOR1 APC KMT2C

Table 4: Driver genes discovered by ARO ranked amongst top 30.

3.3 Performance Evaluation

3.3.1 Exploration and Exploitation Analysis

Artificial Rabbits Optimization tends to have an accelerated convergence at during its iterations. Therefore, at initial iterations, the search individuals can identify promising regions and then speed up the rate of convergence. ARO depicts an effective performance due to the fact that the global search mechanism used is integrated to effectively enhance the exploration level. This makes ARO very competitive in exploration the search space for the best solution in a multimodal function. By balancing exploitation and exploration, ARO and evidently avoid local optima. Furthermore, the ratio of exploration to exploitation is defined by a trade-off between two searches as shown by Hussain K. et al. [39].

3.3.2 Performance Analysis

The performance of the algorithm was done by calculating the precision, recall, f-score and accuracy of the results. Here, true positives are represented as TP, true-negative as TN, false-positive represented as FP, and false-negative represented as FN. Table 5 below shows the number of genes labelled as canonical genes by the NCG7.0 for the different datasets used in this research.

Dataset	Number of Canonical Genes
GBM	24
BRCA	110
OV	11
PRAD	43

Table 5: Canonical driver gene count in the NCG7.0 for datasets

Precision, recall and the f-score are calculated as follows:

$$Precision = \frac{TP}{TP+FP} \tag{13}$$

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

$$F - score = \frac{Precision\ x\ Recall}{Precision+Recall} \tag{15}$$

Where, TP is the total number of genes discovered by the algorithm that have also been labelled as canonical driver genes by the NCG7.0.

TN is the number of genes that have not been labelled by NCG7.0 as canonical driver gene and also have not been ranked amongst the top N genes by the algorithm. FP is the total number of genes identified by our algorithm but have not been labelled by the NCG7.0 as canonical driver genes. FN is the total number of genes which have not been discovered by the algorithm but have been labelled as driver genes by the NCG7.0 database.

Table 6 shows the comparison of results of ARO with HotNet2, Dendrix, DriverNet and QuaDMutNetEx. The table depicts the genes found in the solution for each algorithm, the canonical genes identified, the True Positives and False Positives for each of the datasets.

4. Discussion

The proposed algorithm in this research is an improved version of the bio-inspired Artificial Rabbit's Optimization algorithm proposed by Wang L et al. in 2022 [27]. This algorithm was improved by adding a mutation phase before implementing the detour foraging and random hiding phases of the ARO. The model in this study is as follows. First, using the cancer data for the respective cancer type gotten from the TCGA data repository and a biological interaction network similar to that presented by Sarkar A et al., a bipartite graph was generated as that presented in DriverNet and QuaDMutNetEx [40]. ARO was then applied on this bipartite graph to identify cancer driver genes. This improved ARO algorithm is implemented as thus. A random solution is initialized in the neighborhood space. Mutation was then applied on this solution to generate a more diverse solution that covers the search space.

Dataset	Algorithm	Genes in Solution	Canonical Genes	TP	FP	f-score
OV	ARO	11	6	6	5	0.545
	HotNet2	11	4	4	7	0.363
	DriverNet	-	-	-	-	-
	Dendrix	11	3	3	8	0.273
	QuaDMutNetEx	**11**	**7**	**7**	**5**	**0.636**
GBM	**ARO**	**24**	**11**	**11**	**13**	**0.458**
	HotNet2	**24**	**11**	**11**	**26**	**0.458**
	DriverNet	24	9	9	8	0.375
	Dendrix	24	4	4	20	0.167
	QuaDMutNetEx	24	6	6	18	0.250
PRAD	**ARO**	**43**	**19**	**19**	**24**	**0.442**
	HotNet2	43	9	9	34	0.209
	DriverNet	43	14	14	32	0.326
	Dendrix	43	8	8	35	0.186
	QuaDMutNetEx	43	13	13	30	0.302
BRCA	**ARO**	**110**	**55**	**55**	**55**	**0.500**
	HotNet2	110	18	18	92	0.164
	DriverNet	110	33	33	77	0.300
	Dendrix	110	16	16	94	0.145
	QuaDMutNetEx	110	25	25	85	0.227

Table 6: Comparison of canonical genes with HotNet2, DriverNet, Dendrix and QuaDMutNetEx.

Next, the energy factor, A, was calculated to determine whether detour foraging or random hiding will be implemented on the current rabbit solution. This is the energy shrinking mechanism which is used by rabbits for survival. If the energy factor is greater than 1, then detour foraging (exploration) is implemented to generate the new position of the rabbit. If the energy factor is less than or equal to 1, then random hiding is implemented to generate the new position of the current rabbit. At every iteration, the fitness of the current solution was calculated by using the Schwefel 2.22 function and compared with that of the candidate solution in order to select the new current solution. The best solution was then gotten at the end of the iterations. Performance analysis was then applied to evaluate the results of this model. Performance analysis was carried out on 4 benchmarking datasets: Glioblastoma Multiforme (GBM) brain cancer dataset, Prostate Adenocarcinoma (PRAD) prostate cancer dataset, Breast Cancer (BRCA) dataset and Ovarian Cancer (OV) dataset. Comparison on results was performed with 4 other state-of-the-art models used for identifying cancer driver mutations: Dendrix, DriverNet, HotNet2 and QuaDMutNetEx. ARO proposed in this study proved to be quite efficient in identifying cancer driver mutations in comparison with these methods. ARO was able to identify the following driver genes labelled as canonical genes by the NCG7.0. For GBM: TP53, PTEN, PIK3R1, PIK3CA, RB1, NF1, PDGFRA, PTPN11 and STAG2 were identified among the top 30 ranked; for OV: TP53, BRCA1, RB1 and FAT3 were identified among the top 30 ranked; for PRAD: TP53, CTNNB1, FOXA1, PIK3CA, ATM, PTEN, SPOP, HRAS, KMT2D, NCOR1, APC and KMT2C were identified among the top 30 ranked; for BRCA: TP53, PIK3CA, EP300, AKT1, CDH1, KMT2C, GATA3, PIK3R1, MAP3K1, NOTCH1, ERBB3, NCOR1, ERBB2, NF1, CBFB, EGFR, ARID1A, BRCA1, CTCF, SMAD4, ATR, RB1, RUNX1 and PTEN were identified among the top 30 ranked.

There are several aspects to why ARO serves are an appropriate algorithm for this study. In ARO, the detour foraging helps in accomplishing global search, while random hiding helps in accomplishing local search. There parameter, R, presented in Equation 2 could be adaptatively adjusted as the number of iterations increase, in order to foster the gradual alternation from exploration to exploitation. Approximately half of the iterations are assigned to exploitation (A ≤ 1) and the other half assigned to exploration (A > 1), as an outcome of the energy factor, A. This energy factor is a time dependent factor effectively switches between exploitation and exploration as well as enhances these phases. Furthermore, ARO has a good ability for bearing fault diagnosis gotten from its ability to balance exploitative and explorative search. The computation complexity of ARO is linear and given by Equation 16 below:

Despite having a superior performance, ARO also has a couple of shortcomings. ARO lacks multiple search mechanisms during optimization of problems that have variable types of certainty. Also, ARO faces shortcomings in handling unimodal problems that have multiple extrema. Hence ARO is less efficient in solving NP-Hard problems. These problems could be solved by implementing multi-objective and binary versions of the algorithm.

$$O(ARO) = O(1+ n + Tn + 0.5{*}Tnd + 0.5\ Tnd) = O(Tnd + Tn + n). \quad \ldots\ldots\ldots\ldots \quad (16)$$

5. Conclusion

This study implements a modified version of a newly developed bio-inspired optimizer which mimics the natural behavior of rabbits. The optimizer models the strategies used by rabbits for survival, through random hiding and detour foraging. This model, known as Artificial Rabbits Optimization, was developed to accurately determine global optima for multimodal, unimodal and composite functions while performing both local and global search. In this study, we use make use of this algorithms property to identify known cancer driver mutations found in cancer patients. The Artificial Rabbits Optimization algorithm is significantly important in tackling engineering problems that have constrained as well as unknown search spaces. With that respect, it is important to mention that this algorithm properly serves cancer driver mutation identification as the amount of available cancer data in the world increases day by day. That is, the search space for cancer driver genes is continuously changing. To highlight the efficiency of the algorithm in this context, experiments are carried out in this research on 4 different cancer types: Breast cancer, ovarian cancer, brain cancer and prostate cancer. Even though this algorithm shows some shortcomings in finding multiple search mechanisms for exploration, it still portrays outstanding performance in achieving efficient results.

Declaration

Ethical Approval and Consent to Participate

Not Applicable.

Consent to publication

Not Applicable.

Data Availability Statement

Data, Codes and Materials are available privately now at https://github.com/Alangeh/Improved-ARO.

Conflict of Interest

Authors declare that there was no conflict interest.

Funding

No funding was gotten for this research.

Acknowledgements

Not Applicable.

Authors Contributions

Both authors of this paper were actively involved in the development of the research study. Mr. Lionel was mainly focused on developing and implementing the code for the proposed algorithm, gathering and analyzing the results, while Dr. Yilmaz was behind the logic of the algorithm, better shaping it to provide the best results.

References

1. Hanahan, D., & Weinberg, R. A. (2011). Hallmarks of cancer: the next generation. *cell, 144*(5), 646-674.
2. Zack, T. I., Schumacher, S. E., Carter, S. L., Cherniack, A. D., Saksena, G., Tabak, B., ... & Beroukhim, R. (2013). Pan-cancer patterns of somatic copy number alteration. *Nature genetics, 45*(10), 1134-1140.
3. Stratton, M. R., Campbell, P. J., & Futreal, P. A. The cancer genome. Nature [Internet]. 2009; 458 (7239): 719–24.
4. Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R., Ozenberger, B. A., Ellrott, K., ... & Stuart, J. M. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics, 45*(10), 1113-1120.
5. List, M., Hauschild, A. C., Tan, Q., Kruse, T. A., Baumbach, J., & Batra, R. (2014). Classification of breast cancer subtypes by combining gene expression and DNA methylation data. *Journal of integrative bioinformatics, 11*(2), 1-14.
6. Vandin, F., Upfal, E., & Raphael, B. J. (2012). De novo discovery of mutated driver pathways in cancer. *Genome research, 22*(2), 375-385.
7. Ciriello, G., Cerami, E., Sander, C., & Schultz, N. (2012). Mutual exclusivity analysis identifies oncogenic network modules. *Genome research, 22*(2), 398-406.
8. Miller, C. A., Settle, S. H., Sulman, E. P., Aldape, K. D., & Milosavljevic, A. (2011). Discovering functional modules by identifying recurrent and mutually exclusive mutational patterns in tumors. *BMC medical genomics, 4*, 1-11.
9. Bokhari, Y., Alhareeri, A., & Arodz, T. (2020). QuaDMutNetEx: a method for detecting cancer driver genes with low mutation frequency. *BMC bioinformatics, 21*, 1-12.
10. Wei, P. J., Zhang, D., Xia, J., & Zheng, C. H. (2016). LNDriver: identifying driver genes by integrating mutation and expression data based on gene-gene interaction network. *BMC bioinformatics, 17*, 221-230.
11. Vandin, F., Upfal, E., & Raphael, B. J. (2011). Algorithms for detecting significantly mutated pathways in cancer. *Journal of Computational Biology, 18*(3), 507-522.

12. Leiserson, M. D., Vandin, F., Wu, H. T., Dobson, J. R., Eldridge, J. V., Thomas, J. L., ... & Raphael, B. J. (2015). Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature genetics, 47*(2), 106-114.
13. Hofree, M., Shen, J. P., Carter, H., Gross, A., & Ideker, T. (2013). Network-based stratification of tumor mutations. *Nature methods, 10*(11), 1108-1115.
14. Cho, A., Shim, J. E., Kim, E., Supek, F., Lehner, B., & Lee, I. (2016). MUFFINN: cancer gene discovery via network analysis of somatic mutation data. *Genome biology, 17,* 1-16.
15. Song, J., Peng, W., Wang, F., & Wang, J. (2019). Identifying driver genes involving gene dysregulated expression, tissue-specific expression and gene-gene network. *BMC medical genomics, 12,* 1-12.
16. Song, J., Peng, W., & Wang, F. (2019). A random walk-based method to identify driver genes by integrating the subcellular localization and variation frequency into bipartite graph. *BMC bioinformatics, 20*, 1-17.
17. Zeng, Z., Mao, C., Vo, A., Li, X., Nugent, J. O., Khan, S. A., ... & Luo, Y. (2021). Deep learning for cancer type classification and driver gene identification. *BMC bioinformatics, 22,* 1-13.
18. Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular pharmaceutics, 13*(5), 1445-1454.
19. Mao, C., Yao, L., Pan, Y., Luo, Y., & Zeng, Z. (2018, December). Deep generative classifiers for thoracic disease diagnosis with chest x-ray images. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1209-1214). IEEE.
20. Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular systems biology, 12*(7), 878.
21. Quang, D., & Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic acids research, 44*(11), e107-e107.
22. Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology, 33*(8), 831-838.
23. Angermueller, C., Lee, H. J., Reik, W., & Stegle, O. (2017). DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome biology, 18,* 1-13.
24. Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods, 12*(10), 931-934.
25. Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press.
26. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11,* 341-359.
27. Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 114,* 105082.
28. Bashashati, A., Haffari, G., Ding, J., Ha, G., Lui, K., Rosner, J., ... & Shah, S. P. (2012). DriverNet: uncovering the impact of somatic driver mutations on transcriptional networks in cancer. *Genome biology, 13*, 1-14.
29. Erten, C., Houdjedj, A., & Kazan, H. (2021). Ranking cancer drivers via betweenness-based outlier detection and random walks. *BMC bioinformatics, 22*, 1-16.
30. Repana, D., Nulsen, J., Dressler, L., Bortolomeazzi, M., Venkata, S. K., Tourna, A., ... & Ciccarelli, F. D. (2019). The Network of Cancer Genes (NCG): a comprehensive catalogue of known and candidate cancer genes from cancer sequencing screens. *Genome biology, 20,* 1-12.
31. Tůmová, E., Martinec, M., & Chodová, D. (2011). Analysis of Czech rabbit genetic resources. *Scientia agriculturae bohemica, 42*(3), 113-118.
32. Juan, Q. (2017). Rabbits do not eat grass around the nest. Knowl. Window.13, 39.
33. Tynes, V. V. (Ed.). (2010). *Behavior of exotic pets*. John Wiley & Sons.
34. Camp, M. J., Rachlow, J. L., Shipley, L. A., Johnson, T. R., & Bockting, K. D. (2014). Grazing in sagebrush rangelands in western North America: implications for habitat quality for a sagebrush specialist, the pygmy rabbit. *The Rangeland Journal, 36*(2), 151-159.
35. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software, 95,* 51-67.
36. cBioPortal for Cancer genomics. 24th May 2022, http://www.cbioportal.org.
37. Trivedi, I. N., Pradeep, J., Narottam, J., Arvind, K., & Dilip, L. (2016). Novel adaptive whale optimization algorithm for global optimization. *Indian Journal of Science and Technology.*
38. Leiserson, Vandin, F., Wu, HT., Dobson, JR., Eldridge, JV., Thomas, JL., (2018) HotNet2.
39. Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications, 31*(11), 7665-7683.
40. Sarkar, A., Atay, Y., Erickson, A. L., Arisi, I., Saltini, C., & Kahveci, T. (2019). An efficient algorithm for identifying mutated subnetworks associated with survival in cancer. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 17*(5), 1582-1594.