

BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

MASTER'S THESIS

Brno, 2023

Bc. Paweł Santarius



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

EVOLUTIONARY ALGORITHMS IN PARAMETRIC ESTIMATION FOR NONLINEAR REGRESSION MODELS

EVOLUČNÍ ALGORITMY PRO ODHAD PARAMETRŮ MODELŮ NELINEÁRNÍ REGRESE

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Bc. Paweł Santarius

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. Martin Mézl, Ph.D.

BRNO 2023

Master's Thesis

Master's study program **Bioengineering**

Department of Biomedical Engineering

Student: Bc. Paweł Santarius

ID: 203202

**Year of
study:** 2

Academic year: 2022/23

TITLE OF THESIS:

Evolutionary algorithms in parametric estimation for nonlinear regression models

INSTRUCTION:

1) Study principles of nonlinear regression and using of evolutionary algorithms for estimation of unknown models' parameters. 2) Make a review of mathematical models used in perfusion imaging by ultrasound. 3) Make a curve-fitting task with using proper model with using simulated or phatnotm datasets. Try at least two evolutionary algorithms implemented in Matlab or Python. 4) Select and implement at least three evolutionary algorithm which can't be found in freely published packages. 5) Test these algorithms with using phatnom and clinical datasets and compare your algorithms with conventional algorithms in available packages. 6) Discuss results with respect to reliability of fits.

RECOMMENDED LITERATURE:

[1] GULSEN, M., A. E. SMITH a D. M. TATE. A genetic algorithm approach to curve fitting. International Journal of Production Research. 2007, 33(7), 1911-1923. ISSN 0020-7543. DOI:10.1080/00207549508904789

[2] MACIEL, L., F. GOMIDE a R. BALLINI. A differential evolution algorithm for yield curve estimation.

Mathematics and Computers in Simulation. 2016, 129, 10-30. ISSN 03784754.

DOI:10.1016/j.matcom.2016.04.004

**Date of project
specification:** 6.2.2023

**Deadline for
submission:** 22.5.2023

Supervisor: Ing. Martin Mézl, Ph.D.

doc. Ing. Radim Kolář, Ph.D.
Chair of study program board

WARNING:

The author of the Master's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRACT

Tato diplomová práce je zaměřená na evoluční algoritmy používané v diagnostickém ultrazvuku, které by měly pomoci v při akvizici a úpravě diagnostické křivky získané pomocí techniky Bolust burst.

KEYWORDS

Ultrazvuk Evoluční algoritmy Bolust burst Ultrazvukové modely Nelineární Regrese

ABSTRAKT

This diploma thesis focuses on the evolutionary algorithms used in diagnostic ultrasound, which should help in the acquisition and modification of the diagnostic curve obtained using the Bolust burst technique.

KLÍČOVÁ SLOVA

Ultrasound Evolutionary algorithms Bolust burst Ultrasound models Nonlinear Regression

SANTARIUS, Paweł. *Evolutionary algorithms in parametric estimation for nonlinear regression models*. Brno, 2023, 78 p. Master's Thesis. Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství. Advised by Ing. Martin Mézl, PhD.

ROZŠÍŘENÝ ABSTRAKT

Tato diplomová práce se zaměřila na hodnocení výkonnosti různých modelů v rámci různých meta-heuristických optimalizačních algoritmů pro konkrétní datové sady ultrazvukových smímků z datasetů 1 a 2. V první části je kladen důraz na genetické algoritmy (GA), kde je prozkoumáno jejich inicializace, selekce, mutace, křížení a ukončení procesů.

Práce dále pokračuje pokročilými evolučními algoritmy, včetně Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony Algorithm (ABC), Spider Monkey Optimization, Ant Colony Optimization (ACO), Cuckoo Search (CS) a Firefly Algorithm (FIR). Každý z těchto algoritmů je podrobně prozkoumán s cílem porozumět jeho unikátním vlastnostem a možnostem použití v kontextu perfúzního ultrazvukového zobrazování.

V rámci každého algoritmu je provedena detailní analýza, jak tyto algoritmy fungují, jaké jsou jejich klíčové komponenty a jak mohou být použity pro optimalizaci v ultrazvukovém perfúzním zobrazování. Práce tedy poskytuje hluboké pochopení těchto evolučních algoritmů, které jsou klíčové pro pokročilé techniky zpracování obrazu a jejich aplikace v medicínské diagnostice.

Tato diplomová práce také obsahuje detailní prozkoumání několika matematických modelů používaných v ultrazvukovém perfúzním zobrazování. První model, který je prozkoumán, je model lognormální distribuce. Tento model je založen na předpokladu, že hodnoty zobrazené v datech jsou lognormálně distribuované. Detailní analýza tohoto modelu ukazuje, jak je tato distribuce využívána v kontextu perfúzního zobrazování. Dále byly využity modely LDRW, Gamma Variate, který je mezijinými široce používán v medicínském zobrazování pro modelování příchodu krve do různých částí těla. Dále je zkoumán model FPT a Lagged Normální. Každý z těchto modelů je podrobně prozkoumán s cílem porozumět jeho klíčovým vlastnostem a použití pro interpretaci dat získaných z ultrazvukového perfúzního zobrazování. Tato část práce tak poskytuje hluboké pochopení různých modelů, které mohou být použity pro interpretaci dat získaných z ultrazvukového perfúzního zobrazování.

Dalším důležitým aspektem této práce je analýza perfúze, která se zaměřuje na hodnocení průtoku krve v tělesných tkáních a orgánech pomocí specifických ultrazvukových technik.

Práce detailně prozkoumává Dopplerův ultrazvuk, který je široce používán pro sledování a měření průtoku krve v těle. Tato metoda využívá Dopplerův efekt pro zjištění rychlosti pohybu krve v cévách.

Následně je podrobně popsáno použití mikrobublin, které se v medicínském zobrazování často využívají jako kontrastní činidla. Jsou zde zkoumány dva hlavní

postupy sledování mikrobublin: sledování bolusu a metoda reperfúze. Sledování bolusu se zaměřuje na monitorování průtoku krve po injekci kontrastního činidla, zatímco metoda reperfúze zkoumá obnovu průtoku krve po dočasné ischemii.

V diplomové práci byly použity dva různé datasety. První dataset, označený jako "Dataset 1," pochází z fantomových dat. Model pro tato data byl vytvořen s využitím dialyzátoru (cartridge pro dialýzu) a dvou robustnějších trubiček poskytujících konstantní tok v systému bez recirkulace. Experiment byl opakován pro čtyři různé hodnoty průtoku. Jako kontrastní látka byl použit Sonovue, řádně naředěný, aby nedocházelo k výraznému útlumu mikrobublin. Zobrazování bylo prováděno pomocí systému GE System FiVe s 2,5 MHz sektorovou sondou v harmonickém módu. Druhý dataset, označený jako "Dataset 2," byl odvozen z obrázků myokardu prasete otevřeného hrudníku. Obrázky byly pořízeny přímo na srdci pomocí systému GE Vivid 7 s lineární sondou o frekvenci 3,5 MHz. Tyto obrázky, pořízené v takzvaném pohledu krátké osy, byly klíčovány pomocí EKG pro zobrazení maximálního naplnění komor.

Empirické výsledky získané z tohoto výzkumu poskytují významné poznatky. Zjištění zdůrazňují model Lagged Normal jako konzistentního hráče napříč všemi optimalizačními algoritmy z hlediska skóre R^2 a NRMSE. Skóre R^2 , které je ukazatelem, jak dobře se předpovědi modelu shodují s aktuálními výsledky, bylo nejvyšší pro model Lagged Normal napříč všemi algoritmy. To ukazuje, že tento model byl schopen vysvětlit větší část variance v datové sadě. Podobně NRMSE, ukazatel odchylky předpovězených hodnot od pozorovaných hodnot, byl nejnižší pro model Lagged Normal, což ukazuje na superioritu předpovědi.

Zajímavé je, že model Log Normal konzistentně prokázal špatný výkon, s negativní hodnotou R^2 napříč více algoritmy, což naznačuje, že tento model datovou sadu dobře nevystihoval. Spearmanova korelace, neparamecká míra korelace podle pořadí, naznačila silnou souvislost mezi předpovězenými a pozorovanými daty pro většinu modelů, přičemž model FPT ukázal nejvyšší korelace u většiny algoritmů.

V datovém souboru 1 se zdá, že model LDWR používající algoritmus Cuckoo Search dosáhl nejlepšího výkonu přes všechny metriky s R^2 0,8460, Spearmanovou korelací 0,9134 a NRMSE 0,4057. Tyto hodnoty naznačují vysoký stupeň vysvětlující síly, silnou korelaci a nízkou chybu, odpovídající jejich významům. Na druhou stranu ve datovém souboru 2 není snadné identifikovat jednoznačného vítěze mezi algoritmy.

Celkově výsledky této diplomové práce potvrzují užitečnost meta-heuristických optimalizačních algoritmů při zvyšování předpovědní přesnosti datových modelů. Konkrétněji, tato studie poskytuje silné empirické důkazy podporující použití modelu Lagged Normal při použití těchto algoritmů pro danou datovou sadu. Avšak tyto výsledky jsou specifické pro datovou sadu použitou v této studii a extrapolace

na jiné datové sady by měla být prováděna s opatrností.

Budoucí výzkum může rozšířit tyto poznatky tím, že se zaměří na výkon těchto a dalších modelů pod různými optimalizačními algoritmy, nebo tím, že aplikuje stejné modely a algoritmy na různé datové sady. Takové studie by přispěly k ucelenějšímu pochopení výkonnostních charakteristik těchto modelů a algoritmů a potenciálně by poskytly další vhledy pro výběr vhodných modelů a algoritmů v praxi.

DECLARATION

I declare that I have written the Master's Thesis titled "Evolutionary algorithms in parametric estimation for nonlinear regression models" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author master's thesis I furthermore declare that, with respect to the creation of this Master's Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno

.....

author's signature

ACKNOWLEDGEMENT

I would like to thank the supervisor of the bachelor thesis Mr. Ing.Mézl, Ph.D. for professional guidance, consultation, patience and suggestive suggestions to the thesis.

Contents

1	Regression analysis	14
1.1	Regression	14
1.1.1	Nonlinear regression	14
1.1.2	Kernel Ridge Regression	15
1.1.3	Support Vector Regression (SVR)	16
2	Evolutionary Algorithms	19
2.1	Genetic algorithms - GA	20
2.1.1	Initialization	20
2.1.2	Selection	20
2.1.3	Mutation	23
2.1.4	Crossover	25
2.1.5	Termination	27
2.2	Advanced Evolutionary algorithms	27
2.2.1	Particle Swarm Optimization - PSO	28
2.2.2	Differential Evolution - DE	29
2.2.3	Artificial Bee Colony Algorithm - ABC	31
2.2.4	Spider Monkey Optimization - SMO	32
2.2.5	Ant Colony Optimization - ACO	36
2.2.6	Cuckoo Search - CS	38
2.2.7	Firefly algorithm - FIR	39
3	Basic models in perfusion US imaging	42
3.1	Models	42
3.2	Lognormal distribution model	43
3.3	Gamma variate	44
3.4	LDRW - local density random walk model	44
3.5	FPT	44
3.6	Lagged Normal	45
4	Perfusion analysis	46
4.1	Doppler	46
4.2	Microbubbles	47
4.2.1	Bolus tracking monitoring	48
4.2.2	Reperfusion method	49
4.3	Preparation	50

5	Implementation of data in models, python	51
5.1	Libraries	51
5.2	Project structure	51
5.3	Flow Diagram	52
5.4	Used data	52
5.5	Evaluation	53
5.5.1	R-squared	53
5.5.2	NRMSE (Normalized Root Mean Squared Error)	54
5.5.3	Spearman Correlation	54
5.6	Results	55
5.6.1	Graphical representation of results	55
5.6.2	Statistical results	58
	Conclusion	71
	Bibliography	73
	List of symbols, physical constants and abbreviations	76
	List of appendices	77
	A Content	78

List of Figures

1.1	Kernel functions	15
1.2	graphical representation of kernels	17
1.3	SVR	18
2.1	General schema of evolutionary algorithm	19
2.2	Roulette selection	21
2.4	Rank selection	21
2.3	Tournament selection	22
2.5	Bit Flip	23
2.6	Swap mutation	24
2.7	Inversion mutation	24
2.8	Scramble mutation	24
2.9	Single point	25
2.10	Two point	25
2.11	Uniform crossover	26
2.12	Whole arithmetic recombination	26
2.13	OX1	27
2.14	Basic Particle Swarm Optimization	29
2.15	Differential Evolution	31
2.16	Artificial Bee Colony algorithm	31
2.17	Spider Monkey Foraging	33
2.18	Spider Monkey Optimization	35
2.19	Ant Colony Optimization pseudo-code	37
2.20	Cuckoo Search process	39
2.21	Firefly algorithm pseudo algorithm	41
4.1	Bolus tracking method	48
4.2	Reperfusion	49
5.1	Flow Diagram	52
5.9	R-squared - dataset 1	68
5.10	Spearman - dataset 1	68
5.11	NRMSE - dataset 1	69
5.12	R-squared - dataset 2	69
5.13	Spearman - dataset 2	70
5.14	NRMSE - dataset 2	70

List of Tables

5.1	Ant Colony Algorithm, Dataset 1	58
5.2	Artificial Bee Colony Algorithm, Dataset 1	59
5.3	Cuckoo Search, Dataset 1	59
5.4	Firefly Algorithm, Dataset 1	60
5.5	Spider Monkey, Dataset 1	61
5.6	PSO, Dataset 1	62
5.7	DE, Dataset 1	62
5.8	Ant Colony Algorithm, Dataset 2	64
5.9	Artificial Bee Colony, Dataset 2	64
5.10	Cuckoo search, Dataset 2	65
5.11	Firefly algorithm, Dataset 2	65
5.12	Spider Monkey, Dataset 2	66
5.13	PSO, Dataset 2	67
5.14	DE, Dataset 2	67

Introduction

Ultrasound perfusion imaging is a widely adopted technique that provides critical insights into tissue health and function by allowing the visualization of blood flow. To extract meaningful conclusions from these images, a sophisticated combination of mathematical modeling and computational algorithms is utilized. Central to these processes is the application of nonlinear regression and evolutionary algorithms, used for the estimation of unknown model parameters.

This thesis aims to delve into the principles of nonlinear regression and the utilization of evolutionary algorithms, with a primary focus on their role in ultrasound perfusion imaging. It seeks to critically review the different mathematical models employed in this imaging technique, shedding light on the strengths and limitations of each.

Moreover, a curve-fitting task will be carried out using suitable mathematical models on simulated and phantom datasets. At least two evolutionary algorithms, implemented in Python, will be employed in this exercise. The objective of this task is to assess the efficacy and accuracy of these algorithms in accurately fitting the models to the dataset, thus providing reliable image interpretation.

To enhance the diversity of the tools available for perfusion imaging, this study also intends to develop and implement at least three novel evolutionary algorithms that are currently not available in freely published packages. These algorithms will be tested using both phantom and clinical datasets. This approach will provide a comparative understanding of these newly developed algorithms versus conventional algorithms available in existing software packages.

The results will be analyzed and discussed in the context of the reliability of fits. This refers to the accuracy with which the mathematical models can be fitted to the imaging data using the evolutionary algorithms, which is a crucial aspect influencing the reliability and practicality of perfusion imaging.

In summary, this study will contribute to the field of ultrasound perfusion imaging by enhancing the understanding of current modeling techniques and potentially providing novel and more efficient algorithms for image interpretation. Through a methodical evaluation and comparison, this study also hopes to guide future research in the selection of optimal mathematical models and algorithms for ultrasound perfusion imaging.

1 Regression analysis

In diploma thesis we work with data of continues character. This chapter describes basic concepts and principles of regression methods, learning, problems and possibilities of usage or evaluation. Due to nonlinear character of the provided data, nonlinear methods and models will be introduced. In the matter of fact most data in the world of computing are nonlinear, rarely linear computation methods are used or seen. Regression applications are used in variety of models in financial sector, insurance or agricultural research.

1.1 Regression

Regression is a statistical method for estimating the relationships among variables. In other words regression aims to find a continuous function which captures a relation between labels and data, then generalizing following inputs into it. In machine learning regression plays important role in work with continuous variables and thus is able to complement a common discrete classification. Linear regression intend to fit a line to data-label space by estimating its parameters. The aim of nonlinear regression intends to fit a curve to the data-label space. In literature mapping function is also known as a *regressor*

$$f : \chi \rightarrow R \tag{1.1}$$

where χ represents a set of n d -dimensional vectors $\chi = X_1, X_2, \dots, X_n$ Its applications are used in variety of models in financial sector, insurance or agricultural research.

1.1.1 Nonlinear regression

Nonlinear regression is used in the cases when relationship between data and observations is nonlinear. Unlike linear regression it does not fit a line into data-label space but searches for a curve with significantly less error. Solving nonlinear regression problems is way more harder than finding parameters for linear line. It is more flexible and accurate, more versatile in the terms of curves it can accept. Nonlinear regression can be very efficient in working with complex and multidimensional data, which does not mean it results without a loss in prediction.

Ridge Regression (L2 regression)

This regression method estimates the coefficients of models where the independent variables are highly correlated. In literature it is also known as Tikhonov regularization. Generally method provides improved efficiency in parameter estimation problems also with minor, acceptable impact on bias and variance. The penalty is defined as a square of the Euclidean norm of the coefficients controlled by *complexity parameter*, α . Ridge regression is no other than LASSO with squared sum penalization of weights.

$$\beta_{est} = \operatorname{argmin}_{\beta_{est}} \|X\beta - y\|_2^2 + \alpha \|\beta\|_2^2 \quad (1.2)$$

L2 regularization is able to learn complex data patterns and gives more accurate predictions when the output variable is the function of whole input variables also is not robust to outliers.

1.1.2 Kernel Ridge Regression

Many real world problems cannot be described by linear function. If linear regression is used it will naturally have poor prediction error. A common approach is to map samples to higher dimensional space using nonlinear mapping and after that train the model in high dimensional space.

Linear	$\Phi(x_i, x_j) = x_i^\top x_j$
Polynomial	$\Phi(x_i, x_j) = (ax_i^\top x_j + r)^d$
Gaussian	$\Phi(x_i, x_j) = \exp(-\ x_i - x_j\ ^2 / (2\sigma^2))$
Sigmoid	$\Phi(x_i, x_j) = \tanh(ax_i^\top x_j + r)$

Fig. 1.1: Kernel functions [2]

For this purpose Kernel method is applied to conduct learning procedure. Figure 1.1 shows commonly used kernel functions. A Kernel is nothing but a function of our lower-dimensional vectors x , and x^* that represents a dot product of $\Phi(x)$ and $\Phi(x^*)$ in higher-dimensional space. Kernel functions are making non-linear data linearly separable also helping in avoidance of high computational demand in higher feature space.

$$K(x, x^*) = \Phi(x)^T \Phi(x^*) \quad (1.3)$$

KRR (Kernel Ridge regression) is a combination of both methods mentioned above: Kernel and Ridge regression. It is represented by equations:

$$\operatorname{argmin} \frac{1}{n} \sum_{i=1}^n \|f_i - y_i\|_2^2 + \lambda \|f\|_H^2 \quad (1.4)$$

$$f_i = \sum_{j=1}^n \alpha_j \Phi(x_j, x_i) \quad (1.5)$$

In training phase algorithms tries to figure out final parameter α to predict the regressand. The algorithm searches for the best σ and λ , which means KRR is iteration process of finding the best parameters sets. In practise the best parameters are found by cross-validation process. For more information about modern kernel techniques read [14]

1.1.3 Support Vector Regression (SVR)

Support vector regression (SVR) is a robust computational machine learning method based on the support vector machine algorithm. It is supervised learning algorithms which predicts discrete values. For more details about see [15].

Similarly to SVM it is looking for the best fit line (SVR the best hyperplane with maximum number of points). Unlike other regression models which main objective is to minimize error between the real and predicted values, SVR tries to fit line within given treshold. Threshold is nothing else than distance between the hyperplane and boundary line. SVR uses three hyperparameters: *hyperplane*, *kernel*, *boundary lines* - ϵ to control learning process. Model produced by SVR is dependant only on subset of training data, because cost function ignores training data which are close to prediction.

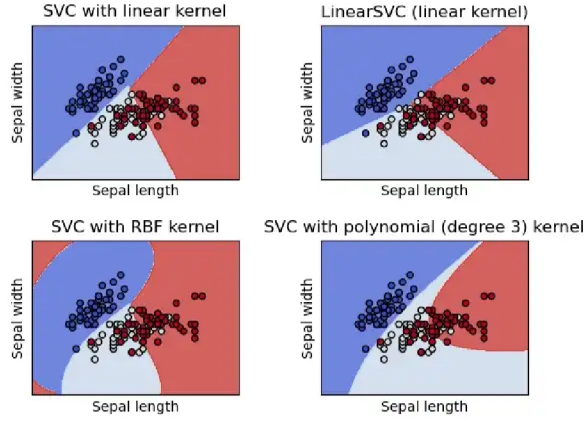


Fig. 1.2: graphical representation of kernels [3]

Support Vector Machine (SVM)

SVM method was engineered by MIT professor Vladimir Vapnik in 1979 [9]. In case of classification, method is based on finding a separating hyperplane in feature space between data of different classes. The uniqueness of this method is, in fact that it always searches for optimal split hyperplane. In other words it maximizes the width of the boundary that divides data. In this matter loss function helps out.

$$L(h_{\theta}, y) = \begin{cases} \max(0, 1 - \theta^T x), & \text{if } y = 1 \\ \max(0, 1 + \theta^T x), & \text{if } y = 0 \end{cases} \quad (1.6)$$

$$h_{\theta}(x) = \begin{cases} 1, & \text{if } \theta^T x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1.7)$$

To equation above we have to add regularization parameter λ helping in balancing a margin maximization and loss. Adding regularization parameter λ in form of $C = 1/\lambda$, equation now has a form of:

$$J(\theta) = \sum_{i=1}^m y^i L(\theta^T(x^i)) + (1 - y^i)L(\theta x^i) + \frac{1}{2} \sum_{j=1}^n \theta_j^2 \quad (1.8)$$

Large values of C cause wider margin and higher sensitiveness to outliers.

ϵ -SVR

It is similar to SVM but with notable differences like tunable parameter ϵ , which determines the width of the "tube" around the hyperplane. Next difference is that

support vectors lay outside the tube unlike the ones on the margin in SVM. Additionally we have "slack" ξ measuring the distance of the support vector to the hyperplane. Generally speaking, the aim is to define the minimal possible error by a hyperplane which puts most original points in the tube, reducing ξ at the same time.

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n |\xi_i| \quad (1.9)$$

$$|y_i - w_i x_i| \leq \epsilon + |\xi_i| \quad (1.10)$$

Generally SVRs have couple advantages, namely in robustness in to outliers, they are relatively easily updated also are excellent in generalization with relatively high prediction accuracy.

As another machine learning methods, SVRs are not an exception and also have drawbacks. SVRs are not suitable for large datasets, noisy data, class overlapping and cases where features for each data point exceed the number of training data samples.

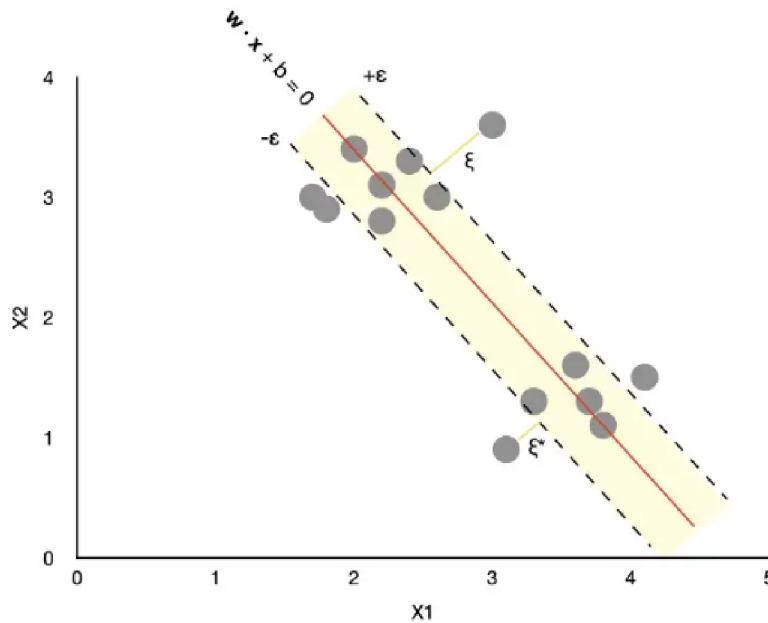


Fig. 1.3: SVR [15]

2 Evolutionary Algorithms

Evolutionary algorithms are popularised since Charles Darwin introduced the theory of Natural Selection, the theory which became a driving force behind evolution and its scientific fields like molecular biology, evolutionary computing etc. In 20th century discovery of DNA took place with huge impact on unlocking the key to genetic code by determining hereditary traits and later structure of molecule. Neo-Darwinism, newly accumulated knowledge, was a fundamental pillar for family of evolutionary algorithms. Biological evolution is powerful, problem-solving mechanism which attempts to find solutions good enough for a individual capability of survival in current environment. EAs use mechanisms such as mutation, recombination, selection and reproduction. Each individual in population is represented by a chromosome on which genetic operators are applied. By a selective force a population is updated with better individuals. The quality of the individual in relation to the given problem is determined by the so-called fitness function.

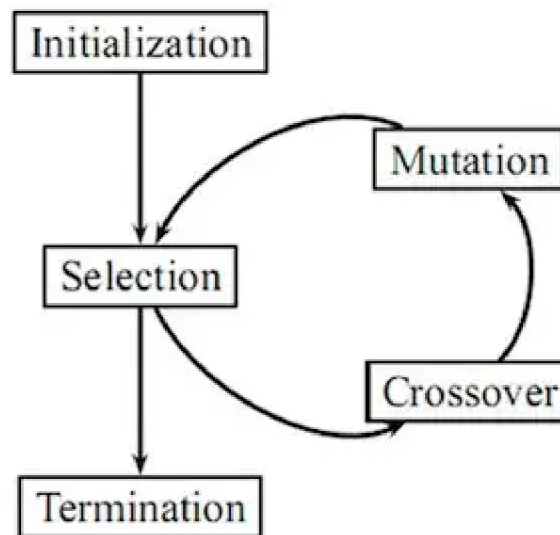


Fig. 2.1: General schema of evolutionary algorithm [5]

Evolutionary algorithms are divided into four main groups: evolutionary programming, evolutionary strategy, genetic algorithms and genetic programming. In the past they were evolving separately, thus having very similar approaches and strong similarities. Next chapters describe their implementations and techniques they use.

2.1 Genetic algorithms - GA

Development of algorithms mostly took place in 1960s and were popularised in 1970s by Mr.Holland [4]. Generally, GAs solve optimisation and adaptation problems in complex system. This chapter explains basic keyword terms in genetic algorithms usage. For a purpose of optimisation genetic operators such as selection, crossover and mutation are used working together in conjunction to achieve successful result of the algorithm.

2.1.1 Initialization

For a start here we need to define how many individuals will be in the population. Population size is represented by *popsiz*e variable.

2.1.2 Selection

Initialization is followed by process of selection, which is interpreted as natural selection in Darwin theory, fitter individuals have higher opportunity to breed. In bioinformatics a memory of selected individual capable of mating is called mating pool. Process ends with two variation operators, mutation and crossover. Mutation and crossover are imitation of reproduction, generating offspring. we can embody natural selection in many ways.

roulette wheel selection (rws)

Roulette wheel selection (RWS) can be thought of as spinning a roulette wheel, where the fitter individuals are allocated more space on the wheel. First we must determine an individual fitness value f_i as follows.

$$p_i = \frac{f_i}{\sum_{i=1}^{popsiz} f_i} \quad (2.1)$$

In programming a rotation of the wheel is an accumulated process in which after generating a random integer *rand*, we know position of the pointer on a wheel. All individuals probabilities are accumulated into memory and compared to *rand*. Whenever we find an individual, who satisfies condition 2.2 is selected into *mating pool*.

$$\sum_{i=1}^k p_i < rand < \sum_{i=1}^{k+1} p_i \quad (2.2)$$

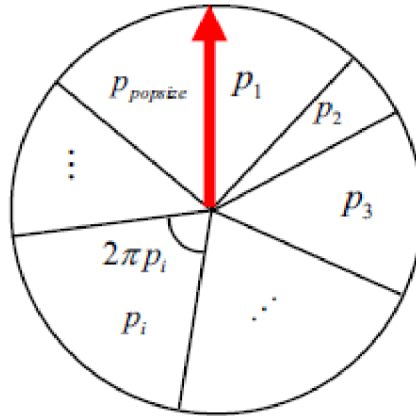


Fig. 2.2: Roulette selection[6]

tournament selection

Programmatically, process of selection is based on roulette wheel selection (RWS). In this process fitter individuals have more space allocated. The fittest individual in the current population is always the tournament winner. See 2.3.

rank selection

The rank of an individual is a number indicating how many individuals in the population have poorer fitness. This rank number is then used to calculate the probability of the selection of an individual. The rescaled fitness values rather than the original fitness values are used in the selection process. The equations for recalculation can take several forms.

$$f_{rank} = 2 - P + 2(P - 1) \frac{rank - 1}{n - 1} \quad (2.3)$$

where n is a number of members in population, P is a scaling factor determining a selection pressure, rank is the fitness ranking of individual in population (the least fit unit has rank = 1)

Table 3.6. Rank ordering and selection fitness

Ranking	1	2	3	4	5
Rescaled fitness	0	0.5	1	1.5	2
Selection probability	0	0.10	0.20	0.30	0.40

Fig. 2.4: Rank selection[6]

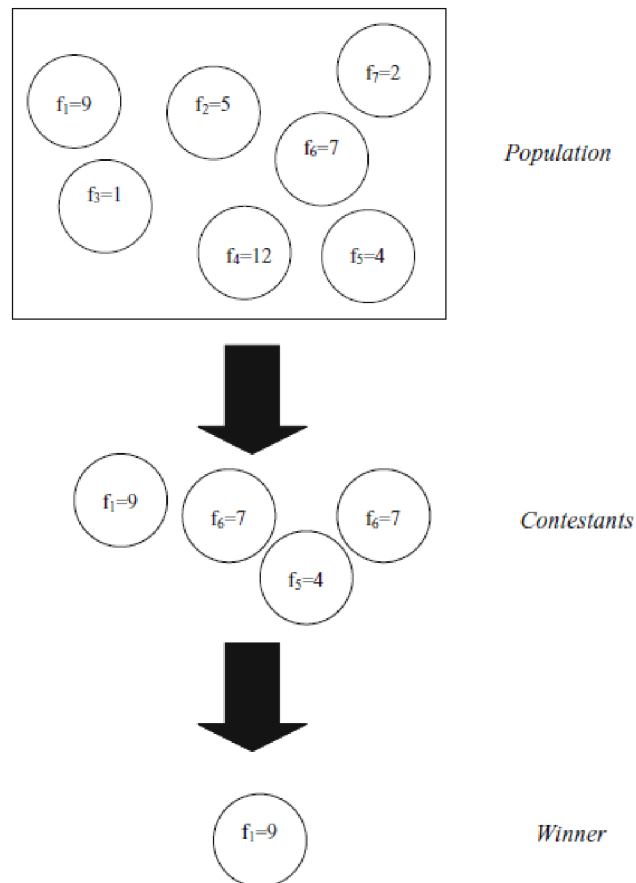


Fig. 2.3: Tournament selection[6]

Rank selection has two advantages. First, is that it lessens the risk of biasing the search process as a result of too-intensive selection of the better solutions in the early generations of the GA. Second, is that it only requires relative measures of fitness. This could be an advantage if fitness measures are noisy.

2.1.3 Mutation

Mutation is analogous to biological mutation. Mutation is a process with high importance in GA because it ensures that the search process never stops. In iteration process it might discover a novelty. Mutation rate implicates the usefulness of selection and crossover process. High mutation leads to highly overpower selection and crossover and GA is capable of effective reassembly of a random search process. But it is desirable to balance mutation, higher mutation could lead to faster convergence, thus destroying possible novelty along the search. In EA all mutation operators apply following requirements:

1. every point must be reachable by one or more mutations
2. no drift, no preference in direction in the search space
3. small mutations more likely occur then large ones

Bit Flip

It is used in binary encoded GAs. It selects one or more bits randomly and flips them to opposite values.

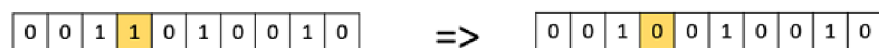


Fig. 2.5: Bit Flip [7]

Swap

Swap is very common in permutations. it randomly selects two positions on the chromosome.

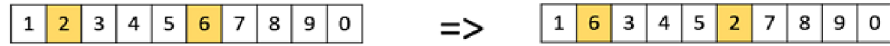


Fig. 2.6: Swap mutation [7]

Inversion

Inversion in the first phase selects a subset of genes, then inverts them entirely without shuffling.

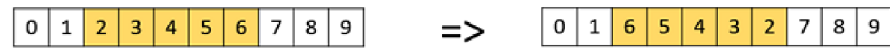


Fig. 2.7: Inversion mutation [7]

Scramble

Scramble is used in permutation representations. It selects a subset of genes from a whole chromosome and shuffles their values randomly or scrambles them.

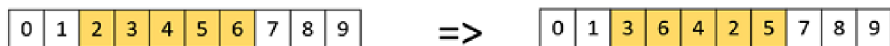


Fig. 2.8: Scramble mutation [7]

Random reset

This type of mutation is for a integer representation. In this case a randomly chosen value from a set is assigned to random gene. It is basically a bit flip with integer values.

2.1.4 Crossover

In literature is also known as recombination. Cross over combines the genetic information of two parents to generate new offspring. In contrast, crossover benefit allows to search more intensively around previously discovered good solutions. It is analogous to reproduction and biological reproduction. In GAs it is usually applied with a high probability.

single point

Single point crossover belongs to binary operations. One point is selected and one part of the parent are switched with another part of the second parent.

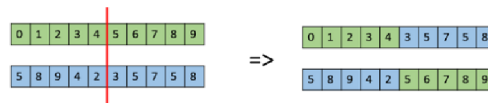


Fig. 2.9: Single point [6]

multi-point

This process works very similarly to single point, but with multiple separation points. Alternating segments are swapped to get a new off-springs. his technique is commonly used in genetic algorithms because it allows for the exchange of information between solutions while maintaining diversity in the population, leading to faster convergence towards the optimal solution.



Fig. 2.10: Two point [6]

uniform

In uniform crossover the chromosome is not divided into segments, each gene is treated separately. This process randomly chooses a gene, thus decides if the gene will be included in the off-spring. Uniform crossover is useful for maintaining diversity in the population and exploring the search space more thoroughly, leading to faster convergence towards the optimal solution.

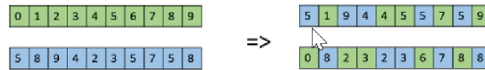


Fig. 2.11: Uniform crossover [6]

whole arithmetic recombination

This operation is mostly used in integer operations using a weighted average of the two parents. Whole arithmetic recombination takes a percentage of each parent gene and adds them to produce new solutions. The percentage of each parent gene present in the child gene is determined by a parameter alpha. Child formula is equal to:

$$Child = \alpha x + (1 - \alpha)y \quad (2.4)$$

Where X and y are parent genes. An alpha of 0.5 will produce identical child chromosomes as shown in the image below.

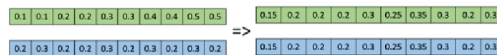


Fig. 2.12: Whole arithmetic recombination [6]

David's order crossover (OX1)

OX1 is used in permutation based crossovers. Its intention is to transmit information about relative ordering to the off-springs. OX1 is a genetic algorithm that is used to solve optimization problems. It operates on a set of solutions represented as ordered chromosomes, and the algorithm works by taking two parent solutions and generating two offspring solutions by combining the order of elements from the parents. The idea is to maintain the useful features of both parents, while allowing for exploration of new solutions. OX1 is widely used in evolutionary algorithms for problems in scheduling, routing, and permutation-based optimization.

There are several similar genetic algorithms that operate on ordered chromosomes, and some of the commonly known ones include: Cycle Crossover (CX), Partially Mapped Crossover (PMX), Edge Recombination Crossover (ERX).

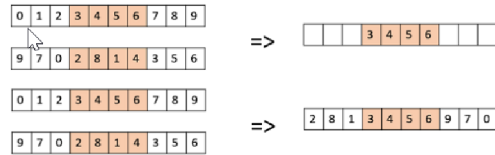


Fig. 2.13: OX1 [6]

2.1.5 Termination

In evolutionary algorithms, the optimization process is an iterative one that involves the generation of a population of solutions and the evaluation of their fitness (or quality) over time. The termination criteria determine when the optimization process has reached a satisfactory state, such as when the best solution has converged to a satisfactory level of quality, or when a maximum number of generations has been reached. Termination criteria are critical to the success of evolutionary algorithms, as they control the duration of the optimization process and determine the quality of the final solution. Common termination criteria include:

1. Convergence - process stops when the quality of the best solution has stabilized, or when the improvement in quality over a number of generations has slowed to an acceptable rate
2. Maximum number of generations - process stops after a fixed number of generations have been reached
3. Time limit - process stops after a specified amount of time has passed
4. Solution quality threshold - process stops when the best solution reaches a specified quality threshold

The choice of termination criteria depends on the nature of the problem being solved, as well as the desired balance between computational time and solution quality.

2.2 Advanced Evolutionary algorithms

AEAs are widely used in a variety of applications, including machine learning, data analysis, and engineering design optimization, among others. They are particularly useful for problems where the search space is large and the solution space is complex and non-linear. They are a type of metaheuristic algorithms and include genetic algorithms, particle swarm optimization, differential evolution, ant colony optimization, and artificial bee colony algorithm, among others. They differ from

traditional optimization algorithms in that they are not based on gradient-based methods, but instead rely on random search and exploration.

2.2.1 Particle Swarm Optimization - PSO

Particle Swarm Optimization (PSO) is a swarm intelligent algorithm, inspired from 'birds' flocking or fish schooling for the solution of nonlinear, nonconvex or combinatorial optimization problems that arise in many science and engineering domains.[16]

What differs swarm intelligent algorithms from standard evolutionary algorithms is a principle of cooperation. As we know Standard evolutionary algorithms are based on competition.

In short, in evolutionary algorithms a new population is evolved in every generation iteration while in swarm intelligent algorithms in every generation iteration individuals make themselves better. Identity of the individual does not change over the iterations. Creators of PSO model Kennedy and Eberhart defined five fundamental principles that determine if group of agents is swarm or not [16]:

- Proximity - the population should be able to respond to quality factors in the environment.
- Quality - the population should be able to respond to quality factors in the environment
- Diverse Response - the population should not commit its activity along excessively narrow channels
- Stability - the population do not change mode of behaviour when environment changes
- Adaptivity - the population should be able to change its behaviour mode when it is worth

process

PSO is a swarm intelligent search algorithm. This search is done by a set of randomly generated potential solutions. This collection of potential solutions is known as *swarm* and each individual potential solution is known as a *particle* [16].

PSO search is influenced by two types of learning: *social learning* (ability to learn from others) and *cognitive learning* (own learning process). Result of social learning is *gbest*, the best solution which the particle stores in memory, wheres *pbest* is the best solution of individual it met on its way.

Any particle also have a factor of velocity, which represents a change of direction and the magnitude in time (iteration). Velocity update equation follows as:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (2.5)$$

Where d represents a dimension, i is iteration, t is time, c_1 and c_2 are constants called cognitive and social scaling parameters, respectively or simply acceleration coefficients r_1 and r_2 (in range from $[0,1]$). Next is position update equation.

$$x_{t+1id} = x_{id}^t + v_{id}^{t+1} \quad (2.6)$$

This algorithm has also a stopping criteria parameter. Popular stopping criteria are usually based on maximum number of function evaluations or iterations which are proportional to the time taken by the algorithm and acceptable error[16]. If an algorithm does not improve in a certain number of iterations, search should be stopped.

```

for  $t=1$  to the maximum bound on the number of iterations do
  for  $i=1$  to  $S$  do
    for  $d=1$  to  $D$  do
      Apply the velocity update equation 1;
      Apply position update equation 2;
    end
    Compute fitness of updated position;
    If needed, update historical information for pbest and gbest;
  end
  Terminate if gbest meets problem requirements;
end

```

Fig. 2.14: Basic Particle Swarm Optimization[16]

2.2.2 Differential Evolution - DE

Differential Evolution (DE) is a powerful and popular algorithm in the field of evolutionary computation. Its success lies in its simplicity, robustness, and ease of use. In the following subsections, we detail each step of the algorithm.

The DE process begins with the initialization of a population of potential solutions. These solutions, often selected randomly within predefined boundaries, form the base for the search of an optimal solution. If the optimization problem is D -dimensional, we initialize a population with NP individuals, each represented as a D -dimensional vector. An individual vector is denoted as $X[i, G]$, where i is the individual's index, and G is the current generation [17]

Next the process of mutation follows, in which DE generates new parameter vectors by adding the weighted difference of two vectors from the population to a third vector. This operation, known as mutation, is described mathematically as

$$V[i, G + 1] = X[r1, G] + F * (X[r2, G] - X[r3, G]) \quad (2.7)$$

$r1, r2, r3$: three randomly selected individuals from the current population

F : scaling factor, values 0.0 and 2.0

After mutation, DE performs a crossover operation to produce trial vectors $U[i, G + 1]$. These vectors are a combination of the mutant vector and the original target vector $X[i, G]$. For each parameter j in vector $U[i, G + 1]$, the value is taken either from the mutant vector $V[i, G + 1]$ or the original vector $X[i, G]$. The choice depends on whether a random number is less than the crossover probability CR . If the random number is less than CR , the value from the mutant vector is used. Otherwise, the value from the original vector is chosen. However, to ensure diversity, at least one parameter is always taken from $V[i, G + 1]$.

In the selection phase, the trial vector $U[i, G + 1]$ is compared with the target vector $X[i, G]$. If the trial vector yields a better fitness score, it replaces the target vector in the next generation.

The mutation, crossover, and selection operations are iteratively repeated until a termination condition is met. This could be reaching a maximum number of generations, achieving a minimum fitness threshold, or finding a satisfactory solution.

```

generate initial population of size  $NP$  ;
Do
  For each individual  $i$  in the population
    generate 2 random integers,  $r_1, r_2 \in \{1, 2, \dots, NP\}$ ,  $r_1 \neq r_2 \neq i$  ;
    generate a random integer  $j_{rand} \in \{1, 2, \dots, D\}$  ;
    For each parameter  $j$ 
       $diff = F * (x_{best,j} - x_{i,j} + x_{r_1,j} - x_{r_2,j})$  ;

       $u_{i,j} = \begin{cases} x_{i,j} + diff & \text{if } rand(0,1) < CR \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases}$  ;
    End For
    Replace  $\bar{x}_i$  with the trial individual  $\bar{u}_i$ , if  $\bar{u}_i$  is better;
  End For
Until the termination condition is achieved.

```

Fig. 2.15: Differential Evolution[17]

2.2.3 Artificial Bee Colony Algorithm - ABC

ABC algorithm was developed in 2005 which simulates behaviour of honey bees. It tries to find the most profitable source by using local and global search mechanisms with various selection mechanisms performed by bees. Algorithm is based on three different groups of bees (Employed, Onlooker, Scout). These groups also represent phases in iterative process of the algorithm.

- 1: Initialization
- 2: **repeat**
- 3: Employed Bees' Phase
- 4: Onlooker Bees' Phase
- 5: Memorize the best solution achieved so far
- 6: Scout Bee Phase
- 7: **until** Termination criteria is satisfied

Fig. 2.16: Artificial Bee Colony algorithm[16]

process

In initialization phase a population is generated randomly by equation:

$$x_{i,j} = x_j^{min} + rand(0, 1)(x_j^{max} - x_j^{min}) \quad (2.8)$$

In employed bee phase a local search is performed by local searching algorithm.

After it selection between the current solution and its mutant is carried you. The local and greed selection is applied to each food source in the population.

After first phase Onlooker phase continues. In this search solutions are selected stochastically depending on their fitness values. Again high fitness values are likely selected, based on probability in 2.10.

Next the Scout Bee Phase follows. In this phase exploration of the new sources takes place, it is a fluctuation effect which can bring innovation in availability in food sources.

$$\nu_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2.9)$$

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (2.10)$$

where i is iteration, SN is the number of food sources, D the number of design parameters, x_j^{min} and x_j^{max} are lower and upper boundary of j -th dimension, ϕ is a real random number $[-1,1]$, k is neighbour solution chose randomly by ϕ .

Regardless all phases above, ABC algorithm has three control parameters: *number of food sources, the maximum number of cycles, limit of exhausted sources*. Nowadays a lot of modifications of ABC exist and each of this modifications can have its own parameters. ABC has proved itself in many engineering fields like filter design, hydrology, civil engineering, mechanic, control systems, scheduling, data mining etc. Its performance is compares to those of k -means, fuzzy-c-means and particle swarm optimization.

2.2.4 Spider Monkey Optimization - SMO

Spider monkeys have been categorized by biologists as fission-fusion social structure (FFS) based animals. This means they follow fission-fusion social systems, which means they have a mutual interest in working in a large group and sometimes based on need, they divide themselves in smaller groups for more effective foraging. Therefore, their strategy for survival and foraging was implemented in algorithm. For a better understanding of foraging see 2.17.

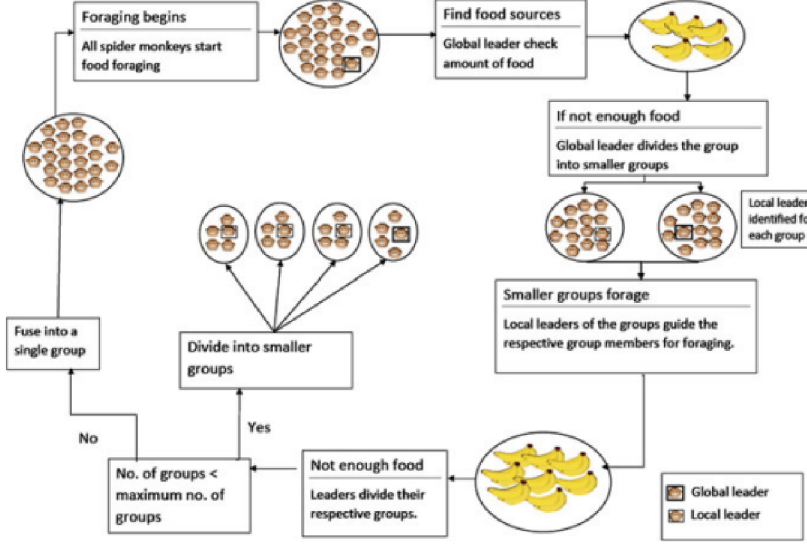


Fig. 2.17: Spider Monkey Foraging [16]

process

Whole process has four steps. First the group led by dominant female starts food foraging and distance evaluation from the food. Next group members evaluate and if beneficiary update them. After update evaluation of the distance happens again. In the third step, local leader updates its best position within the group but if the position is stick (not updated), all members of the group start searching for food in the different directions. Last step, the forth one, is when a leader of the group updates the best met position. In case of stagnation groups is divided into a smaller groups. All steps before are executed until the desired output is achieved.

For a better control of the process, there are two parameters *GlobalLeaderLimit*, *LocalLeaderLimit*. Both of them help the leaders to take appropriate decisions.

Local Leader Phase (LPP) equation helps all spider monkeys to update their position. Modification is based on local leader.

$$SM_{new_{i,j}} = SM_{i,j} + U(0, 1)(LL_{k,j} - SM_{i,j}) + U(-1, 1)(SM_{r,j} - SM_{i,j}) \quad (2.11)$$

Here $U(0, 1)$ is a uniformly distributed random number in the range $(0, 1)$ and same applies to $U(-1,1)$ with own appropriate range. After completion of Global Leader Phase (GLP) follows where update is based on selection probability thanks to fitness function.

$$fitnessfunction = fit_i = \begin{cases} \frac{1}{1+f} & if f_i \geq 0 \\ \frac{1}{abs(f_i)} & if f_i < 0 \end{cases} \quad (2.12)$$

The selection probability $prob_i$ is determined by roulette wheel selection.

$$prob_i = 0.9 \frac{fit_i}{max_{fit}} + 0.1 \quad (2.13)$$

Update position of a global leader is identical to 2.11, where LL is replaced with GL. SMO better balances between exploitation and exploration trying to search optimal solution. It also posses an inbuild mechanism for stagnation check thanks to local and global learning processes. . The local leader decision phase creates an additional exploration while in the global leader decision phase, a decision about fission or fusion is taken. Therefore, in SMO exploration and exploitation are better balanced while maintaining the convergence speed. [16]

SMO shows that it has a great optimizer with a huge potential, however its downsides when using a large number of self dependent parameters. Some studies show SMO in some cases is capable of outpacing ABC, DE and PSO.

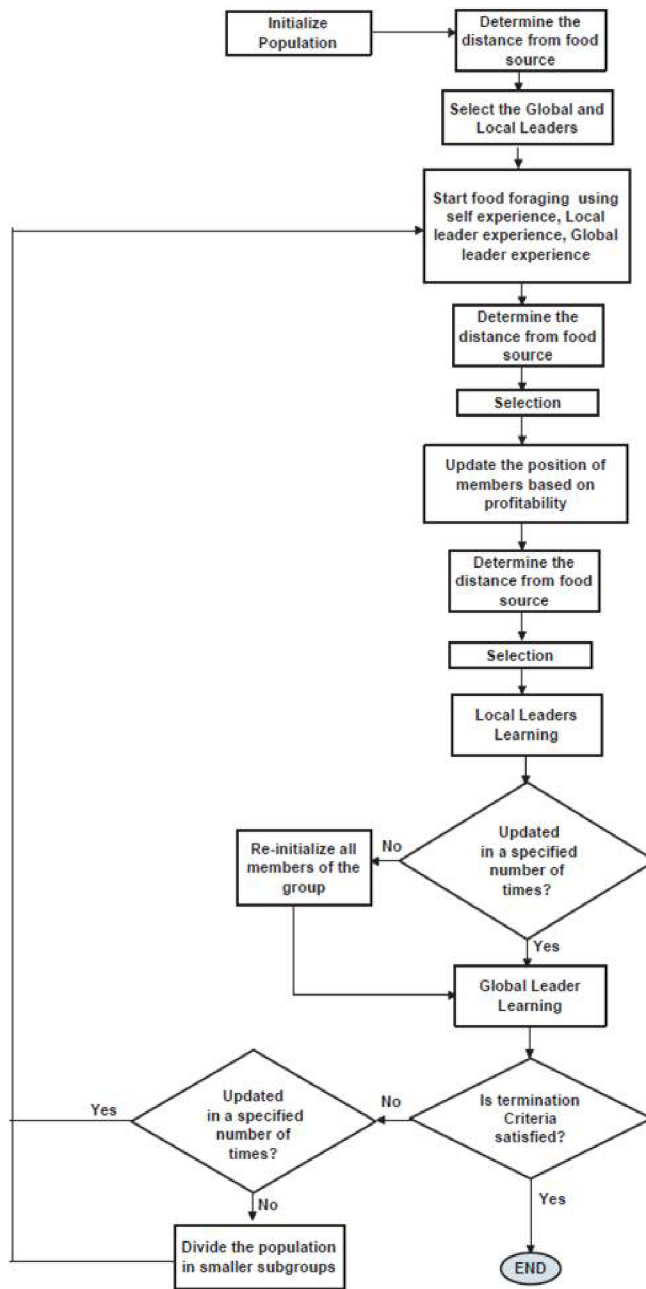


Fig. 2.18: Spider Monkey Optimization[16]

2.2.5 Ant Colony Optimization - ACO

The ACO is a probabilistic technique used for founding approximate solutions to difficult optimization problems, developed by Marco Dorigo in 1992. It belongs to group of swarm intelligence algorithms. The whole algorithm is modeled on the actions of an ant colony finding source of food and its natural behaviours guided by pheromone-based communication of biological ants. Generally it is trying to find shortest possible path through graphs.

process

Informally, an ACO algorithm can be imagined as the interplay of three procedures: *ConstructAntsSolutions*, *UpdatePheromones*, and *DaemonActions*.

ConstructAntsSolutions manages a colony of ants that concurrently and asynchronously visit adjacent states of the considered problem by moving through neighbor nodes of the problem's construction graph. They move by applying a stochastic local decision policy that makes use of pheromone trails and heuristic information[19]. This is the way how ants are building solutions. Once the solution is constructed or is still under construction ant evaluates the partial solution in *UpdatePheromones* procedure in order to decide how much pheromones should be deposited.

$$p_{ij}^k = \frac{[\tau_{ij}]^{\alpha} [N_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}]^{\alpha} [N_{il}]^{\beta}}, \quad \text{if } j \in N_i^k \quad (2.14)$$

where τ_{ij} is the pheromone level on edge (i,j), N_{ij} is the heuristic value of node j, α and β are parameters that control the influence of pheromones and heuristics, respectively, and the denominator is a normalizing factor.

UpdatePheromones is the process in which pheromone trails are updated/modified. When and ant finds a great source of food, it starts to strengthen pheromone trail back. This has valuable information provided to ant coworkers so that they can find a source of food more easily. but we also should remember that pheromones are evaporating. In other cases when food source was not found or inefficient in nutrients, ant worker will not mark the trail on its return. Evaporating can be interpreted as a form of forgetting. Due to a forgetting ACO avoids a too rapid convergence towards a suboptimal region, thus favoring the new exploration areas in search space. Pheromone evaporation:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad \forall (i,j) \in L \quad (2.15)$$

where ρ is the pheromone evaporation rate and $\delta\tau_{ij}$ is the amount of pheromone deposited on edge (i,j) , which depends on the quality of the solution that uses that edge.

Pheromone deposit:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum \Delta\tau_{ij}^k, \quad \forall (i,j) \in L \quad (2.16)$$

where $\delta\tau_{ij}^k$ is the amount of pheromones ant k

The final and optional procedure is *Daemon Action serving* as a main component for implementation of centralized actions, which cannot be done by single ants. Examples of daemon actions are the activation of a local optimization procedure, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a non-local perspective. As a practical example, the daemon can observe the path found by each ant in the colony and select one or a few ants (e.g., those that built the best solutions in the algorithm iteration) which are then allowed to deposit additional pheromone on the components/connections they used [19].

```

procedure ACO_Metaheuristic
  ScheduleActivities
    ConstructAntsSolutions
    UpdatePheromones
    DaemonActions      % optional
  end-ScheduleActivities
end-procedure

```

Fig. 2.19: Ant Colony Optimization pseudo-code[19]

As mentioned before, like other swarm algorithms, it has many modifications and extensions. Mostly known and used modified versions are: Ant colony system (ACS), Ant system (AS), Max-min ant system (MMAS), Parallel ant colony optimization (PACO). ACO has been applied in many case e.g network routing problem, vehicle traffic.

2.2.6 Cuckoo Search - CS

Cuckoo Search is nature-inspired metaheuristic algorithm developed by in 2009 by Xin-She Yang and Suash Deb. CS is based on brood parasitism, which is typical feature of Cuckoo birds. What's more algorithm does not use simple isotropic random walks, but combines them with Levy's flights. For better general understanding CS uses 3 rules:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest
- The best nests with high-quality eggs will be carried over to the next generations
- The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p \in (0, 1)$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest [18]

As mentioned before ACO is combination of locally applied random walk and globally Levy's distances. Random local walk can be written like:

$$x_i^{t+1} = \alpha S \cdot H(p_a - \epsilon) \cdot (x_j^t - x_k^t) \quad (2.17)$$

whereas global random walk is carried out by using Levy's flights. Levy flight involves generating new solutions by performing a random walk, inspired by the Levy flight pattern of cuckoo birds. The new solutions are generated by adding a random perturbation to the current solution, where the perturbation is generated according to a Levy distribution. The Levy distribution has a heavy tail, which allows for larger steps in the search process [18].

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \quad (2.18)$$

where L is the characteristic scale of the problem of interest:

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0) \quad (2.19)$$

In equations above $\alpha > 0$ is a step size scaling factor, p_a is a switching parameter, H is heuristic function, ϵ is a random number drawn from a uniform distribution, s is the step size, x_j^t and x_k^t are two different solutions selected randomly by random permutation.

process

1. Randomly generate an initial population of n nests at the positions, $\mathbf{X} = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_n^0\}$, then evaluate their objective values so as to find the current global best g_t^0 .
2. Update the new solutions/positions by

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \otimes L(\lambda). \quad (14)$$

3. Draw a random number r from a uniform distribution $[0, 1]$. Update $\mathbf{x}_i^{(t+1)}$ if $r > p_a$. Then, evaluate the new solutions so as to find the new, global best g_t^* .
4. If the stopping criterion is met, then g_t^* is the best global solution found so far. Otherwise, return to step (2).

Fig. 2.20: Cuckoo Search process [18]

CS has been applied in many areas with high demand on optimization proving its promising efficiency. It has superiority in terms of performance over other algorithms for a continuous optimization problems e.g spring design and welded beam design problems. Cuckoo search has many modifications made by different authors. It was applied in e.g neural network optimizations, embedded systems, optimized semantic web search process, training neural network. Also CS has been used to solve nurse scheduling problems, software testing and test data generation. More can be found [18]. The applied

2.2.7 Firefly algorithm - FIR

The Firefly Algorithm (FA) is a metaheuristic algorithm for global optimization, which was introduced by Yang in 2008, and is inspired by the flashing behavior of firefly insects [22]. Fireflies utilize flashing behavior to attract other fireflies, primarily for signaling purposes to the opposite sex. Since its development, FA has received significant attention and has been successfully applied in various applications. For instance, FA has been employed for efficient clustering [23], tracking expanding oil spills and estimating their areas, discovering opinion leaders in online social networks, and solving optimization problems in the computer-aided processing planning system [23].

Flashing lights serve as two purposes: to warn potential predators and to attract mating partners. Flashing light intensity obeys three idealized rules [22].

- Fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
- The attractiveness is proportional to the brightness and they both decrease as their distance increases. Thus for any two flashing fireflies, the less brighter one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly.
- The brightness of a firefly is determined by the landscape of the objective function.

This algorithm faces two principal main issues, which are: the variation of light intensity and formulation of the attractiveness. For simplicity we assume that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function [23]. The attractiveness of a firefly, represented by β , is dependent on the distance r_{ij} between itself and other fireflies. The brightness I of a firefly at a specific location x can be determined as $I(x) \propto f(x)$. The light intensity $I(r)$ decreases with distance r from the source, and its absorption in the medium depends on the light absorption coefficient γ . To avoid singularity at $r = 0$, the combined effect of both the inverse square law and absorption can be approximated using a Gaussian form. This is given by

$$I(r) = I_0 e^{-\gamma r^2} \quad (2.20)$$

Together with a distance of any two fireflies i and j the distance variable r can be defined by the Euclidean distance or any another distance equations can be used Mahattan, Minkowski, Cherebychev, Jaccard etc. Each of them has its own strengths and weaknesses.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (2.21)$$

So if we combine two equations above, we get an equation of the movement of a firefly i which is attracted to more brighter (attractive) firefly j .

$$x_i = x_i + I_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha (rand - \frac{1}{2}) \quad (2.22)$$

Here, the second term represents attraction and the third term randomization. Rand is a random number generator $[0, 1]$, where α is randomization parameter, *gamma* characterizes the variation of the attractiveness. The *gamma* is crucial in determining the speed of convergence, which implies to whole algorithm behaviour.

process

Firefly Algorithm

Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_d)^T$
Generate initial population of fireflies \mathbf{x}_i ($i = 1, 2, \dots, n$)
Light intensity I_i at \mathbf{x}_i is determined by $f(\mathbf{x}_i)$
Define light absorption coefficient γ
while ($t < \text{MaxGeneration}$)
 for $i = 1 : n$ all n fireflies
 for $j = 1 : i$ all n fireflies
 if ($I_j > I_i$), *Move firefly* i towards j in d -dimension; **end if**
 Attractiveness varies with distance r via $\exp[-\gamma r]$
 Evaluate new solutions and update light intensity
 end for j
 end for i
 Rank the fireflies and find the current best
end while
Postprocess results and visualization

Fig. 2.21: Firefly pseudo algorithm [22]

As the authors of the the research paper claim [23], the firefly algorithm is highly efficient in its basic form. However, as the optima approach, the solutions continue to change. By gradually reducing the randomness and varying the randomization parameter, the solution quality can be improved and the algorithm's convergence can be further enhanced. These are important topics for future research. Additionally, the firefly algorithm can be extended to solve multi-objective optimization problems and combined with other algorithms for even more exciting research possibilities.

3 Basic models in perfusion US imaging

Perfusion is defined as the flow of blood (or other body fluids) through the tissues. Adequate blood flow is necessary for the proper function of a tissue or organ, ensures the supply of oxygen and nutrients, and also ensures the outflow of metabolic products.

Perfusion is influenced by blood pressure, the activity of the heart, the amount of blood and fluids in the body. Perfusion analysis is mainly used for oncological and ischemic observations and patient examination. Perfusion is a promising tool for the exclusionary resolution of cancer or inflammation. The advantage of ultrasound perfusion measurement is real-time measurement and monitoring along with the low cost of the examination. Nevertheless, ultrasound images suffer from noises, such as poor image quality, artifacts (especially attenuation), which can cause difficulties in quantitative analysis. The basis for monitoring perfusion is the temporal recording of changes in contrast agent concentration in tissue after intravascular administration. There are two basic approaches for quantitative analysis, namely the perfusion method and bolus tracking.

3.1 Models

All perfusion models mentioned in this chapter belong to article *A multi-model framework to estimate perfusion parameters using contrast-enhanced ultrasound imaging* [8]. Modeling of perfusion curves is important for comparison of physiological perfusion parameters and phantom models to verify their accuracy. Comparison of models and perfusion curves is based on perfusion parameters. Basic parameters are:

MTT	mean transit time, which represents mean time of the flow through region of interest (ROI) [s]
t_p	time-to-peak, time in which a contrast agent reaches a maximum concentration [s]
Blood flow	in TOI per tissue weight [$ml \cdot g^{-1}$]
Volume	in TOI per tissue weight [$ml \cdot min^{-1}g^{-1}$]
AUC	area under the curve [$g \cdot min \cdot ml^{-1}$]

The flow rate (F) and volume model (V) are calculated by using Steward-Hamilton relations, where (m) represent a mass [12].

$$F = \frac{m}{AUC} \quad (3.1)$$

$$V = F \cdot MTT \quad (3.2)$$

The single perfusion model is used for a estimation of perfusion parameters because of the sensitivity to initial values for perfusion parameters and selection of appropriate boundary conditions for algorithms, which may result in unrealistic values. Nowadays in clinical CEUS we face with 5 different perfusion models:

AUC : Area under the curve

t_0 : Time offset

C : Baseline intensity offset

MTT : Mean time to resolution

μ : Mean

σ : Standard deviation

α : Number of equally sized homogeneous compartments in a series

β : Volume of each compartment divided by the flow rate

Γ : Count of compartments

λ : Péclet number divided by 2

3.2 Lognormal distribution model

In clinical practice, empirically lognormal model is used in Hearth and Breast diagnostics. This model is based on the vessels with bifurcations, which lead to fractal behaviour [8].

$$I(t) = \frac{AUC}{\sqrt{2\pi}\sigma(t-t_0)} e^{\frac{\ln(t-t_0)-\mu}{2\sigma^2}} + C \quad (3.3)$$

$$MTT = e^{\mu-\sigma^2/2} \quad (3.4)$$

$$t_p = e^{-\sigma^2} \quad (3.5)$$

3.3 Gamma variate

This model takes in count constant flow through series of equally-sized, homogeneous compartments. Distribution fits best to a carotid imaging [8].

$$I(t) = \frac{AUC}{\beta^{\alpha+1}\Gamma(\alpha+1)}(t-t_0)^\alpha e^{-\frac{t-t_0}{\beta}} + C \quad (3.6)$$

$$MTT = \beta(\alpha+1) \quad (3.7)$$

$$t_p = \alpha\beta \quad (3.8)$$

In ultrasound imaging, Gamma variate fitting is often used to evaluate tissue perfusion. This technique tracks the enhancement of an ultrasound contrast agent over time in Dynamic Contrast-Enhanced Ultrasound (DCE-US) imaging. By analyzing the resulting time-intensity curves, information about blood flow and tissue perfusion can be obtained.

3.4 LDRW - local density random walk model

LDRW distribution represents a fluid model with diffusion within a vessel. Model assumes the microbubbles pass vessel boundaries multiple times [8].

$$I(t) = AUC\left(\frac{e^\lambda}{\mu}\right)\sqrt{\frac{\lambda}{2\pi}}\frac{\mu}{(t-t_0)}e^{-\frac{1}{2}\lambda\left(\frac{\mu}{(t-t_0)}+\frac{(t-t_0)}{\mu}\right)} + C \quad (3.9)$$

$$MTT = \mu \quad (3.10)$$

$$t_p = \frac{\mu}{2\lambda}(\sqrt{1+4\lambda^2}-1) \quad (3.11)$$

3.5 FPT

Model also considers fluidity and diffusion in a vessel, similarly to LDRW. Difference is that FPT assumes a microbubbles pass only once through the vessel. First passage time model best fits to a carotid[8].

$$I(t) = AUC\left(\frac{e^\lambda}{\mu}\right)\sqrt{\frac{\lambda}{2\pi}}\left(\frac{\mu}{(t-t_0)}\right)^{\frac{3}{2}}e^{[-\frac{1}{2}]\lambda\left(\frac{\mu}{(t-t_0)}+\frac{(t-t_0)}{\mu}\right)} + C \quad (3.12)$$

$$MTT = \mu \quad (3.13)$$

$$t_p = \frac{\mu}{2\lambda}(\sqrt{9+4\lambda^2}-3) \quad (3.14)$$

3.6 Lagged Normal

This model was created for a situation in which dispersion of microbubbles in large vessels is random and followed by merging in micro-vascular network. Authors of the [8] article recommend using this model for a liver modeling.

$$I(t) = \frac{AUC}{2} \lambda e^{-\lambda(t-t_0) - \frac{\mu^2}{2\sigma^2} + \frac{(\mu+\lambda)^2}{2\sigma^2}} \left[1 + \frac{(t-t_0) - \mu - \lambda\sigma^2}{\sqrt{2}\sigma^2}\right] + C \quad (3.15)$$

$$MTT = \mu + 1\lambda \quad (3.16)$$

4 Perfusion analysis

Perfusion is the process of liquid flow through the tissues. Blood flows is essential for the proper functioning of the tissue and organs ensuring the supply of the nutrients and oxygen, also provides the drainage of the metabolic wastes. Perfusion is dependent on couple of factors, namely: blood pressure, heart activity, amount of the blood and fluids, condition of the vessels supplying the organ and rheological characteristics of the blood and fluids.

Perfusion is the act of a fluid traversing through body tissues. It is crucial for tissues and organs to function correctly because it ensures that they receive necessary nutrients and oxygen. It also helps to eliminate metabolic by-products. Numerous factors influence perfusion, including blood pressure, heart function, blood volume, the health of blood vessels serving the organ, and the flow properties of the blood and fluids.

Ultrasound technology can be employed to visualize perfusion in two primary ways: the conventional method using Doppler and a newer method involving microbubble contrast agents. The Doppler method is straightforward and cost-effective, but it is restricted to larger vessels with higher flow speeds. It is ineffective for examining the microvasculature because the general tissue motion exceeds capillary flow speed. This method has been used for assessing liver and tumour blood flow.

On the other hand, contrast studies offer more comprehensive insights and can evaluate both macro and microcirculation. One strategy involves examining time-intensity curves in a specific area, such as a tumour, heart muscle, or brain, after an intravenous bolus injection. Another strategy gauges the duration it takes for microbubbles to traverse a vascular bed of interest. These time measures can be advantageous in assessing both widespread and localized liver diseases and kidney conditions.

Information derived from time-intensity curves after microbubbles' bolus intravenous injections forms sensitive early indicators of how tumours respond to anti-vascular medications. This method, called dynamic contrast-enhanced ultrasound (DCE-US), has gained recognition as a valid procedure for monitoring tumour response by several authorities. [20]

4.1 Doppler

In its very classic form, Doppler ultrasound method measures the change in frequency in pulses of long or continues character thanks to reflection from a moving target. The difference between sent and received bounced signal is processed to determine direction of the flow and the speed of moving fluids.

It has limitations during movement of the tissue (cardiac activity and breathing), when frequency shifts are indistinguishable from blood flow. Especially slow flows under 1 mm/s in capillaries.

4.2 Microbubbles

In sonography microbubbles are used for enhancing a quality of ultrasound signal. First times date back to 60s. Specialist at that time was Dr. Raymond Gramiak from Rochester University, USA. His first attempts connected echography units to oscilloscope, which help him with demonstration in ultrasound imaging with B-mod and M-mode. He was among first who propagate usage of microbubbles in cardiac examination.

Microbubbles are small complex units with size of diameter from 1 - 10 μm , nowadays even tens and hundreds of nanometers. Microbubbles should meet conditions of biodegradability and biocompatibility. A typical microbubble have a shell and gas core. The shell can be composed of surfactants, proteins, lipids, polymers or combination of mentioned materials.

Microbubbles after injection into vascular system are less durable and more unstable, especially when transporting contrast agents or drugs. Its stability is mostly influenced by internal environment. Most decisive factors are pressure and temperature. With higher pressure, which occurs in a heart and vessels during systole make sustainability requirements more demanding. Another important and decisive factor in stability of the molecule is concentration of dissolved inner gas with impact on surface tension and liquid-gas interface concentration gradient. Solubility and inner gas composition also have their role. Microbubbles are divided into three groups.

1. Air-filled microbubbles

Inner structure of the first generation of microbubbles consist of air. This bubbles had couple disadvantages, namely high solubility in blood what instantly led to filtration via lungs affecting lifespan of microbubbles. This problem was partly solved by adding surface active substance which bonds into bubble surface prolonging lifespan for couple of seconds. At first very basic physiological saline was used then followed Almubex®, Levovist® and Sonogel®.

2. Transpulmonary vascular (with lifespan < 5 min)

Second-generation contrast agents contain microbubbles that are stable and

small enough to enter the systemic circulation. After intravenous administration, they increase the Doppler signal in various vessels. However, their contrast effect ends after a few minutes because their lifetime is not very long, usually less than 5 minutes. Most of the molecules include Fluor atoms in the form of perfluorocarbons or sulphur hexafluoride. This generation is commercially available, known as Optison®, SonoGen®, and SonoVue®.

3. Transpulmonary vascular (with lifespan > 5 min)

Third generation of microbubbles uses physiologically inert gases with low solubility capable of prolonging time lifespan in bloodstream over 5 minutes. Third generation is used in almost every organ of human body reachable for ultrasound diagnostic devices, also providing better stability and echogenicity in B-mode. An example of the third generation is EchoGen®.

4.2.1 Bolus tracking monitoring

The bolus monitoring method is based on the administration of a bolus in the form of a compact contrast substance. Next the substance is being focused and tracked in bloodstream. Strength of the signal is monitored in time. The goal of bolus tracking is to monitor the perfusion of an organ, which provides information about blood flow and oxygenation.

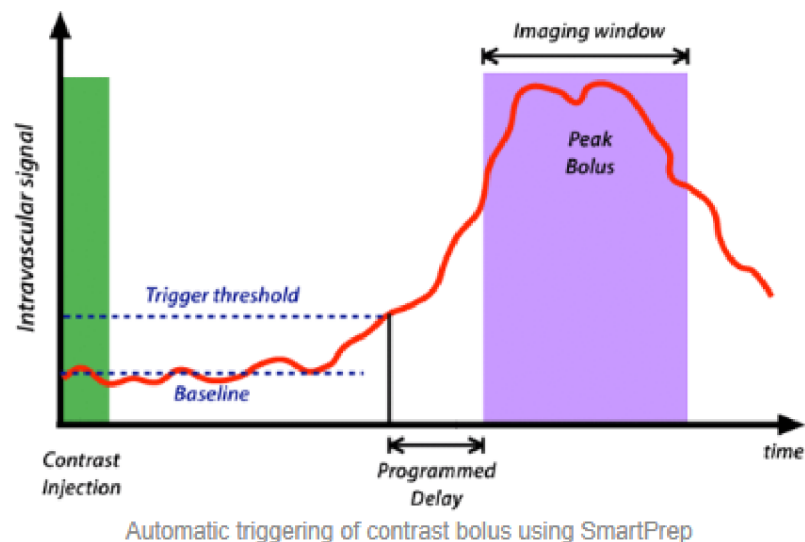


Fig. 4.1: Bolus tracking method [13]

Bolus tracking has a number of applications, including the evaluation of blood flow in the brain, liver, and kidney, and the assessment of organ function in patients with various medical conditions such as stroke, tumors, and heart disease. Bolus tracking can also be used to guide interventional procedures, such as catheterization and embolization, by providing information about the flow of contrast agent in real-time.

4.2.2 Reperfusion method

Reperfusion monitors blood flow through kidney tissue, liver and myocardium. This method uses a combination of very intensive destructive pulses at the beginning which destroy microbubbles within. Then the scanner switches to an imaging mode with low mechanical index. Next monitoring of microbubbles refill takes place. Refill is represented by raising slope β - velocity of inflow, while A refers to fractional vascular volume.

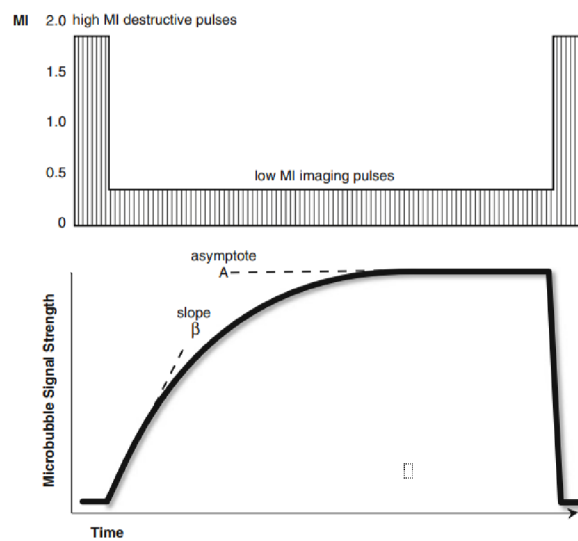


Fig. 4.2: Reperfusion [20]

4.3 Preparation

The first encapsulated microbubbles were prepared by aerating a sample of the patient's blood, and is still the simplest and most commonly used method for preparing microbubbles. Today a mechanical shaker, low-frequency ultrasound or coaxial electrohydrodynamic atomization (CEHDA), microfluid T-junction, higher shear rate emulsification or membrane emulsification is used.

5 Implementation of data in models, python

This section provides a detailed overview of the program's design, formal structure, and functional capabilities. The program is organized into Python packages comprising various classes, demonstrating adherence to the principles of Object-Oriented Programming (OOP). Execution is initiated via the `main.py` script.

Upon launch, the program autonomously loads data present in `.mat` files into a `Data` object. As part of this loading process, signals are filtered and prepared for future utilization. The `evolutionary` package encompasses the evolutionary algorithms employed within the program.

The models outlined in Section 3.1 can be located within the `functions` package, constituting the fundamental functional units of the program.

5.1 Libraries

PySwarms	An open-source research toolkit for Particle Swarm Optimization (PSO) in Python. Chosen for projects dealing with optimization problems.
Scipy	A Python library for scientific and technical computing. Essential for projects involving mathematical computations.
Sklearn	Also known as Scikit-Learn, this library is popular for machine learning in Python. Useful for projects involving building predictive models.
Numpy	Adds support for large, multi-dimensional arrays and matrices in Python, along with numerous high-level mathematical functions. Fundamental for scientific computing in Python.

5.2 Project structure

data	where data, load functions are stored.
error	directory for statistical measurement functions
evolutionary	contains evolutionary algorithms
functions	directory with models
modules	contains a different modules for processing previous structures and classes

5.3 Flow Diagram

The flow diagram below represents how program works. More comments can be found in code project.

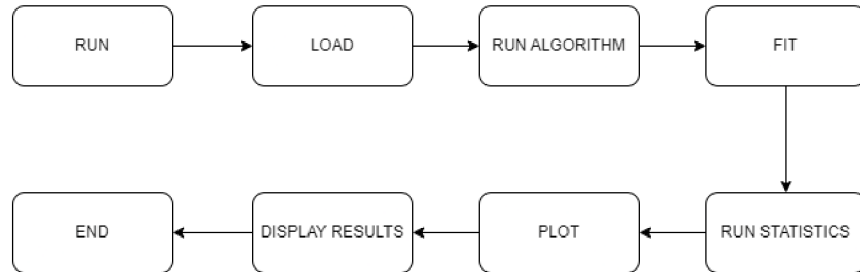


Fig. 5.1: Flow Diagram

5.4 Used data

This diploma thesis incorporates two distinct datasets. Each of these datasets has been individually prepared for subsequent use. Both datasets consist of .mat files, a format commonly associated with MATLAB. However, Dataset 1 possesses different properties compared to Dataset 2, necessitating the creation of two distinct functions for data loading. To facilitate further use, these datasets have been converted to a format readable by Python using the pymatreader library.

Dataset 1:

Phantom data was created using a dialyzer cartridge (for dialysis) and two stronger tubes with a constant flow in a non-recirculating system. The experiment was repeated for 4 different flow rates. Sonovue was used as a contrast agent in an appropriately diluted concentration to prevent significant attenuation on the microbubbles. Imaging was performed using the GE System FiVe with a 2.5 MHz sector probe in harmonic mode.

- exp13_aifnova_inp_tis_111017.mat
- exp13_roinove_inp_tis_111017.mat
- exp13_roivelke_inp_tis_111007.mat
- exp14_AIF2_inp_tis_111010.mat
- exp14_roimale2_inp_tis_111010.mat
- exp14_roivelke_inp_tis_110907.mat
- exp15_AIF_inp_tis_111007.mat
- exp15_roimale_inp_tis_111007.mat

- exp15_roivelke_inp_tis_111007.mat
- exp17_AIF_inp_tis_111007.mat
- exp17_roimale_inp_tis_111007.mat
- exp17_roivelke_inp_tis_111007.mat

Dataset 2:

Images of a pig’s myocardium were taken directly on the heart in an open chest setup using the GE Vivid 7 system with a 3.5 MHz linear probe. The images were taken in what is called the short axis view and were keyed using an ECG to display the maximum filling of the ventricles.

- per02_2_13_trig_DR60_inp_con_mreg_121113.mat
- per02_2_4_trig_DR60_inp_con_mreg_121113.mat
- per02_2_8_trig_DR60_inp_con_mreg_121113.mat
- per02_3_2_trig_DR60_inp_con_mreg_121120.mat
- per02_3_7_trig_DR60_inp_con_mreg_121120.mat
- per02_3_8_trig_DR50_inp_con_mreg_130226.mat

5.5 Evaluation

When evaluating the performance of predictive models, it is important to use appropriate evaluation metrics that provide valuable insights into their effectiveness. R-squared, Spearman Correlation, and NRMSE (Normalized Root Mean Squared Error) are three commonly used measures that offer information about the accuracy, relationship strength, and predictive power of models. Understanding these metrics is essential for researchers, data scientists, and decision-makers who want to effectively evaluate and compare different models.

5.5.1 R-squared

It is also known as the coefficient of determination, measures the proportion of the variance in the dependent variable that can be explained by the independent variables in a regression model. It ranges from 0 to 1, with higher values indicating a better fit to the data. A high R-squared suggests that a larger portion of the observed variation is captured by the model, demonstrating its ability to explain and predict the outcome variable. However, R-squared alone does not guarantee model validity, as it can be influenced by the number of variables and their interdependencies, The R-squared (coefficient of determination) is calculated as 5.1:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.1)$$

y_i : Observed values of the dependent variable

\hat{y}_i : Predicted values by the regression model

\bar{y} : Mean of the observed values

n : Number of data points

5.5.2 NRMSE (Normalized Root Mean Squared Error)

NRMSE calculates the normalized error between predicted and actual values, offering a measure of predictive accuracy. It is derived from the Root Mean Squared Error (RMSE), which quantifies the average deviation between predicted and actual values. NRMSE normalizes the RMSE by dividing it by the range of the data, providing a relative measure of error. A lower NRMSE indicates better predictive performance, as it represents a smaller average error relative to the data range.

$$NRMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{\max(y) - \min(y)} \right)^2} \quad (5.2)$$

y_i : Observed values of the dependent variable

\hat{y}_i : Predicted values of the dependent variable

n : Number of data points

$\max(y)$: Maximum value of the observed values

$\min(y)$: Minimum value of the observed values

5.5.3 Spearman Correlation

Spearman Correlation measures the monotonic relationship between two variables, providing insights into their association regardless of linearity. It ranges from -1 to 1, with values close to -1 or 1 indicating a strong negative or positive monotonic relationship, respectively. A Spearman Correlation of 0 suggests no monotonic relationship. Unlike R-squared, Spearman Correlation focuses on the order of the data points rather than the actual values, making it suitable for assessing non-linear relationships.

$$\rho = 1 - \frac{6 \sum_{i=1}^n (R_i - S_i)^2}{n(n^2 - 1)} \quad (5.3)$$

ρ : Spearman Correlation coefficient

R_i : Ranks of the observed values for the first variable

S_i : Ranks of the observed values for the second variable

n : Number of data points

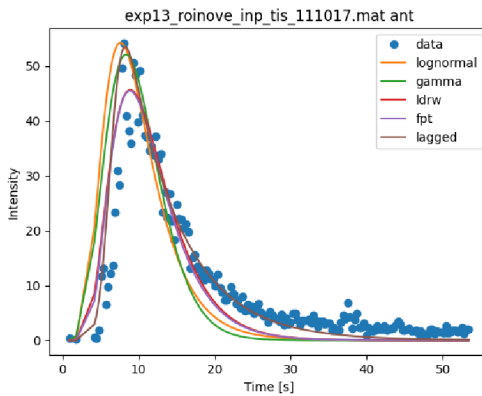
5.6 Results

In this study, we compared the performance of six metaheuristic algorithms, namely the cuckoo search algorithm, spider monkey algorithm, artificial bee colony, firefly algorithm, pso, and ant colony algorithm on a set of benchmark functions. The results showed that all six algorithms were able to find good solutions to the optimization problems, but with varying levels of efficiency.

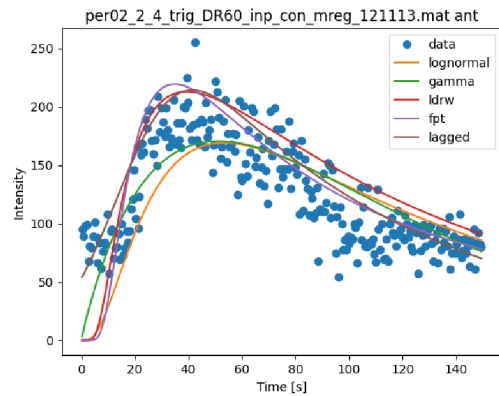
5.6.1 Graphical representation of results

The images below represent the application of each algorithm, with all five models fitted to the data. Subsections are divided based on whether they present results from the implemented algorithm or a commonly-used library algorithm for data fitting. These two approaches will be compared in this thesis.

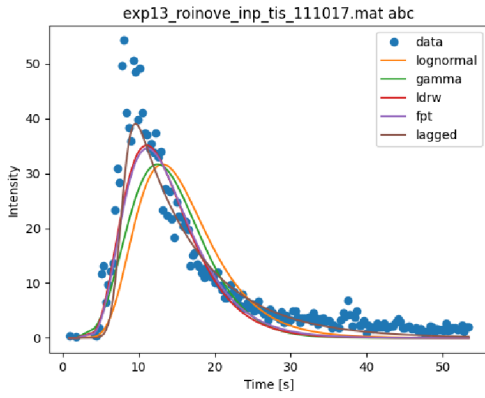
implemented algorithms



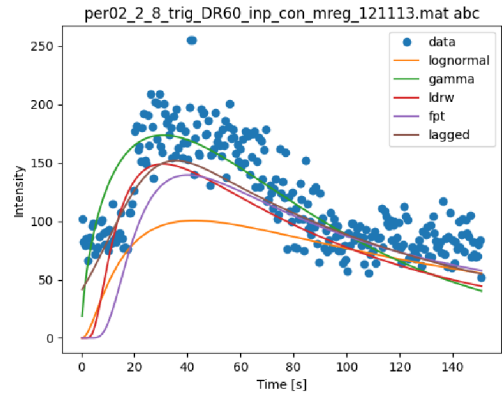
(a) Ant example - dataset 1



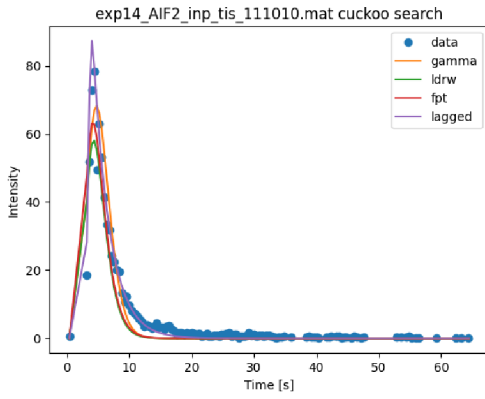
(b) Ant example - dataset 2



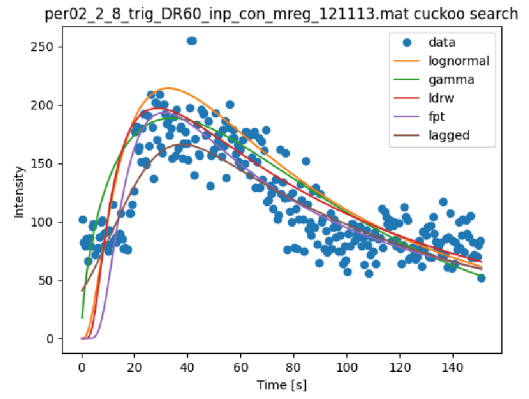
(a) ABC example - dataset 1



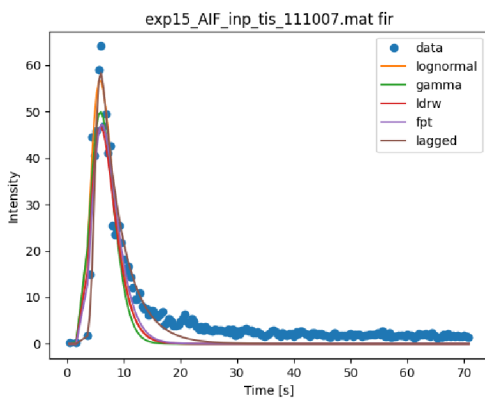
(b) ABC example - dataset 2



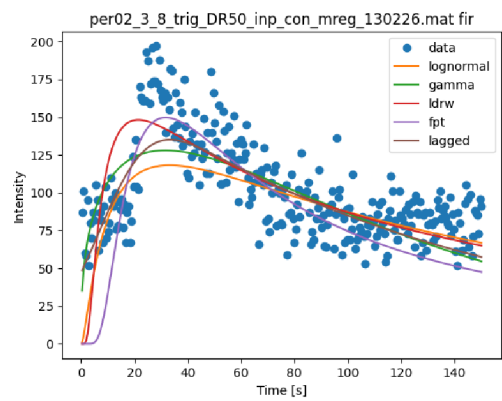
(a) CS example - dataset 1



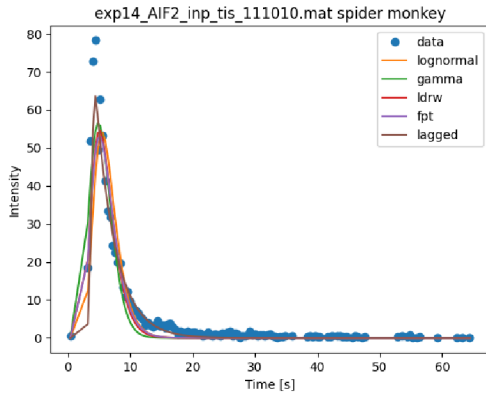
(b) CS example - dataset 2



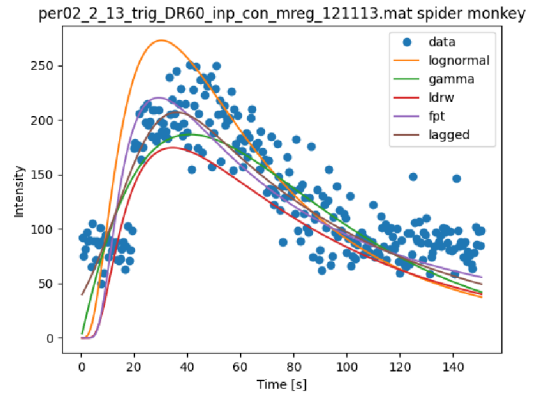
(a) FIR example - dataset 1



(b) FIR example - dataset 2

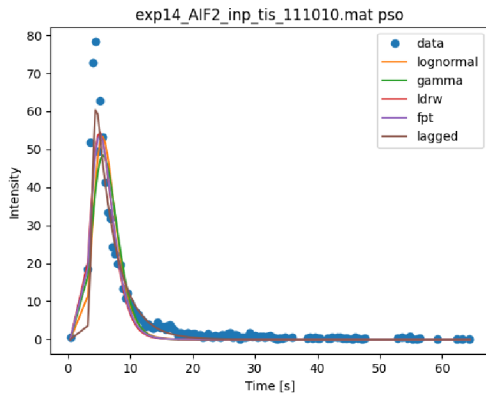


(a) SM example - dataset 1

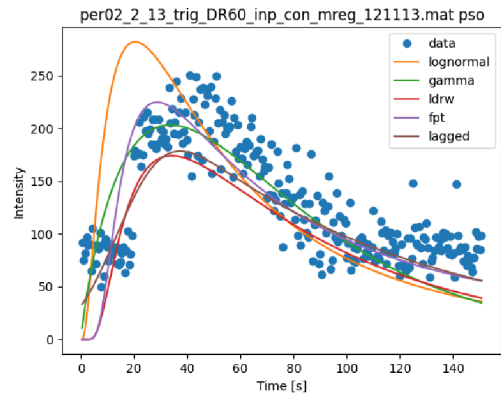


(b) SM example - dataset 2

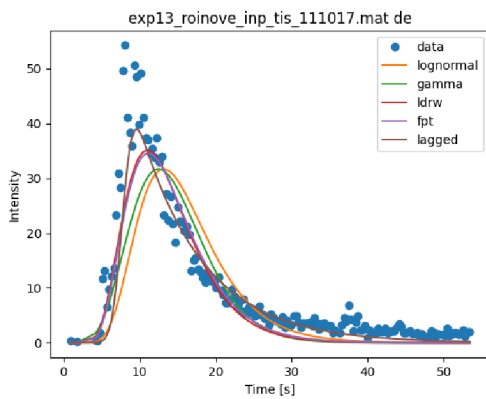
library algorithms



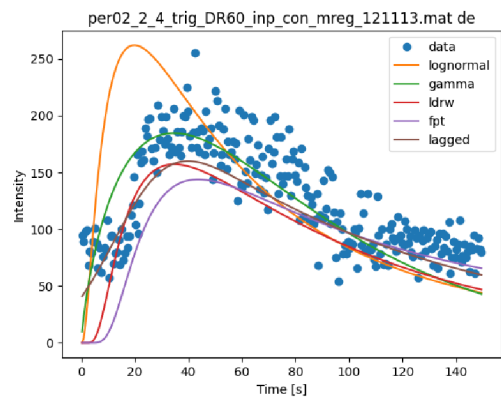
(a) PSO example - dataset 1



(b) PSO example - dataset 2



(a) DE example - dataset 1



(b) DE example - dataset 2

5.6.2 Statistical results

In this section, we will present the statistical analysis of our dataset. This analysis includes a variety of metrics aimed at providing a comprehensive understanding of the underlying patterns and characteristics of the data.

The statistics presented in the tables below represent the mean averages computed over the entire dataset. These statistics are indicative of the overall trends and can provide us with a general understanding of the data's behavior.

The analysis of these statistical measures will provide a broad and comprehensive understanding of the data, which is crucial for further processing, modeling, and interpretation.

Results from dataset 1

Implemented algorithms

ants = 20, iterations = 100, decay = 0.1, alpha = 1, beta = 2			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	0.08152	0.8191	0.7027
Gamma Variate	-0.0313	0.5870	0.8214
LDWR	0.7554	0.9001	0.4870
FPT	0.2071	0.8710	0.6403
Lagged Normal	0.2347	0.8709	0.7027

Tab. 5.1: Ant Colony Algorithm, Dataset 1

The table presents the results of the Ant Colony Algorithm applied to Dataset 1, with specific parameter settings. The algorithm was run for 100 iterations with 20 ants, using a decay rate of 0.1, an alpha value of 1, and a beta value of 2. In summary, the Ant Colony Algorithm applied to Dataset 1 with the given parameters yielded varying results for different models. The LDWR model demonstrated the best overall performance, exhibiting a strong fit to the data and the lowest error. The Log Normal, FPT, and Lagged Normal models achieved moderate fits, while the Gamma Variate model struggled to capture the underlying patterns in the data.

bees = 15, iteration = 100, trials = 10			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.0471	0.4131	0.7425
Gamma Variate	-0.0150	0.4995	0.8102
LDWR	0.7270	0.8802	0.5077
FPT	0.2015	0.8588	0.6410
Lagged Normal	0.2082	0.8472	0.6200

Tab. 5.2: Artificial Bee Colony Algorithm, Dataset 1

From the results, it is clear that the LDWR model exhibits the strongest performance across the three metrics. It has a notably high R-squared value of 0.7270, suggesting a good fit and strong predictive power. The Spearman correlation is also the highest amongst all models at 0.8802, indicating a strong monotonic relationship. Moreover, LDWR's Normalized Root Mean Square Error (NRMSE) is the lowest at 0.5077, indicating the least amount of error. On the other hand, the Log Normal and Gamma Variate models exhibit poor performance, indicated by their negative R-squared values and high NRMSE values. These results suggest that these models may not be suitable for this dataset. The FPT and Lagged Normal models show moderate performance with positive R-squared values and relatively lower NRMSE values when compared to Log Normal and Gamma Variate. In conclusion, based on the statistical measures, the LDWR model appears to provide the best fit and predictive accuracy for Dataset 1 when using the Artificial Bee Colony Algorithm with the specified parameters.

nests = 25, iterations = 100, pa = 1.0			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	0.7716	0.8971	0.4927
Gamma Variate	0.2378	0.6277	0.5948
LDWR	0.8460	0.9134	0.4057
FPT	0.2679	0.8769	0.5647
Lagged Normal	0.3146	0.8907	0.4895

Tab. 5.3: Cuckoo Search, Dataset 1

The LDWR model demonstrates the strongest performance with an R-squared value of 0.8460, indicating a very good fit between the predicted and actual values. It also shows the highest Spearman correlation (0.9134), suggesting a strong monotonic

relationship. The NRMSE value for LDWR is the lowest at 0.4057, implying the smallest prediction error among the models.

The Log Normal model, surprisingly, also exhibits strong performance with an R-squared value of 0.7716 and a high Spearman correlation of 0.8971. It has a moderately low NRMSE value (0.4927), indicating a reasonable accuracy in prediction.

The Gamma Variate, FPT, and Lagged Normal models show lower R-squared values (0.2378, 0.2679, and 0.3146, respectively), suggesting a weaker fit to the data compared to LDWR and Log Normal. However, their Spearman correlations are relatively high, indicating a decent monotonic relationship.

In summary, LDWR and Log Normal appear to offer the most reliable predictive performance for Dataset 1 under the Cuckoo Search algorithm with the given parameters.

fireflies=10, max iter=100, $\alpha=0.5$, $\beta=0.5$, $\gamma=1.0$			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	0.3731	0.7464	0.5468
Gamma Variate	0.2265	0.6159	0.5940
LDWR	0.8212	0.9016	0.4159
FPT	0.2635	0.8715	0.5664
Lagged Normal	0.3078	0.8892	0.5022

Tab. 5.4: Firefly Algorithm, Dataset 1

The Gamma Variate, FPT, and Lagged Normal models exhibit lower R-squared values, implying weaker predictive accuracy compared to the LDWR and Log Normal models. However, their Spearman correlations are relatively high, indicating a decent monotonic relationship.

In summary, when employing the Firefly Algorithm with the provided parameters on Dataset 1, the LDWR model shows the best overall performance in terms of fit, correlation, and prediction error. However, all these statistical results should be interpreted in the context of the specific dataset and problem at hand.

monkeys=30, spiders=10, radius=0.2, $\alpha=0.1$, $\gamma=0.1$, $\beta=2$			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	0.4218	0.6632	0.6155
Gamma Variate	0.0796	0.5198	0.6642
LDWR	0.7544	0.8855	0.4738
FPT	0.2290	0.8638	0.5919
Lagged Normal	0.2630	0.8730	0.5659

Tab. 5.5: Spider Monkey, Dataset 1

The table summarizes the performance of different models on Dataset 1 for predicting Spider Monkey behavior. The LDWR model outperforms other models with the highest R^2 value of 0.7544, strong Spearman correlation coefficient of 0.8855, and the lowest NRMSE of 0.4738. The Log Normal model also performs reasonably well with an R^2 of 0.4218 and a moderate Spearman correlation of 0.6632. The Gamma Variate, FPT, and Lagged Normal models have lower performance in terms of R^2 , Spearman correlation, and NRMSE.

Library algorithms

maxiter=100, swarmsize=10			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.0641	0.4137	0.7385
Gamma Variate	-0.0150	0.4995	0.8103
LDWR	0.7401	0.8785	0.5003
FPT	0.2073	0.8597	0.6310
Lagged Normal	0.2107	0.8486	0.6200

Tab. 5.6: PSO, Dataset 1

The table shows the performance of different models on Dataset 1 using the Particle Swarm Optimization (PSO) algorithm with maxiter=100 and swarmsize=10. The LDWR model performs the best with a high R^2 of 0.7401, strong Spearman correlation of 0.8785, and low NRMSE of 0.5003. The Log Normal and Gamma Variate models perform poorly, with negative R^2 values and relatively high NRMSE values. The FPT and Lagged Normal models have moderate performance with positive R^2 values and similar NRMSE values. In summary, the LDWR model outperforms other models in accurately predicting Spider Monkey behavior, while the Log Normal and Gamma Variate models perform poorly using the PSO algorithm.

max iterations = 100			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.0471	0.4131	0.7426
Gamma Variate	-0.015	0.4995	0.8103
LDWR	0.7271	0.8802	0.5077
FPT	0.2016	0.8588	0.641
Lagged Normal	0.2081	0.847	0.641

Tab. 5.7: DE, Dataset 1

The performance of the Log Normal and Gamma Variate models is particularly poor, as evidenced by their negative R^2 values, low Spearman correlations, and high NRMSE values. These results suggest that these models fail to accurately capture the relationships inherent in the data. Contrastingly, the LDWR model shows a notable improvement with a high R^2 value and Spearman correlation. This implies a strong fit and an effective capture of rank correlation, respectively. Although

the NRMSE value isn't as low as desired, it's significantly better than the aforementioned models. The FPT and Lagged Normal models exhibit moderate performance. While their R^2 values are low, their high Spearman correlations indicate that these models successfully capture the rank order of the data. However, the relatively high NRMSE values suggest larger prediction errors.

Results from dataset 2

Implemented algorithms

ants=50, iter=100, $\alpha=1$, $\beta=1$, evaporation=0.5, Q=100, $\tau=2$			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.4273	0.7155	0.3189
Gamma Variate	0.5123	0.7786	0.2481
LDWR	0.2240	0.7775	0.3136
FPT	0.3377	0.7804	0.2841
Lagged Normal	0.5695	0.7816	0.3189

Tab. 5.8: Ant Colony Algorithm, Dataset 2

Here in a table, the Lagged Normal model demonstrated the strongest fit to Dataset 2 using the Ant Colony Algorithm, based on the highest R^2 value. However, all the models had relatively low R^2 values, suggesting that they may not provide a strong fit to the data. The Gamma Variate and FPT models also showed reasonably good performance, while the Log Normal and LDWR models had weaker fits.

bees = 15, iteration = 100, trials = 10			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.6659	0.6978	0.3527
Gamma Variate	0.5360	0.7752	0.2446
LDWR	0.2679	0.8273	0.3073
FPT	0.1052	0.8147	0.3397
Lagged Normal	0.5900	0.8370	0.2299

Tab. 5.9: Artificial Bee Colony, Dataset 2

Overall the Lagged Normal model showed the strongest fit to Dataset 2 using the Artificial Bee Colony algorithm, based on the highest R^2 value, Spearman correlation, and the lowest NRMSE. The Gamma Variate model also showed reasonably good performance, while the Log Normal, LDWR, and FPT models had weaker fits to the data.

nests = 25, iterations = 100, pa = 1.0			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.0426	0.7235	0.2808
Gamma Variate	0.6574	0.8239	0.2102
LDWR	0.4420	0.8269	0.2683
FPT	0.5414	0.8471	0.2432
Lagged Normal	0.7303	0.8374	0.1865

Tab. 5.10: Cuckoo search, Dataset 2

Among the models, the Lagged Normal model performed the best, achieving the highest R^2 value of 0.7303, indicating a strong fit to the data. It also had a high Spearman correlation of 0.8374 and the lowest NRMSE of 0.1865. The Gamma Variate, LDWR, and FPT models also showed reasonably good performance with moderate to high R^2 values and Spearman correlations. On the other hand, the Log Normal model had the weakest fit to the data, reflected by a negative R^2 value. Summarizing it, the Lagged Normal model exhibited the strongest performance, while the Log Normal model had the poorest fit among the models evaluated.

fireflies=10, max iter=100, $\alpha=0.5$, $\beta=0.5$, $\gamma=1.0$			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.3892	0.6331	0.3225
Gamma Variate	0.5859	0.7929	0.2311
LDWR	0.4384	0.8304	0.2691
FPT	0.3934	0.8487	0.2797
Lagged Normal	0.6198	0.8391	0.2214

Tab. 5.11: Firefly algorithm, Dataset 2

Again, the Lagged Normal model showed the highest level of accuracy, as indicated by its superior R^2 value of 0.6198, signifying a strong fit to the data. It also exhibited a high Spearman correlation of 0.8391 and the lowest NRMSE of 0.2214. The Gamma Variate and LDWR models also demonstrated reasonably good performance with moderate R^2 values and strong Spearman correlations. Conversely, the Log Normal and FPT models exhibited weaker fits to the data.

monkeys=30, spiders=10, radius=0.2, $\alpha=0.1$, $\gamma=0.1$, $\beta=2$			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.6263	0.5949	0.3345
Gamma Variate	0.6308	0.8324	0.2182
LDWR	0.5019	0.8032	0.2535
FPT	0.4922	0.8453	0.2559
Lagged Normal	0.7053	0.8223	0.1949

Tab. 5.12: Spider Monkey, Dataset 2

The Lagged Normal model had the best fit to the data with an R^2 value of 0.7053, a Spearman correlation of 0.8223 and the lowest NRMSE of 0.1949. The Gamma Variate, LDWR and FPT models also performed well with R^2 values between 0.4922 and 0.6308 and strong Spearman correlations. However, the Log Normal model had the weakest fit with a negative R^2 value of -0.6263. Overall, the Lagged Normal model was the best performer while the Log Normal model had the poorest fit among the models evaluated using the Spider Monkey algorithm.

Library Algorithms

maxiter=100, swarmsize=10			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.7105	0.6924	0.3502
Gamma Variate	0.5360	0.7752	0.2446
LDWR	0.2678	0.8273	0.3073
FPT	0.1052	0.8147	0.3397
Lagged Normal	0.7017	0.8202	0.1961

Tab. 5.13: PSO, Dataset 2

In Optimization (PSO) algorithm, also The Lagged Normal model achieved the highest R^2 value of 0.7017, indicating a strong fit to the data. It also had the lowest NRMSE of 0.1961, suggesting accurate predictions. The Log Normal and FPT models had the weakest fits with negative R^2 values and relatively high NRMSE values.

max iterations = 100			
Model	R^2	Spearman Correlation	NRMSE
Log Normal	-0.6662	0.6977	0.3527
Gamma Variate	0.4662	0.7346	0.2605
LDWR	0.2089	0.7619	0.3177
FPT	0.0612	0.753	0.3412
Lagged Normal	0.7017	0.8202	0.1961

Tab. 5.14: DE, Dataset 2

The Log Normal model underperforms with a negative R^2 value, while the Gamma Variate and LDWR models show modest performance. The FPT model also shows limited predictive power with a low R^2 value. The standout here is the Lagged Normal model, which showcases superior performance across all three metrics, namely R^2 , Spearman Correlation, and NRMSE, indicating a strong fit, good correlation, and accurate predictions.

Results in histograms

In histograms displayed below, I provide an informative visual summary of the performance metrics obtained from evolutionary algorithms. It allows us to analyze the distribution and variability of results, identify outliers or patterns, and compare the performance of different algorithms.

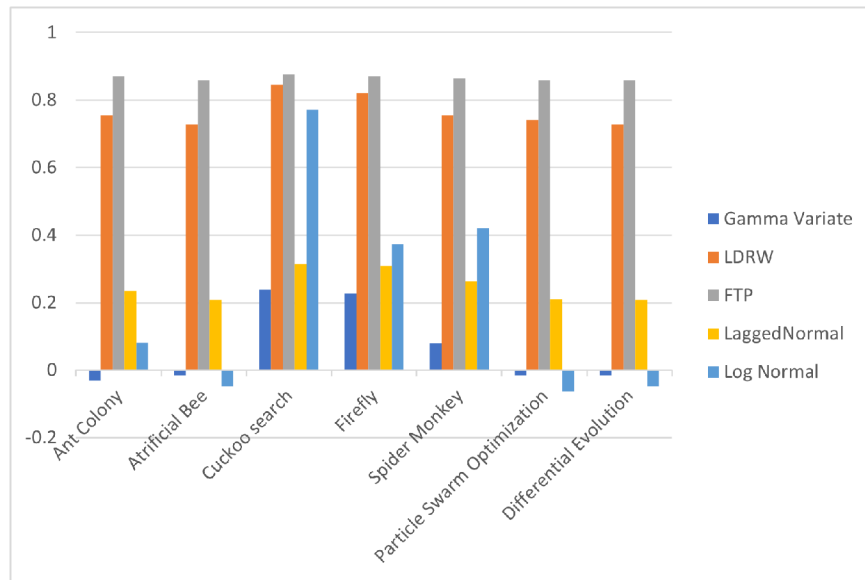


Fig. 5.9: R-squared - dataset 1

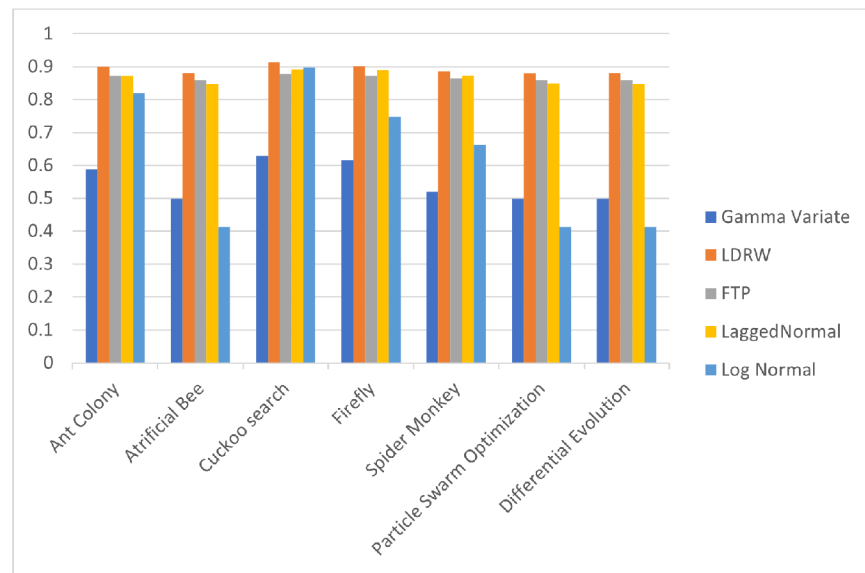


Fig. 5.10: Spearman - dataset 1

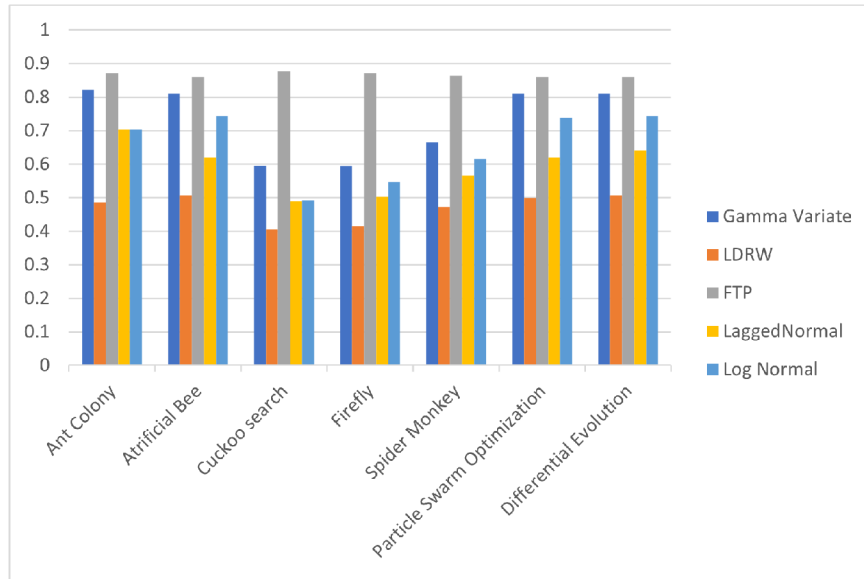


Fig. 5.11: NRMSE - dataset 1

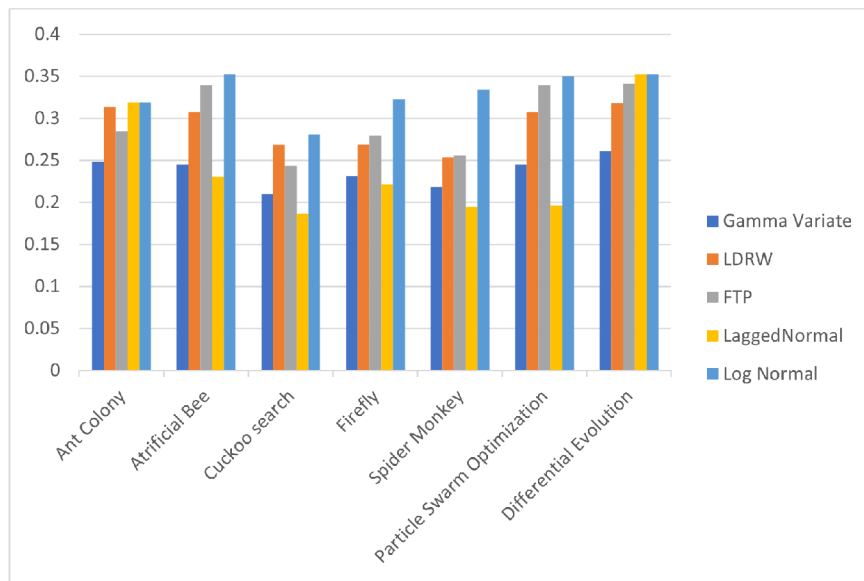


Fig. 5.12: R-squared - dataset 2

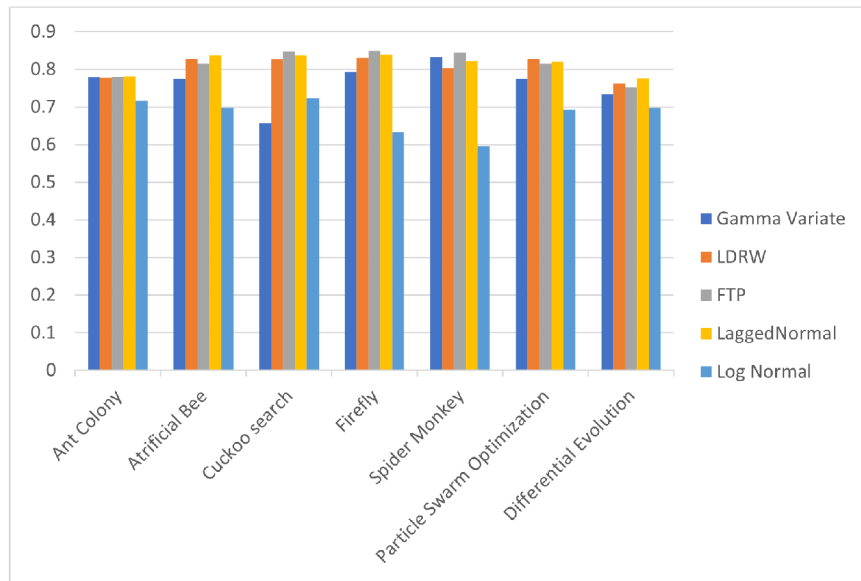


Fig. 5.13: Spearman - dataset 2

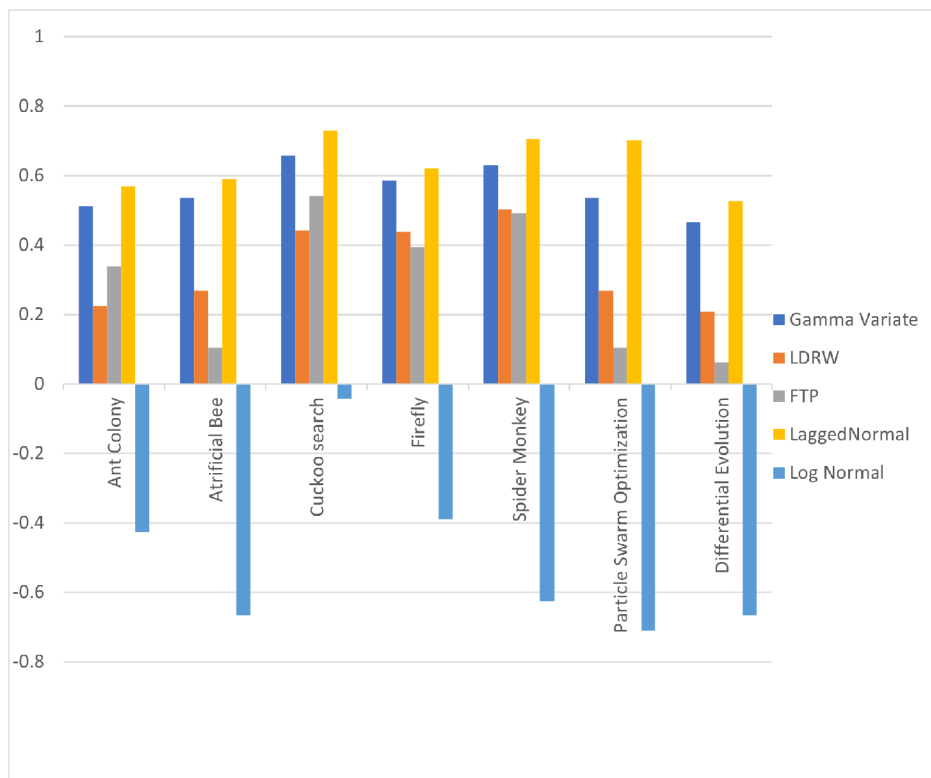


Fig. 5.14: NRMSE - dataset 2

The Final Conclusion

This study aimed to evaluate the performance of various models under the paradigm of different meta-heuristic optimization algorithms for a specific dataset (Dataset 2). The models evaluated included Log Normal, Gamma Variate, LDWR, FPT, and Lagged Normal. The meta-heuristic algorithms used for this purpose included Artificial Bee Colony, Cuckoo Search, Firefly Algorithm, Spider Monkey Optimization, Particle Swarm Optimization (PSO) and Differential Evolutionary (DE).

The empirical results derived from this research provide significant insights. The findings highlight the Lagged Normal model as a consistent performer across all the optimization algorithms in terms of the R^2 score and NRMSE. The R^2 score, being a measure of how well the model predictions conform to the actual outcomes, was highest for the Lagged Normal model across all algorithms. This indicates that this model was able to explain a larger proportion of variance in the dataset. Similarly, the NRMSE, a measure of the deviation of the predicted values from the observed values, was the lowest for the Lagged Normal model, indicating superior prediction accuracy.

Interestingly, the Log Normal model consistently showed poor performance, with a negative R^2 value across multiple algorithms, signifying that this model did not fit the dataset well. The Spearman Correlation, a non-parametric measure of rank correlation, indicated a strong relationship between the predicted and observed data for most of the models, with the FPT model showing the highest correlations in most of the algorithms.

In Dataset 1, the LDWR model using the Cuckoo Search algorithm appears to have performed the best across the metrics with an R^2 of 0.8460, Spearman correlation of 0.9134, and NRMSE of 0.4057. These values indicate a high degree of explanatory power, strong correlation, and low error, respectively. On the other hand, in Dataset 2, it is challenging to identify a clear winner among the algorithms.

Overall, the results of this thesis reinforce the utility of meta-heuristic optimization algorithms in enhancing the prediction accuracy of data models. More specifically, this study provides strong empirical evidence supporting the use of the Lagged Normal model when using these algorithms for the given dataset. However, these results are specific to the dataset used in this study, and extrapolation to other datasets should be undertaken with caution.

Future research may extend these findings by investigating the performance of these and other models under different optimization algorithms, or by applying the same models and algorithms to different datasets. Such studies would contribute to a more comprehensive understanding of the performance characteristics of these models and algorithms, and would potentially provide further insights to guide the

selection of appropriate models and algorithms in practice.

Bibliography

- [1] SCHONBACH, Christian, RANGANATHAN, Shoba a Kenta NAKAI, ed., 2018. Encyclopedia of Bioinformatics and Computational Biology [online]. 1. United States of America: Elsevier [cit. 2023-01-26]. ISBN 9780128114148. Dostupné z: <https://shop.elsevier.com/books/T/A/9780128114322full-description>
- [2] VOVK, Vladimir, 2013. Kernel Ridge Regression. Empirical Inference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013-10-9, 2018, 105-116. ISBN 978-3-642-41135-9. Dostupné z: doi:10.1007/978-3-642-41136-6₁₁
- [3] Support Vector Machines [online]. [cit. 2023-01-26]. Dostupné z: <https://scikit-learn.org/stable/modules/svm.html>
- [4] HOLLAND, John H., 1984. Genetic Algorithms and Adaptation. Adaptive Control of Ill-Defined Systems. Boston, MA: Springer US, 1984, 317-333. ISBN 978-1-4684-8943-9. Dostupné z: doi:10.1007/978-1-4684-8941-5₂₁
- [5] Towards datascience, 2020. Www.towardsdatascience.com [online]. [cit. 2023-01-29]. Dostupné z: <https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac>
- [6] BRABAZON, Anthony, Michael O'NEIL a Sean MCGARRAGHY, 2015. Natural Computing Algorithms. 1. Dublin: Springer-Verlag Berlin Heidelberg 2015. ISBN 978-3-662-43630-1.
- [7] Tutorialspoint.com. Www.tutorialspoint.com [online]. [cit. 2023-01-31]. Dostupné z: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm
- [8] AKHBARDEH, Alireza, Hersh SAGREIYA, Ahmed EL KAFFAS, Jürgen K. WILLMANN a Daniel L. RUBIN, 2019. A multi-model framework to estimate perfusion parameters using contrast-enhanced ultrasound imaging. Medical Physics. 46(2), 590-600. ISSN 0094-2405. Dostupné z: doi:10.1002/mp.13340
- [9] CORTES, Corinna a Vladimir VAPNIK, 1995. Support-vector networks. Machine Learning. 20(3), 273-297. ISSN 0885-6125. Dostupné z: doi:10.1007/BF00994018
- [10] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.
- [11] V. HARABIŠ, R. KOLÁŘ, M. MÉZL, R. JIŘÍK, Comparison and evaluation of indicator dilution models for bolus of ultrasound contrast agents. PHYSIOLOGICAL MEASUREMENT, 2013, roč. 34, č. 2, s. 151-162. ISSN: 0967- 3334.

- [12] PROFANT, M., K. VYSKA a U. ECKHARDT, 1978. The Stewart–Hamilton Equations and The Indicator Dilution Method. *SIAM Journal on Applied Mathematics*. 34(4), 666-675. ISSN 0036-1399. Dostupné z: doi:10.1137/0134053
- [13] Automatic triggering of contrast bolus using SmartPrep, 2021. In: *Mriquestions.com* [online]. Washington D.C: © 2021 AD Elster, ELSTER [cit. 2022-11-24]. Dostupné z: <https://www.mriquestions.com/timing-the-bolus.html>
- [14] YOU, Yang, James DEMMEL, Cho-Jui HSIEH a Richard VUDUC, 2018. Accurate, Fast and Scalable Kernel Ridge Regression on Parallel and Distributed Systems. *Proceedings of the 2018 International Conference on Supercomputing*. New York, NY, USA: ACM, 2018-06-12, 2018(8), 307-317. ISBN 9781450357838. Dostupné z: doi:10.1145/3205289.3205290
- [15] Drucker, Harris; Burges, Christ. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997); "", in *Advances in Neural Information Processing Systems 9*, NIPS 1996, 155–161, MIT Press.
- [16] *Evolutionary and Swarm Intelligence Algorithms*, 2019. 1. New Dehli: Springer-Link. ISBN 978-3-319-91341-4.
- [17] HU, Zhong-bo, Qing-hua SU, Sheng-wu XIONG a Fu-gao HU, 2008. Self-adaptive Hybrid differential evolution with simulated annealing algorithm for numerical optimization. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, 1(46), 1189-1194. ISBN 978-1-4244-1822-0. Dostupné z: doi:10.1109/CEC.2008.4630947
- [18] YANG, Xin-She, 2014. *Cuckoo Search and Firefly Algorithm*. 1. London: Springer International Publishing Switzerland 2014. ISBN 978-3-319-02140-9. Dostupné z: doi:10.1007/978-3-319-02141-6
- [19] DORIGO, Marco a Thomas G. STÜTZLE, 2004. *Ant colony optimization*. 1. Cambridge: MIT Press. ISBN 02-620-4219-3.
- [20] COSGROVE, David a Nathalie LASSAU, 2010. Imaging of perfusion using ultrasound. *European Journal of Nuclear Medicine and Molecular Imaging*. 37(S1), 65-85. ISSN 1619-7070. Dostupné z: doi:10.1007/s00259-010-1537-7
- [21] AKHBARDEH, Alireza, Hersh SAGREIYA, Ahmed EL KAFFAS, Jürgen K. WILLMANN a Daniel L. RUBIN, 2019. A multi-model framework to estimate perfusion parameters using contrast-enhanced ultrasound imaging. *Medical Physics*. 46(2), 590-600. ISSN 0094-2405. Dostupné z: doi:10.1002/mp.13340

- [22] YANG, Xin-She, 2009. Firefly Algorithms for Multimodal Optimization. Stochastic Algorithms: Foundations and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, 1(2), 169-178. Lecture Notes in Computer Science. ISBN 978-3-642-04943-9. Dostupné z: doi:10.1007/978-3-642-04944-6₁₄
- [23] FISTER, Iztok, Xin-She YANG, Dušan FISTER a Iztok FISTER, 2014. Firefly Algorithm: A Brief Review of the Expanding Literature. Cuckoo Search and Firefly Algorithm. Cham: Springer International Publishing, 2014-11-1, 1(1), 347-360. Studies in Computational Intelligence. ISBN 978-3-319-02140-9. Dostupné z: doi:10.1007/978-3-319-02141-6₁₇

List of symbols, physical constants and abbreviations

US	Ultrasound
TOI	Tissue of Interest
ROI	Region of interest
AIF	Arterial I function
EA	Evolutionary Algorithm
GA	Evolutionary Algorithm
AEA	Advanced Evolution Algorithms
CS	Cuskoo Search
SM	Spider Monkey
FIR	Firefly Algorithm
ACO	Ant Colony Optimization
AUC	Area Under The Curve

List of appendices

A Content

78

A Content

- project.zip
- diploma thesis.pdf
- dataset