PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

Series N°: ………. /2023

University of Kasdi Merbah Ouargla

Faculy of hydrocarbons, renewable energies and science of earth and universe

Drilling and MCP department

Thesis

To obtain Master’s degree

Speciality: Drilling

Presented by:

DIF Bader Eddine Badis
CHERIF Mohamed Aymen
FECHKEUR Mohammed Tahar

-TOPIC-

Prediction of gas flow rates from gas condensate reservoirs through wellhead chokes using Differential Evolution Metaheuristic Algorithm

Discussed in: 08/06/ 2023 in front the examining committee

<u>Jury members:</u>

HACHANA Oussama
KHENTOUT Abdelkader
FENAZI Bilal

Appreciation

At first we thank Allah who gave us the strenght and courage to complete this task.

Our gratitude goes to our supervisor Pr. Hachana for his guidance and expertise have been instrumental throughout this process. His insightful feedback and encouragment have helped shape the quality and direction of our work. We would also like to, extend our thanks to the faculty members and instructors who have imparted their knowledge and provided valuable resources during our academic journey.

Lastly, we would like to express our deepest appreciation to our families for their unwavering support and belief in our abilities, and all those who have contributed to the completion of this work.their support, guidance and belief in our potential have been vital to our achievement.

Dedication To:

My parents: source of our courage and inspiration.

To my grandmothers

To my brother and sisters;

To my uncles and aunts;

To all my family;

As well as all my friends without exception.

I dedicate this work to you . . . .

With my love Aymen.

Dedication To:

To my lovely mother : my source of courage and strength

To my father god may mercy on him soul

To my brothers ali and aissa

To my dearsest sisters

To my aunts fatiha and djamila khuiled

To my angle grandmother

As well as all our friends without exception.

I dedicate this work to you with all of

My love Tahar.

Dedication To:

To my source of tenderness, patience and motivation, without whom I would be nothing, who always give me hope to live and who sacrificed so much for me so that I could succeed in my life and for their prayers and encouragement in difficult times. to my dear mother and my dear father kaddour May God bless Them.

To my brothers “Ayman” and “Mohamed” and “Hocine” and my dearests sisters Words are not enough to thank them for their support, God bless Them

To my grandmother “Meriem”

to my uncles

I dedicate this work to you

with all of My love BaderEddine.

TABLE OF CONTENTS

LISTE OF FIGURES

LISTE OF TABLES

LIST OF SYMBOLS

DE :Differential Evolution .

$\mathbf{x}_i^t$:is the position of the i-th element in iteration t.

$X_{max,j}$: upper limit

$X_{min,j}$: lower limit.

$rand$: is a uniform distribution that can generate any real value between 0 and 1.

Y_i^t: is constructed from a mutation process on the basis of a given target vector of X_i^t

r_1: is the population index of the DE solution selected as the base vector.

r_2, r_3, r_4, r_5: are the population indices of DE solutions randomly selected to construct the mutant vector.

F : is a scaling factor used to control the mutation process and has a value in the range between [0, 1].

Cr : crossover rate.

$rand_{i,j}$:is a random number lies in the range [0, 1].

Z_i^t **:** is inherited from the donor vector Y_i^t.

$\langle n \rangle_D$:indicates a modulus function of D.

X_i^{t+1} **:** of the search process while retaining the population size of DE in every generation.

f(.) : is an operator used to determine the objective function or fitness value of an individual solution.

$v_{i,j}^{g+1}$: is the mutation vector.

$x_{r3,n}^g$ **:**is the base vector.

$\overrightarrow{X_{\imath,seed,g}}$ **:** is the seed of the target vector.

$F_{i,j,g}$:is the mutation factor for each generation.

ANN :Artificial neural Networks.

WOA : Whale optimization Algorithm.

FA :firefly algorithm.

ACO : Ant colony optimization.

GA : Genetic algorithm.

Qg: gas production rate.

Q_{scfd}: well flow rate (STB/D).

Q_{BPD}: well flow rate (STB/D).

Q: flow rate (STB/D).

P_{up}: upstream choke pressure (psi).

P_d: downstream choke pressure (psi).

ΔP: Pressure drop (m/Lt 2), psia.

GOR: gas oil ratio (SCF/STB).

LGR: Liquid-Gas ratio (L3 /L 3), Stb/MMscf.

D: choke diameter (1/64 in).

S: Choke size (L), 1/64 inch.

SpGr: liquid specific gravity with respect to water.

Wct : water cut (%) / Water cut (Wct.) = (Volume of water / Total volume of fluids) * 100.

PWF: flowing wellhead pressure (psi))(pressure).

γ_g :is gas specific gravity (without units).

T :is flowing fluid temperature (degrees R) Pup is upstream pressure (Psig).

$\beta_1$to β_5 :are unknown parameters.

a, b, c and d: Empirical unknown parameters.

K: is the specific heat ratio K=Cp/Cv (Gould, 1947).

NP: Number of population.

N_it: Number of iterations.

max_s: Maximum number of tests.

NFE: Number of function evaluations.

OF: Objective Function.

RMSE: root mean square error.

STD: Standard deviation

ABSTRACT:

The prediction of gas flow rates through wellhead chokes in condensate reservoirs involves considering various factors, including the properties of the hydrocarbon gas mixture, the choke design and specifications, reservoir conditions, and the interaction between the gas and natural gas liquids. These complex interactions influence the flow regime, pressure drop, and overall system behavior, making prediction a challenging task, the Researchers and engineers have developed various mathematical models, empirical correlations to predict gas flow rates through wellhead chokes in condensate reservoirs. These models often consider fundamental fluid mechanics principles, such as conservation of mass and energy, combined with empirical data from field measurements. The proposed models usually require a feasible parameter extraction of the unknown coefficients. This identification process could be achieved by means of an optimization method. Differential Evolution (DE) is one of the effective evolutionary algorithms used to solve global optimization problems in different domain. The basic concept of DE algorithms is built by three strategies: Mutation, Crossover and Selection. This study shows the importance and strength of this meta-heuristic technique. By using real datasets from two different wells and two different models, the findings by using DE algorithm have been compared with the results reached by means of ACO algorithm and others from a recently published paper.

ملخص:

يتضمن التنبؤ بمعدلات تدفق الغاز من خلال اختناقات رأس البئر في خزانات المكثفات النظر في عوامل مختلفة، بما في ذلك خصائص خليط الغاز الهيدروكربوني، وتصميم ومواصفات الاختناق، وظروف الخزان، والتفاعل بين الغاز وسوائل الغاز الطبيعي. تؤثر هذه التفاعلات المعقدة على نظام التدفق، وانخفاض الضغط، وسلوك النظام الكلي، مما يجعل التنبؤ مهمة صعبة، وقد طور الباحثون والمهندسون نماذج رياضية مختلفة، وارتباطات تجريبية للتنبؤ بمعدلات تدفق الغاز من خلال اختناقات رأس البئر في خزانات المكثفات. غالبًا ما تأخذ هذه النماذج في الاعتبار مبادئ ميكانيك السوائل الأساسية، مثل الحفاظ على الكتلة والطاقة، جنبًا إلى جنب مع البيانات التجريبية من القياسات الميدانية. عادة ما تتطلب النماذج المقترحة استخراج معاملات غير معروفة. ويمكن تحقيق عملية هذه عن طريق أسلوب التحسين الأمثل. التطور التفاضلي (DE) هو أحد الخوارزميات التطورية الفعالة المستخدمة لحل مشاكل التحسين العالمية في مجالات مختلفة. تم بناء المفهوم الأساسي لخوارزميات DE من خلال ثلاث استراتيجيات: الطفرة والتقاطع والاختيار. تُظهر هذه الدراسة أهمية وقوة هذه التقنية. باستخدام مجموعات بيانات حقيقية من بئرين مختلفين ونموذجين مختلفين، تمت مقارنة النتائج باستخدام خوارزمية DE بالنتائج التي تم التوصل إليها عن طريق خوارزمية ACO و غيرها من ورقة بحثية نشرت مؤخرا.

GENERAL INTRODUCTION:

The prediction of gas flow rates from condensate reservoirs through a wellhead choke is a crucial aspect in the field of petroleum engineering. Accurate estimation of gas flow rates plays a pivotal role in optimizing production processes and ensuring efficient reservoir management. In this context, the selection of an appropriate model is of high importance. Meanwhile, accurate models consider much more variables and hence further unknown parameters which should be well extracted. For this reason, the utilization of meta-heuristic algorithms has gained considerable attention due to their ability to tackle complex optimization problems. Among these algorithms, the differential evolution (DE) metaheuristic algorithm has proven to be effective in various applications.

The DE algorithm is a population-based optimization technique inspired by natural evolution processes. It utilizes a set of candidate solutions and iteratively evolves them through mutation, crossover, and selection operations to search for the optimal solution. By incorporating DE into the prediction of gas flow rates, significant improvements can be achieved in terms of accuracy and efficiency.

The primary objective of this study is to leverage the power of the DE metaheuristic algorithm for predicting gas flow rates through a wellhead choke from condensate reservoirs. The algorithm’s versatility allows it to explore the parameter space effectively and identify the most suitable settings for optimizing gas flow predictions. This is by considering several variants which depend on the optimal model selection.

Through rigorous experimentation and analysis, this study aims to validate the effectiveness of the DE metaheuristic algorithm in predicting gas flow rates from condensate reservoirs through a wellhead choke. The obtained results will be compared with existing prediction models and empirical data to evaluate the accuracy and reliability of the proposed approach. Ultimately, the successful implementation of the DE algorithm in this context would contribute to enhanced decision-making processes, improved production planning, and increased operational efficiency in the petroleum industry.

CHAPTER I: GAS FLOW THROUGH WELLHEAD CHOKES MODELING

I.1 Introduction:

Operating production wells and understanding multi-phase flow through wellhead chokes are crucial aspects of the oil and gas industry. Production wells are drilled to extract hydrocarbon resources from underground reservoirs. The efficient and effective operation of these wells is essential for maximizing oil and gas production. In order to control, organize and anticipate production we need understanding Multi-phase flow through wellhead chokes refers to the flow of multiple phases, such as gas, oil, and water, through constriction devices known as chokes located at the wellhead. This flow presents unique challenges due to the complex interactions between different phases and their impact on production. So that's Wellhead equipment which is a crucial component of the production process in the oil and gas industry. It refers to the collection of equipment installed at the top of a wellbore to control and facilitate the extraction of hydrocarbon resources from underground reservoirs. Wellhead equipment serves several essential functions, including providing a safe and controlled environment for production operations.

In order to efficiently predicting and understanding the fluid mixtures behavior during oil and gas production, an adequate modeling is of crucial importance. Several models could be considered for these purposes, which will be described in this chapter.

I.2 Gas condensate reservoirs:

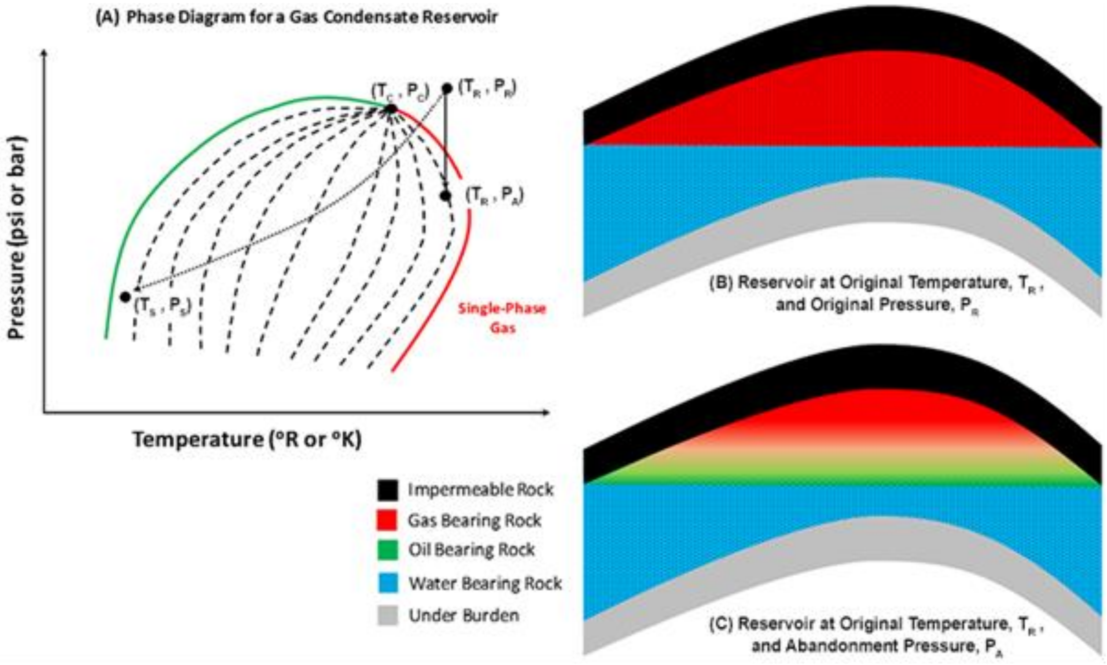
Gas condensate reservoirs typically contain a mixture of hydrocarbons, including natural gas, condensate, and sometimes oil. The composition of the gas and condensate varies depending on the specific reservoir and can include a range of hydrocarbons such as methane, ethane, propane, butane, and pentane, as well as heavier components like hexane and heptane. Gas condensate reservoir is a particular kind of reservoir which has a behavior mediating between that of gas and volatile oil reservoirs. It contains low density liquid hydrocarbons which present as gaseous components in the raw natural gas and condense out of gas when the pressure is lowered below the dew point pressure of hydrocarbons. Due to changes in temperature and pressure, these types of reservoir are bound to instability of flow regime, fluctuation and phase change which may lead to liquid holdup and phase separation. Therefore, multiphase flow is common in gas condensate reservoirs [1].

The flow rate of well is the most significant parameter for characterization of a reservoir and estimation of its behavior. Wellhead chokes (a configuration which an elbow is installed exactly upstream of chokes) normally are used in wells as controlling agents to adjust the flow rate and sustain adequate back pressure to avoid sand issues and water/gas coning. Therefore, estimation of choke performance by implementation and optimization of a relation between size of chokes and wellhead flow rates is only possible through a precise modeling and selection of an optimum choke size. Based on flow regime, the fluid flow through the choke can be characterized as either critical or sub-critical (sonic or subsonic, respectively).When Mach number is equal or more than unity (When the flow velocity is equal or greater than sonic velocity), any pressure disturbance wave from downstream cannot spread through upstream and mass flow rate reaches a maximum amount which only depends upon upstream conditions. This kind of flow is known as critical flow. Accordingly, to avoid any perturbation of this is essentially what is occurring in the reservoir of a gas condensate system but under isothermal conditions. As we pass through the dew point pressure, the heaviest hydrocarbon components in the system begin to drop out and form a second, liquid hydrocarbon phase in the two-phase region of the phase envelope inside the reservoir.

We were to follow the isothermal Path A-A’-A’’, then we would go below through the dew-point pressure, increase the volume percentage of the liquid hydrocarbon phase until it reached a maximum at Point A’ with further reductions in pressure resulting in a lower volume percentage of the liquid hydrocarbon phase. We could also continue the isothermal pressure reduction, reenter the single-phase gas region, and stop at Point A’’. The analogy for our multi-component system is that if we start at the point of maximum liquid volume (Point A’ in Figure (I.1) and reduced the pressure isothermally, then we would get the conventional behavior for a pure system along Path A’-A’’. Conversely, if we were to start at Point A’ and increased the pressure isothermally to Point A in the single-phase gas region, then we would get the behavior opposite of that for a pure system along Path A’-A. This behavior, opposite to a pure system, is referred to as retrograde behavior. This behavior occurs in the green shaded region in Figure (I.1). This region, formed by connecting all of the points of maximum temperature on the quality lines, is referred to as the retrograde region of the fluid.

I.3 Flow Regime and well head:

The flow through chokes is classified as subcritical or critical(choked) flow. Critical flow occurs when the flow velocity for the fluid reaches its sonic velocity. Pressure waves, caused by the change in pressure, propagate through the flow with the speed of sound. But if the fluid flows faster than the waves can travel, the waves are unable to move upstream, or have any effect on the flow upstream the restriction.


(A) Phase Diagram for a Gas Condensate Reservoir
(T_C , P_C)
(T_R , P_R)
(T_R , P_A)
(T_S , P_S)
Single-Phase Gas
Pressure (psi or bar)
Temperature (°R or °K)
Impermeable Rock
Gas Bearing Rock
Oil Bearing Rock
Water Bearing Rock
Under Burden
(B) Reservoir at Original Temperature, T_R , and Original Pressure, P_R
(C) Reservoir at Original Temperature, T_R , and Abandonment Pressure, P_A

Figure(I.1): Gas condensate reservoires.

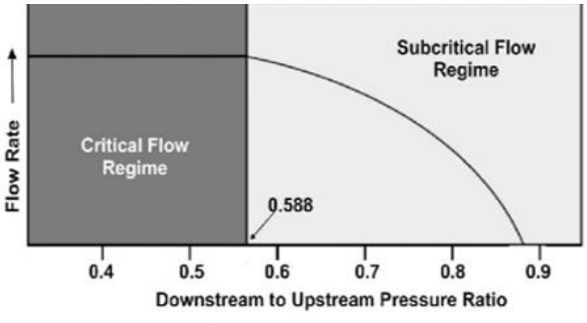
Once this border is reached, the flow is independent of downstream pressure, and the flow is critical. Any further reduction in the down-stream pressure will not increase the flow rate [2].

Critical flow is often desired across a wellhead choke because the well will not be influenced by the conditions in the downstream production pipeline (e.g., pressure fluctuations). This will reduce the risk of damaging the well and reservoir for an incompressible fluid, the sonic velocity is infinite, and critical flow cannot occur in practice. Many liquids are almost in compressible, or treated as that in modelling, and therefore critical flow in liquid-only flow is rare. However, if the downstream pressure is reduced below the bubble point; the liquid will start to vaporize, and critical flow is possible(Munson et al., 1998) [3].

A petroleum engineer's primary responsibility is to increase production lifetime by preventing extra production, controlling production through chokes, and increasing production or reservoir lifetime by selecting an appropriate choke on production wells. A wellhead choke is a tool that is installed through an inflow pipe path to resist pressure, limit and control production, prevent water and gas coning, and control pressure in order to keep the wellhead equipment in good working order [4]. The flow rate through the choke is determined by the wellhead pressure, choke diameter, before choke temperature, and water production rate, which includes free water, sediment water, emulsion, and gas oil ratio (Eq I.1).

$$\frac{P_d}{P_{up}} = \left(\frac{2}{K+1}\right)^{\frac{k}{k-1}} \qquad \text{(Eq I.1).}$$

The passing flow rate is primarily two-phase, and it can produce two types of flow when it passes through flow chokes: critical or sonic flow and sub-critical or sub-sonic flow.



Figure(I.2): Diagram determining critical and subcritical flow in wellhead choke[6].

Figure (I.2) shows that if the pressure rate after the choke (P2) to before the choke (P1) is less than 0.588, the flow is critical; otherwise, it is subcritical [5]. The flow in Iran oil wells of southern crude oil fields is critical, while some condensate gas wells are subcritical. Wellhead chokes which are installed through wells flow, are divided into two main groups that are:

- Positive or fixed choke.
- Adjustable or variable choke.

a) Positive or fixed chokes:

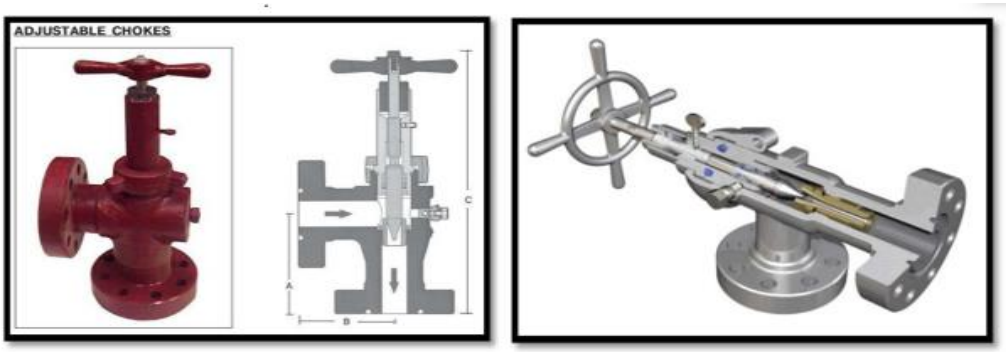
They are used when the production rate is constant for a long time and the production sand or fluid is corrosive Figure(I.3). This choke has the following properties:

- Constant bore diameter.
- Choke is Ceramic, Tangestan carbide or iron type.
- Choke length can be from 2 to 6 inches.

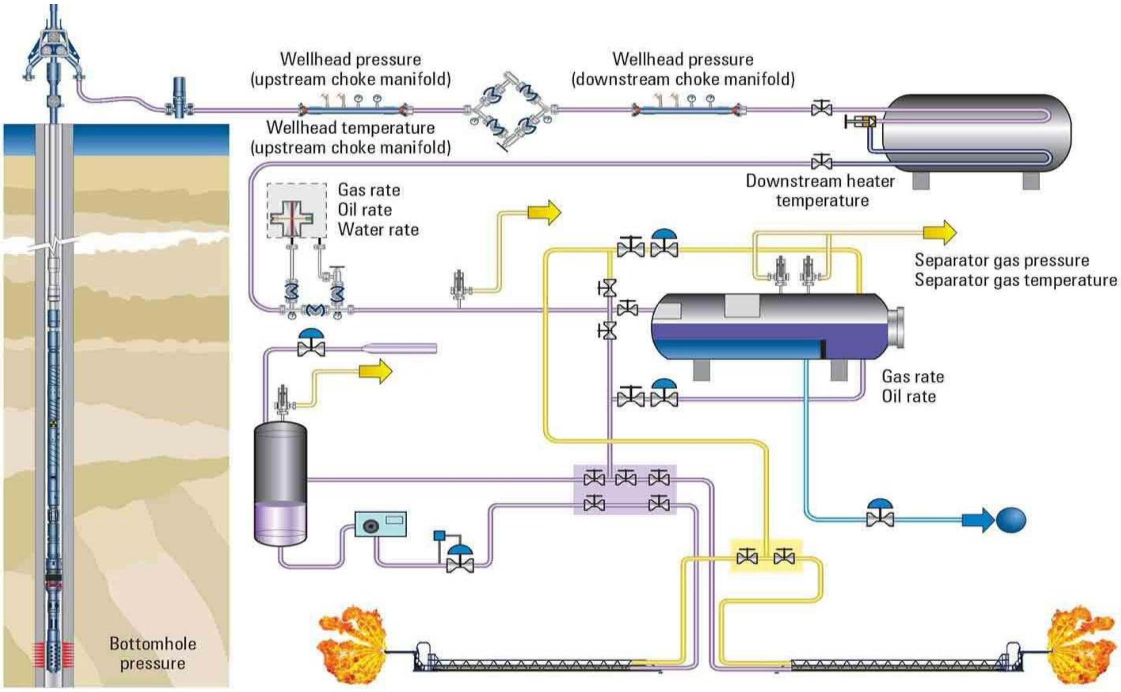
Figure(I.3): Image of fixed or positive wellhead chokes with 32/64 in or ½ in apertures [7].

b) An adjustable or variable choke:

It is installed to adjust the flow rate on the wall, and by rotating it, it varies the entry pulley. In wells which have sand production problem, gate valve is not used to decrease or increase the flow, because in addition to sever erosion and corrosion, sand causes blocking of some part of well column and as a result, it results in a wellhead flow pressure drop and also wellhead equipment erosion. This kind of choke is classified into two groups: Type 1, is similar to a needle valve and has a handle, by which turning the flow can be decreased or increased. The valve handle has gradation and it shows the equipment diameter Figure (I.4). Type 2 has two disks, each with two pores, one of which is constant while the other two rotate to adjust the appropriate flow rate Figure (I.5) adapted from [8].



Figure(I.4): An illustration of the type 1 chokes and An illustration of the type 2 chokes.



Figure(I.5): The optimized offshore well test process included calculating bottom hole pressure in real time for better predicting reservoir and well behavior.

I.4 Theory of two-phase fluid flow through wellhead chokes:

I.4.1 The principles of two-phase fluid flow are as follows:

(A) When a fluid flows through a flow pipeline, the fluid pressure is initially above the bubble point, i.e., the gas remains dissolved in the liquid.

(B) When the fluid is at the bubble point, gas bubbles begin to emerge from the liquid and it becomes a two- phase fluid.

(C) When the fluid pressure drops below the bubble point pressure, the fluid moves in a pipeline as two-phase flow.

Key reasons for involving wellhead chokes in a production flow stream are:

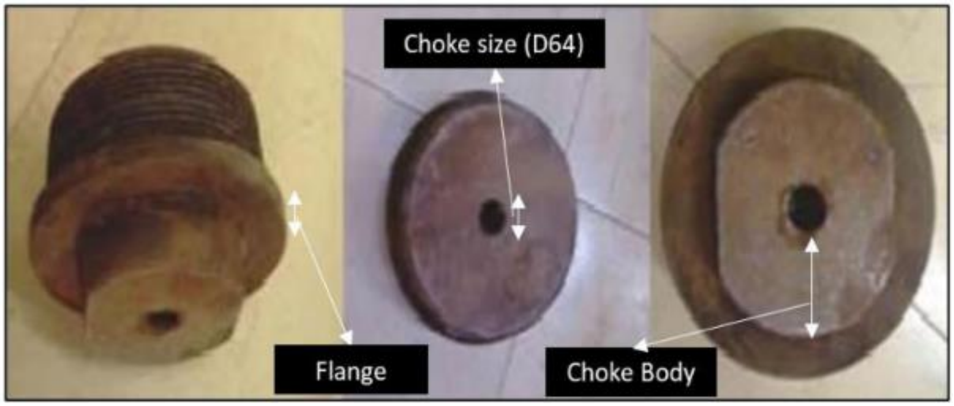
(1) to create a pressure drop in the flow stream to prevent damage to well equipment;

(2) to facilitate separation of gas from oil in separators;

(3) to create a back pressure on the reservoir to assist in maintaining reservoir pressure.

Although biphasic flow-meters were developed decades ago, their high-cost and ongoing calibration requirements to accurately calculate of two-phase fluid flow have limited their uptake. Calculating the exact

GLR value inside the well reducer itself, while a well is online, is costly and requires the installation of accurate sensors. In practice, estimated GLR data are typically used to calculate fluid flow rates through wellhead chokes [9].



Figure(I.6): Image of fixed or positive wellhead chokes with 32/64 in or ½ in apertures.

I.5 Multi-phase flow through wellhead chokes Modeling:

I.5.1 Literature review:

Multi-phase flow models have been extensively studied and developed over the years to understand the complex behavior of fluids through wellhead chokes. The early contributions include the seminal work of Gilbert in 1954 [10], followed by significant advancements by authors such as Dukler et al. (1964), Hagedorn and Brown (1965) [11], Beggs and Brill (1973) [12]. In the following years, Taitel and Duckler (1976)[13], Ishii and Zuber (1979),and Chen and Golan (1979), Orkiszewski (1983), Yen and Dukler (1986), Zhang et al. (1998) [14], and Abdul-Majeed and Sarica (2004) [15]presented their respective models, contributing to the evolving understanding of multi-phase flow phenomena. Notable models in subsequent years include those by Oliemans et al. (2008)[16], Other influential models were developed by and Hidrobo et al. (2012)[17], and Guo,B (2013), Leal model (2013) [22] , Oliemans et al. Model (2017) [18], Aider A. Jumaah (2019) [23], These models have provided valuable insights into the dynamics of multi-phase flow through wellhead chokes, facilitating more accurate predictions and improved optimization strategies in the oil and gas industry.

I.5.2 Models:

a) Gilbert model:

The first investigation on gas-liquid two-phase flow through restrictions was performed by Tangren. He presented an analysis of the behavior of an expanding gas-liquid system (1954). He showed that when gas bubbles are added to an incompressible fluid, above a critical flow velocity, the medium becomes incapable of transmitting pressure change upstream against the flow. Several empirical choke flow models have been developed in the past half-century. They generally take the following form for sonic flow [24]:

$$P_{up} = \frac{aGLR^{b}Q}{D^{c}} \qquad \text{(Eq I.2)}$$

a, b and c are empirical unknown parameters. related to fluid properties.

On the basis of the production data from Ten Section Field in California, Gilbert found the values for a, b and c to be 435, 0.546 and 1.89,respectively .Other values for the constants were proposed different researchers including Baxendell, Ros, and Achong, more than 20 models were developed on the basis of this model [25].

b) Seidi and Sayahi:

The authors have considered that the passing flow rate through wellhead chokes is a function of wellhead pressure, choke diameter, before choke temperature, and water production rate. A new model for the estimation of the oil rate passing through wellhead chokes has been proposed (2015). In this study, 180 tested data for 5 wells from a heavy crude oil field were used to develop a new model for estimating oil rate passing through wellhead chokes [1]. By considering the Gilbert equation, a New Model is formed as equation (Eq.I.3):

$$Q_g = aLGR^{b}S^{c}\Delta P^{d} \qquad \text{(Eq I.3)}$$

a, b, c and d: Empirical unknown parameters.

The obtained general formula is as follows:

$$Q_g = \frac{0.015S^{1.27}\times\Delta P^{0.56}}{LGR^{0.4}} \qquad \text{(Eq I.4)}$$

The optimum solutions (the best empirical unknown parameters.) achieved for the proposed model and a comparison between measured and predicted gas flow rates of all data pointes The obtained solutions were : 0.0164, 0.3931, 1.2624, and 0.556 for a, b, c, and d, respectively and the new equation turns into the following form:

$$Q_g = \frac{0.0164S^{1.2624} \times \Delta P^{0.556}}{LGR^{0.3931}} \qquad \text{(Eq I.5)}$$

Table(I.1): Different parameter ranges of south Iranian field data.

Parameters	**S (1/64inch)**	**LGR (bbl/MMscf)**	**Qg (MMscf/D)**	**Pu (psia)**	**Pd (psia)**	**DP (psia)**	**T (F)**
Minimum	40	0.688	11.3	1131	824.84	14.5	109
Maximum	192	32.215	113	4452	3045.82	1407	211

Table(I.2): The sub-critical data gathered from South Iranian gas-condensate wells [1].

Number	**Pu(psia)**	**Pd(psia)**	**Qg (MMscf/D)**	**S(1/64inch)**	**LGR(bbl/MMscf)**
1	1501.978	824.83258	11.3008	40	12.4624466
2	1518.076	1484.13896	12.7134	40	9.56457596
3	2422	1740	13.1	40	16.1
4	2103	1726	15.1	40	16.1
5	1827	1638	15.641	40	4.81
6	1406.5	1319.5	19.811715	128	19.5209967
7	1450	1334	23.696365	128	32.214988
8	2741	1798	26.84	40	13.536
9	1827.57	1653.534	27.73	64	5.556
10	2059	2044.5	28.1637125	160	6.14448308
11	1319.5	1290.5	28.711095	144	4.4527821
12	1334	1290.5	28.852355	144	4.80900467
13	1305	1290.5	28.852355	144	4.63089339
14	2016.216	1900.45559	31.7832001	64	5.428
15	1636.156	1195.5025	32.4894934	64	27.491349
16	1493.5	1435.5	33.019525	128	14.9969701
17	3901.499	2944.301	38.72	40	4.374
18	1957.5	1885	38.881815	160	7.35599604
19	1232.5	1218	39.5528	192	2.38847232

20	2102.5	2030	44.14375	160	6.23656662
21	1247	1232.5	44.4969	192	1.2200623
22	1972	1899.5	45.238515	144	7.00867903
23	2073.5	2001	45.521035	160	6.43498259
24	1783.5	1566	45.55635	128	14.8936656
25	1160	1131	45.9095	192	2.13021096
26	1605.094	902.85872	46.5564	40	1.53710038
27	1914	1827	47.67525	144	5.58200765
28	2044.5	2001	50.041355	160	5.70811044
29	1290.5	1276	52.2662	192	1.54066261
30	1131	1102	52.9725	192	1.97632281
31	1145.5	1131	54.3851	192	2.44012459
32	1957.5	1870.5	56.398055	160	7.32749823
33	1885	1566	56.85715	128	13.2710718
34	1986.5	1899.5	56.963095	160	5.64505904
35	2451	1769	57.1043	64	8.193
36	2393	1682	57.5674	64	8.193
37	2393.314	1726.41031	58.2692001	64	8.19319639
38	2480.337	1755.41786	58.2692001	64	8.19319639
39	2407.818	1697.40276	58.2692001	64	8.19319639
40	1624	1551.5	58.5769905	176	5.55555556
41	1348.5	1290.5	60.0355	128	1.92003964
42	1870.5	1537	63.567	128	14.6211553
43	1334	1290.5	63.743575	192	2.12842985
44	1392	1348.5	63.743575	128	1.24677899
45	1174.5	1145.5	64.62645	192	1.92360187
46	1841.5	1667.5	66.74535	128	5.04233045
47	1841.5	1667.5	67.063185	128	5.44308084
48	3785.475	2987.81	77.5	64	4.374
49	1682	1595	82.531155	192	5.18481948
50	1624	1508	83.738928	176	5.58659218
51	1798	1624	83.873125	192	8.50659493
52	3350.385	2016.109	84.4	64	6.3
53	4452.613	3045.822	86.76	64	4.374
54	1783.5	1653	87.5970918	192	6.31938836
55	1740	1595	88.57002	176	5.12052131
56	1595	1508	89.91199	192	4.23904856
57	1595	1566	90.2474825	192	0.688222
58	1493.5	1406.5	95.27987	192	3.41973666
59	1566	1464.5	96.2863475	176	3.87570154
60	1653	1479	101.389365	144	3.96155118

61	1653	1479	101.654228	192	5.03894634
62	1537	1435.5	102.06035	176	3.52660343
63	1653	1493.5	103.8261	176	4.19808297
64	1653	1522.5	104.841406	192	5.68174997
65	1972	1696.5	109.702516	144	5.56669008
66	2175	1899.5	111.295223	144	5.41315815
67	1653	1435.5	113.008	160	5.71381

c) Kargapour Model:

Mohammad Ali Karagpur in his work, by using basic concepts of fluid mechanics, it is shown that the rate of two-phase flow through a choke generally depends on the pressure drop across the choke besides other factors by deriving a global choke formula (2019). Based on this finding, a general Gilbert-type choke formula is derived which includes differential pressure across the choke. This new choke formula is validated using a field data bank including 399 data points this module Semi analytical [21].In this module for two cases:

Subsonic single-phase gas flow by adopting the concept of gas mass flow from basic concepts of fluid mechanics (Streeter 1962; White 2011) and, employing the average values for some gas specifications and the concept of discharge coefficient from orifice flow metering, the following formula is derived for subsonic single-phase gas flow.

$$\mathrm{q_{scfd}} = 65554 \times \mathrm{D^2 P_{up}} \sqrt{\left(\frac{\mathrm{P_d}}{\mathrm{P_{up}}}\right)^{1.5625} \left[1 - \left(\frac{\mathrm{P_d}}{\mathrm{P_{up}}}\right)^{0.21875}\right]} \qquad \text{(Eq I.6)}$$

Choke for two-phase (gas and liquid) it is assumed that part of area of choke is occupied by gas stream and liquid flows in the rest. In mathematical form At = Ag + Al , it is written: where ‘At ’, ‘Ag’, and ‘Al’ are total cross-sectional area of choke, assumed area available for gas flow, and assumed area available for liquid flow, respectively. By utilizing Bernoulli’s equation for liquid flow and Eq(4) [22].

(One phase) for gas flow and substituting them in Eq. (two phase), the following equation is generated as a general form of choke formula for estimating the liquid flow rate in two-phase fluid flow:

$$Q_{BPD} = P_{up}D^2 \times \left\{ \frac{\sqrt{P_{up}}}{552\times\sqrt{\frac{\left(1-\frac{P_d}{P_{up}}\right)}{SpGr}}} + \frac{GOR}{65554\times\sqrt{\left(\frac{P_d}{P_{up}}\right)^{1.5625}\left[1-\left(\frac{P_d}{P_{up}}\right)^{0.21875}\right]}} \right\}^{-1} \qquad \text{(Eq I.7)}$$

d) The Leal model:

The equation expresses the gas flow rate from the wellhead choke in terms of choke diameter, gas specific gravity, flowing fluid temperature, upstream pressure and downstream pressure(2013). The Leal equation is used as the objective function for analysis is presented here as equation:

$$Q_g = \beta_1 D^{\beta_2}\left(\frac{P_{up}}{14.7}\right)\sqrt{\left(\frac{1}{\gamma_g T}\right)}\beta_3\left[\left(\frac{P_d}{P_{up}}\right)^{\beta_4} - \left(\frac{P_d}{P_{up}}\right)^{\beta_5}\right] \qquad \text{(Eq I.8)}$$

$\beta_1$to β_5 are unknown parameters.

Where, Leal presented values ideal of constants as follows [22]:

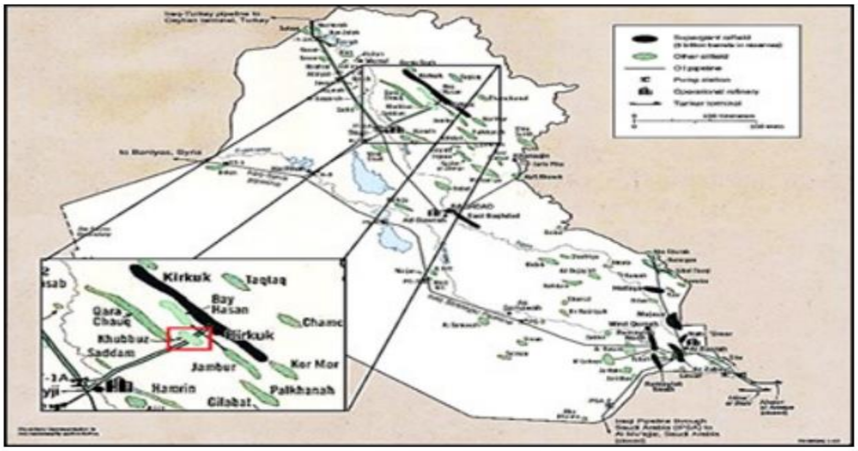
β_1 = 0.00149228 β_2 = 2.118654173 β_3 = 1.586085251 β_4 = 0.739034 β_5= 1.369516866

e) Aider A. Jumaah:

Gilbert correlation has been modified to determine the performance of multiphase fluid-flow through the wellhead and choke. The modification present two sets of new correlations based on statistical analysis of 33 production tests data from 12 wells produce from Tertiary Reservoir in Khabaz oil field, first correlation is modified of Gilbert equation to predict liquid flow rates as a function of wellhead flowing pressure, gas-liquid ratio and choke size, the second correlation takes the effect of water cut and sediment (BS & W) as an effective parameter to minimize error. A comparison between the results of each correlation with measured date has been made to select the best correlation to predict flow rate in newly drilled wells (2019). The oil flow rates predicted by the new correlations show excellent agreement with the measured rates and the second modified Gilbert correlation are found to be closest to all ranges of flow rate variables with an average error of 10 % and R2=0.9493 [23].

Area of study:

The Khabaz oil field is multi-pay zones Carbonate oil fields like most of the carbonate oil fields in the north of Iraqi. It is located in North West of Kirkuk city and far about 12 km from Kirkuk city center as shown in Figure (I.7).



Figure(I.7): Khabaz oil filed location [23].

The Khabaz oil field consists of three main hydrocarbon reservoirs.

1. Tertiary Reservoir.
2. Cretaceous Upper Qamchuqa (Mauddud) Reservoir.
3. Cretaceous Lower Qamchuqa (Shuaiba) Reservoir.

Table(I.3): Production Test Data [21].

No	DATE	Chock size in	Pressure psi	Q bbl/d	GOR scf/bbl	WT %	flow path
1	15/07/2000	0.438	446.5	1000	1112	1	Tubing
2	05/05/2002	0.375	445	850	873	0.5	Tubing
3	22/07/2000	0.375	1102	1300	913	1.3	Tubing
4	01/05/2002	0.375	1233	1150	1290	2.3	Tubing
5	03/09/2000	0.250	1711	400	2595	0	Tubing
6	16/10/2001	0.313	1377.5	450	1976	0	Tubing
7	17/08/2002	0.250	1850	400	2500	0	Tubing
10	15/10/2001	0.375	1116.5	1600	927	4.5	Tubing
11	04/05/2002	0.375	1130	1250	949	4.7	Tubing
12	18/07/2000	0.375	1203.5	1200	1174	0.2	Tubing
13	21/07/2000	0.313	1232.5	850	1221	0.3	Tubing
14	05/09/2001	0.313	1261.5	1100	1078	0.9	Tubing
15	20/10/2001	0.250	1305	750	989	0.8	Tubing
16	10/10/2001	0.563	1102	2600	1027	1.2	Tub.
17	07/07/2000	0.250	1450	500	1780	0	Tubing
18	10/05/2002	0.250	1855	450	1976	0	Tubing

19	16/07/2000	0.500	1160	2250	1121	0	Tubing
20	20/07/2000	0.438	1189	1700	1047	0	Tubing
22	21/05/2002	0.438	1090	1700	1134	0	Tubing
23	04/08/2002	0.500	1190	2200	1079	0	Tubing
24	27/10/2001	0.375	1276	1100	1483	0	Tubing
25	08/05/2002	0.375	1275	1000	1483	0	Tubing
26	29/06/2000	0.656	783	2900	921	0	Tubing
27	01/07/2000	0.563	899	2150	966	0	Tubing
28	03/07/2000	0.500	928	1550	957	0	Tubing
29	23/10/2001	0.250	1363	650	1369	0	Tubing
30	24/08/2002	0.250	1025	700	847	0	Tubing
31	23/07/2000	0.297	1232.5	950	1249	0.8	Tubing
32	17/09/2001	0.297	1322	1250	1068	0	Tubing
33	22/10/2001	0.266	1421	850	1134	0	Tubing

The first correlation for Khabaz field is Gilbert modified equation based on regression analysis of equation (Eq.I.9) as flowing:

$$Q = \mathrm{PWF}\,\frac{D^a}{c * GOR^b} \qquad \text{(Eq I.9)}$$

a, b and c : Empirical unknown parameters.

The second correlation has been developed considering a parameter which has not been covered in the previous correlations; water cut (Wct) measured volume percentage of the production stream, in addition to other parameters which had been added to the correlation, in order to reduce error in the field condition to a minimum, as represented in the following form.

$$Q = \mathrm{PWF}\,\frac{D^a}{c * GOR^b} * \left(1 - \frac{Wct}{100}\right) d \qquad \text{(Eq I.10)}$$

When they use 33 test data of production are gathered from 12 wells in Table (I.1):

- The first correlation:

a = 1.7634, b = 0.9058, c = 0.000275

- The second correlation:

a = 1.733, b = 1.159, c =0.0000486 and d= 1.3936

I.6 Conclusion:

In this chapter, the Multi-phase flow through wellhead chokes modeling has been discussed. It plays a crucial role in accurately predicting and understanding the behavior of fluid mixtures during oil and gas production. These models consider factors such as fluid properties, flow rates, choke geometry, and environmental conditions to simulate flow patterns and pressure drops. By utilizing advanced mathematical and computational techniques, these models enable optimized well production, improved safety measures, and informed decision-making for production operations. However, the need for an appropriate parameter estimation of the models' coefficients is of high necessary. Therefore, the selection of an accurate optimizer will enhance much more the modeling process. The next chapter will discuss the optimization techniques which could be exploited to extract efficiently the model unknown parameters.

CHAPTER II: OPTIMIZATION METHOD

II.1 Introduction:

Differential Evolution (DE) is a simple and effective evolutionary algorithm used to solve global optimization problems in a continuous domain [26]. It was proposed by Price and Storn in 1995 in a series of papers [27] and since then, it has attracted the interest of researchers and practitioners. The important average increase in interest in DE is noticeable from 2004. In the last few years, the number of citations stabilized at a level of over 1000 papers a year, which shows the importance of this meta-heuristic technique. In order to identify the unknown parameters of the selected choke rate flow rate model, DE algorithm will be described and its several variants will be illustrated .

II.2 Basic concept of DE algorithms:

The algorithmic framework of the basic DE consists of four phases, namely, initialization, mutation, crossover and selection, as shown in Fig.II.4 Initialization is a one-time process, while the remaining three mechanisms are repeated in the search process of DE in a D-dimensional solution space until the termination criteria are satisfied.

a) Initialization :

Initialization is the first process that occurs in DE to search for a global optimum solution located in a D-dimensional of real parameter space. The initial solutions for a given multidimensional optimization problem consist of NP real-valued parameter vectors, where NP represents the population size of DE. During the t-th iteration, each i-th individual solution of DE can be represented as a D-dimensional vector as (Eq.II.1)

$$x_i^t = (x_{i,1}\,, x_{i,1}\ \,, x_{i,1}\,, x_{i,1}\,, \ldots\ldots\ldots\ldots\ldots..\, x_{i,D}\,) \qquad \text{(Eq.II.1)}$$

where i = 1,2,...,NP. x_i^t is the position of the i-th element in iteration t. The initial population condition starts at t (iteration) = 0. The initial candidate solutions can be generated during the initialization stage on the basis of the lower and upper limit boundaries of the solution search space represented by (Eq.II.2) and (Eq.II.3), respectively, as follows

$$\mathrm{x}_{\mathrm{min}} = (\mathrm{x}_{\mathrm{min},1}, \mathrm{x}_{\mathrm{min},2}, \mathrm{x}_{\mathrm{min},3}, \mathrm{x}_{\mathrm{min},4}, \dots\dots\dots\dots\dots..\mathrm{x}_{\mathrm{min},D}) \qquad \text{(Eq.II.2)}$$

$$x_{max} = (x_{max,1}, x_{max,2}, x_{max,3}, x_{max,4}, \dots\dots\dots\dots\dots..x_{max,D}) \qquad \text{(Eq.II.3)}$$

For each i-th DE solution, the j-th dimensional component can be initialized by randomly generating a value in between the upper limit of $X_{max.j}$and lower limit of $X_{min.j}$ as shown in (Eq.II.4):

$$X_{i\cdot j}^{(0)} = X_{min\cdot j} + rand\,(X_{max\cdot j} - X_{min\cdot j}) \qquad \text{(Eq.II.4)}$$

while, $rand$ is a uniform distribution that can generate any real value between 0 and 1.

b) Mutation :

In biological terms, mutation is defined as an instant change of characteristic observed from a chromosome gene. In the context of evolutionary computation, mutation is a random perturbation process performed on selected decision variables. In DE philosophy, a mutant or donor vector denoted as Y_i^t is constructed from a mutation process on the basis of a given target vector of X_i^t. Generally, the DE mutation strategy can be represented as the format 'DE/*/n' where n refers to the number of difference vectors involved and * represents the target vector considered during the mutation process. The search mechanisms of five commonly used mutation strategies in DE are represented as follows:

Strategy1 :

$$Y_i^t = X_{r1}^t + F(X_{r2}^t - X_{r3}^t) \qquad \text{(Eq.II.5)}$$

Strategy2 :

$$Y_i^t = X_{r1}^t + F(X_{r2}^t - X_{r3}^t) + F(X_{r4}^t - X_{r5}^t) \qquad \text{(Eq.II.6)}$$

Strategy3 :

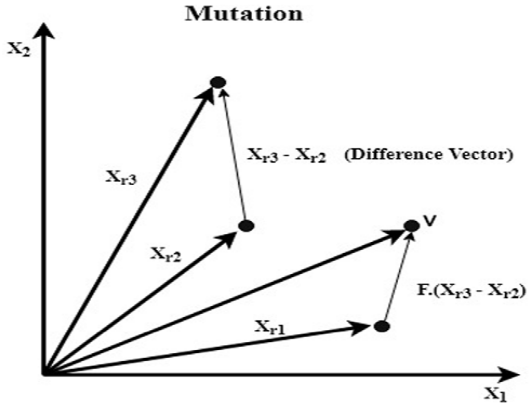
$$Y_i^t = X_{best}^t + F(X_{r1}^t - X_{r2}^t) \qquad \text{(Eq.II.7)}$$

Strategy4 :

$$Y_i^t = X_{best}^t + F(X_{r1}^t - X_{r2}^t) + F(X_{r5}^t - X_{r4}^t) \qquad \text{(Eq.II.8)}$$

Strategy5 :

$$Y_i^t = X_i^t + F(X_{best}^t - X_i^t) + F(X_{r1}^t - X_{r2}^t) \qquad \text{(Eq.II.9)}$$



Figure(II.1): Mutation scheme of DE algorithm.

while, r_1 is the population index of the DE solution selected as the base vector, r_2, r_3, r_4 and r_5 are the population indices of DE solutions randomly selected to construct the mutant vector. r_1, r_2, r_3, r_4, r_5 ∈ [1, NP] and $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq X_{best}^t$ imply that the best individual solution in the DE population is selected as the target vector. F is a scaling factor used to control the mutation process and has a value in the range between [0, 1].

Choosing the appropriate value for F is crucial to achieve proper balancing of exploration and exploitation searches of the algorithm to prevent undesirable drawbacks such as premature convergence or slow convergence speed, when the Figure (II.1) shows the mechanism of mutation vector.

c) Crossover :

In this phase, both the mutant and target vectors cross their components together in a probabilistic manner to produce a trial vector (offspring). This crossover process allows the target solution to inherit the attributes of the donor solution or mutant. Two commonly used crossover operators are known as uniform crossover and exponential crossover. The uniform crossover scheme is controlled by a crossover rate (CR) that has a value between [0,1]. The trial solution generated by uniform crossover can be defined in (10) as follows:

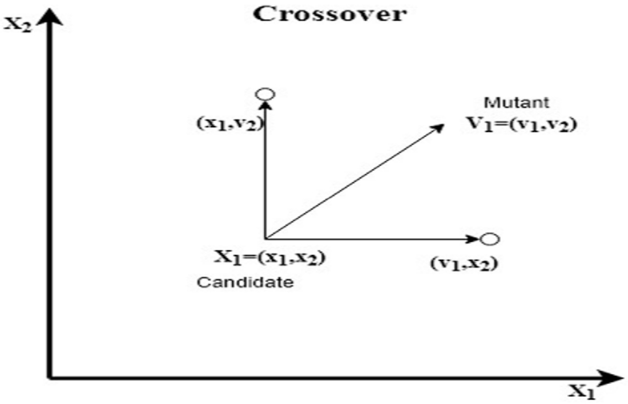
$$Z_i^t = \begin{cases} X_{i\cdot j}^t & \text{if } \; rand_{\,i\cdot j}[0{,}1] \;\leq\; CR \;\; or \; j = k \\ \quad X_{i\cdot j}^t & \qquad \text{Otherwise} \end{cases} \qquad \text{(Eq.II.10)}$$

where, $rand_{i,j}$ is a random number lies in the range [0, 1] and k $\in \{1;\; 2; \ldots;\; \mathrm{D}\}$ is randomly selected dimension index to ensure at least one dimensional component of trial solution Z_i^t is inherited from the donor vector Y_i^t.

For exponential crossover, an integer n $\in \{1;\; 2; \ldots;\; \mathrm{D}\}$ is randomly chosen as the starting point of the dimension index for a target vector to perform crossover with the mutant or donor vector. Another integer L $\in \{1;\; 2; \ldots;\; \mathrm{D}\}$denotes the number of dimensional components to be inherited from the donor or mutant vector to form the trial solution. Referring to the values of n and L, the trial solution (Z_i^t) can be obtained from (Eq.II.11) as follows:

$$Z_i^t \;\; = \begin{cases} y_{i\cdot j}^t \; if j = \langle n \rangle_D \,, \; \langle\, n+1 \rangle_D, \ldots\ldots\ldots, \langle n + L - 1\, \rangle_D \\ \qquad X_{i\cdot j}^t \qquad\qquad\qquad\qquad \text{Otherwise} \end{cases} \qquad \text{(Eq.II.11)}$$

while : $\langle n\, \rangle_D$ indicates a modulus function of D. Exponential crossover reportedly performs better on certain types of optimization problems such as those with the presence of linkages between neighboring decision variables. Figure (II.2) illustrates the mechanism of crossover:



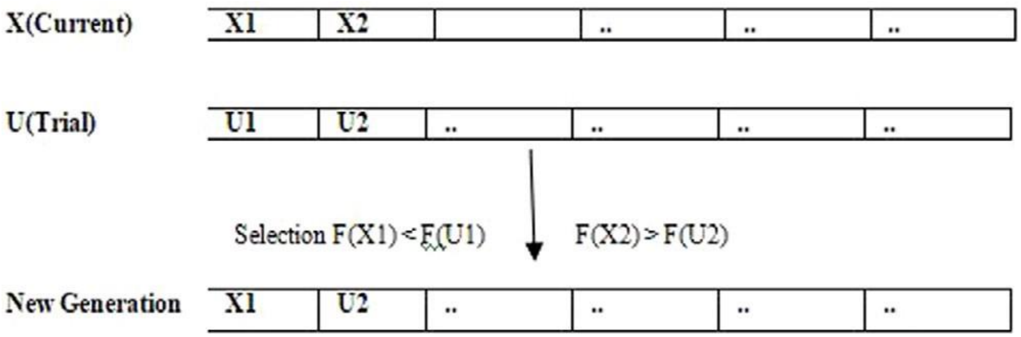
Figure(II.2): Crossover scheme of DE algorithm.

d) Selection :

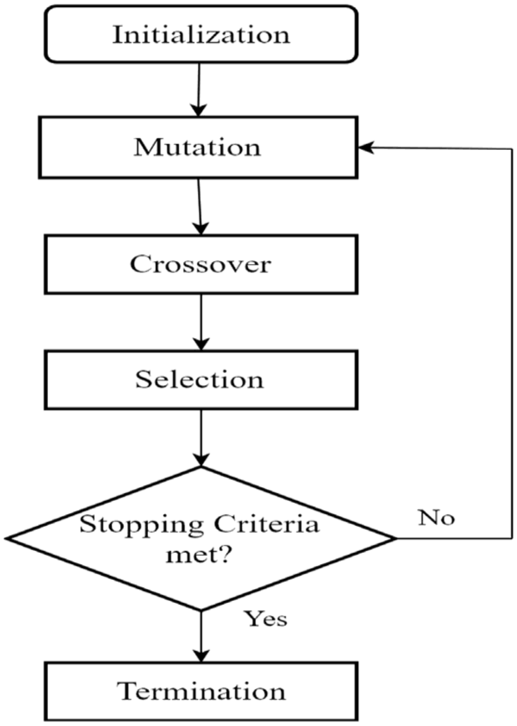
The selection process enables DE to determine the survival of a target (parent) or a trial (offspring) solution in the next iteration (X_i^{t+1}) of the search process while retaining the population size of DE in every generation. Once the new population is formed in the next generation, the iterative processes of mutation, crossover and selection are performed continuously until the termination criteria are satisfied. Two types of selection exist, namely, local and global . The selection process of DE is mathematically described as follows and in Fig.II.3:

$$X_i^{t+1} = \begin{cases} Z_{i\cdot}^{t} \text{ i}f \ \ f(Z_i^t) \leq f(X_i^t\,) \\ \ X_{i\cdot j}^{t} \qquad \text{Otherwise} \end{cases} \qquad \text{(Eq.II.12)}$$

while f(.) is an operator used to determine the objective function or fitness value of an individual solution. If the latest trial vector of Z_i^t produces a better objective function value, then the current target vector X_i^t will be replaced by $Z_{i\cdot}^t$ in the next iteration. The selection process of DE can be implemented through synchronous and asynchronous modes. The DE population can be updated simultaneously during the synchronous mode, whereas the asynchronous mode can be used to update the DE population individually.



Figure(II.3): Selection scheme of DE algorithm.



Figure(II.4): Flowchart of DE algorithm.

Pseudo-code of DE algorithm

```
Step 1: Set the control parameters: scale factor F, crossover rate Cr, the population size NP and problem dimension D

Step 2: Randomly initialize a population of NP. The population at generation G is $\boldsymbol{p_G} = \{\boldsymbol{X_{1,G}}, \boldsymbol{X_{1,G}}, \ldots, \boldsymbol{X_{NP,G}}\}$

and individual, vector $\boldsymbol{X_{i,G}} = \{\boldsymbol{X_{1,i.G}}, \boldsymbol{X_{2,i.G}}, \ldots, \boldsymbol{X_{D.i,G}}\}$;

Each individual is in range [$\boldsymbol{X_{min}}, \boldsymbol{X_{max}}$] with i=[1, 2, 3,…, Np];

Step 3: For each generation G do

For i= 1 to NP do

Step 3.1: Mutation Step

$\boldsymbol{v_{i,G}} = \boldsymbol{X^{r1}_{i,G}} + \boldsymbol{F}\,(\boldsymbol{X^{r2}_{i,G}} - \boldsymbol{X^{r3}_{i,G}})$;

Step 3.2: Crossover Step

Generate a trial vector

$\boldsymbol{U_{i,G}} = \{\boldsymbol{u_{1,i.G}}, \boldsymbol{u_{2,i.G}}, \ldots, \boldsymbol{u_{D.i,G}}\}$;
```

```
$\boldsymbol{u_{j,i.G} = v_{j,i.G}}$ **if** $\boldsymbol{(rand_{\ i.j}[0.1] \leq Cr}$ **or j=**$\boldsymbol{j_{rand})}$**;**
**Where,** $\boldsymbol{j_{rand} = rand_{\ i.j}[0, D-1]}$**;**
**Step 3.3:** Fitness evaluation Step
$\boldsymbol{f_{Ui}}$**=f(**$\boldsymbol{U_{i,G}}$ **);**
$\boldsymbol{f_{xi}}$**=f(**$\boldsymbol{X_{i,G}}$ **);**
**Step 3.4:** Selection Step
**If** $\boldsymbol{f_{Ui} \leq f_{xi}\ then\ X_{i.G+1} = U_{i.G}}$**;**
Else $\boldsymbol{X_{i.G+1} = x_{i.G}}$**;**
**End for**
**End for**
**Step 4:** If stopping criterion is not satisfied go to Step 3.
```

Performance assessment:

According to ‘‘No free lunch theorem for optimization’’, all algorithms that search for an extremum of a cost function perform exactly the same when averaged over all possible cost functions. So, for any search/optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class. Consequently, to prove the efficiency of a meta-heuristic algorithm over the other, many experiments are to be performed to reach a conclusion. Several performance metrics are available in the literature that can be used for evaluating the performance of a meta-heuristic algorithm like that of DE.

Benchmark function:

Problems: The first and foremost step to evaluate the performance of an optimization algorithm is to have an extensive set of test functions and benchmark problems based on properties like modality, scalability etc. Some standard benchmark problems include Ackley, Rosenbrock, Rastrigin, Sphere and many more. Selected benchmark functions along with their properties are provided in Appendix.

Stopping criteria:

Stopping criteria or quantifying ‘‘how and when’’ to stop the algorithm is another measure of evaluating the performance of an algorithm. Basically, stopping criteria provides an idea about the rate of convergence of an algorithm. Commonly used stopping criteria include (a) Number of runs, (b) Number of function evaluation (NFE) (c) Terminating the algorithm before it reaches maximum function evaluation if the error in the function value is less than an absolute function error value

Fitness function

It is a numerical value used to rank solutions. Since the meta-heuristic algorithms are stochastic in nature, the values are observed in the form of mean, median, best, worst, Root Mean Square Error (RMSE), Mean Square Error (MSE) and standard deviation of the fitness values.

II.3 Differential Evolution variants:

A large number of modifications to the original DE algorithm have been proposed so far. These include:

1. Modification of the differential mutation operator, e.g. by choosing the individuals based on their mutual distances [28] or fitness values [29];

2. Modification of the differential Crossover operator and mechanism;

3. Modification of the differential Selection operator mechanism;

4. Enhancing DE with new mechanisms such as population archive [30], genotypic topology [31], opposition-based initialization [32],ect..

5. Hybridization of DE

1) Modification in mutation operator and strategy:

a) Modifications in mutation scheme:

Mutation is the most important step of DE as it produces a new individual in the population. A lot many researchers have worked towards this direction and have proposed modifications in mutation strategies.

Cui et al. (2016)[33] developed a DE algorithm named as MPADE in which the population is divided into three parts based on the fitness values and three mutation strategies are applied on that for exploitation or exploration and further an adaptive technique is designed for parameter adjustment. MPADE was tested on 55 benchmark problems and 15 real life problems.

In Sun and Cai (2017) [34], a novel DE algorithm with neighborhood-dependent mutation operator is proposed called neighborhood-dependent DE (NDE) in which a pool of population topologies is used to define multiple neighborhood relationships for each individual and then the neighborhood relationships are adaptively selected for the functions being solved during the evolutionary process and applied on the benchmark functions from CEC2013. Choudhary et al. (2017) [35], proposed a mutation operator with stochastic mutation factor inspired by levy flight random walk. Yu et al. (2018) [36] proposed a mutation operator for constrained multi-objective problems. They showed that the proposed mechanism can produce well-distributed Pareto optimal front while satisfying the concerning constraints.

b) Modifications in mutation strategy:

Wang et al. [37] proposed a self-adaptive mutation DE algorithm based on PSO (DEPSO) to address the slow convergence and high tendency of premature convergence exhibited by the original DE. An improved DE/rand/1 mutation scheme was introduced based on elite archive strategy to promote the global exploration search of DEPSO. Meanwhile, the convergence speed of DEPSO was enhanced by incorporating another PSO-based mutation scheme. The performance of DEPSO was evaluated using a set of benchmark functions with different dimensional sizes (i.e. 30 and 100) and a real-world optimization problem known as arrival flight scheduling. DEPSO successfully improved its convergence speed in solving simple optimization problems without sacrificing the population diversity. The robustness of their self-adaptive mutation strategy with a relatively simple structure in dealing with more complicated optimization problems remains questionable. Xiao et al presented a multi-strategy different dimensional mutation DE (MDMDE) to address the slow convergence speed and premature convergence of the conventional algorithm. A different dimensional mutation strategy was firstly proposed to enhance population diversity, where each dimensional component of the mutant vector is contributed by the base vector and difference vector from different dimensions, as described (Eq.II 13)

$$v_{i.j}^{g+1} = x_{r3,n}^{g} + F(x_{r1,n}^{g} - x_{r2,n}^{g}) \qquad \text{(Eq.II.13)}$$

while, i$\neq r_1 \neq r_2 \neq r_3$, j$\neq$ n $\neq$ m, $v_{\iota.j}^{g+1}$ is the mutation vector, $x_{r3,n}^{g}$is the base vector and a different vector $x_{r1,n}^{g} - x_{r2,n}^{g}$ from distinct dimensions. A multi-strategy mutation scheme was also designed to enhance the convergence speed of MDMDE by dividing the overall optimization process into four generation units. The first- and third-generation units of MDMDE adopted a conventional mutation strategy of 'DE/ best/10 that promoted exploitation search, whereas the new dimensional mutation strategy was utilized in the second and fourth-generation units under the presence of dynamic mutation factor to prevent the stagnation of population in local optima regions. A new crossover rate scheme varied based on the cosine function was also introduced to further enhance the robustness of MDMDE towards premature convergence. The proposed MDMDE was applied to solve eight simple test functions, and it outperformed its peer algorithms in terms of solution accuracy and convergence speed. Nevertheless, the performance evaluation of the current study might not be sufficient to fully explore the full potential of MDMDE due to the small number of test functions and peer algorithms used. Furthermore, the idea of different dimensional mutation strategy proposed in this study might not be applicable for most real-world optimization problems because the decision variables encoded in different dimensions tend to have different search ranges.

Deng et al. [38] proposed a DE with two dynamic speciation-based mutation strategies (DSM-DE) to solve single-objective optimization problems more effectively. Dynamic speciation technique was firstly performed in DSMDE to partition the population dynamically into multiple numbers of species with each species seed considered as centers. The best vector of each species was also considered base vectors of two proposed mutation vectors, i.e. 'DE/seeds-to seeds' and 'DE/seeds-to-rand', with greater explorative and exploitative strengths, respectively. 'DE/seeds-to-seeds' considered another two species seeds randomly chosen from other species to construct the difference vector, while two random individuals were randomly selected by 'DE/seeds-to-rand' to determine the difference vector as in (Eq.II.14) and (Eq.II.15).

$$\overrightarrow{v_{\cdot\iota J.G}} = \overrightarrow{X_{\iota,J\cdot G}} + F_{i.j.g}\,(\overrightarrow{X_{\iota-seedr1,g}} - \overrightarrow{X_{\iota-seedr2,g}}) \qquad \text{(Eq.II.14)}$$

$$\overrightarrow{v_{\cdot\iota J.G}} = \overrightarrow{X_{\iota,seed.g}} + F_{i.j.g}\,(\overrightarrow{X_{\iota-r1,g}} - \overrightarrow{X_{\iota.r2,g}}) \qquad \text{(Eq.II.15)}$$

while, $\overrightarrow{X_{\iota,seed.g}}$ is the seed of the target vector; $F_{i.j.g}$is the mutation factor for each generation; seedr1, seedr2 are the indices of two randomly selected species seeds; and r1 and r2 are the indices of two randomly chosen solutions from the DE population. The performance of DSMDE was compared with that of its peer algorithms by using CEC 2014, CEC 2015 and Lennard-Jones potential problems. Table.II.1 depicts the different enhancement in the mutation factor.

Table(II.1): Summary of research on enhanced DE mutation scheme.

Author	**Technique introduced**	**Dimensional sizes**	**Results**	**Merits**	**Limitations**
[39]	An adaptive DE algorithm with novel mutation and crossover strategies	30, 50 and 100	The proposed method is able to solve various large-scale optimization problems with improved search performance.	More explorative mutation strategy was proposed to preserve population diversity. Biased parent selection strategy was incorporated into crossover operation to promote more exploitative behavior. These strategies reduced the probability of algorithm to suffer with the premature convergence in dealing with largescale optimization problem.	The performance evaluations only focused on the constant group size over generations.
[29]	DE with ranking based mutation operators	30, 50, 100 and 200	The performance of classical and advanced variants of DE algorithms was improved with the adoption of rankingbased mutation operators.	Enhanced exploitative behavior through ranking-based mutation operators. Simplicity in implementation without increasing the complexity of algorithm significantly.	The proposed ranking-based mutation operators might not effective in improving the performance of DE variants in solving certain types of problems due to the excessive level of exploitation search introduced.
[40]	Self-adaptive DE with discrete mutation control parameters (DMPSADE)	30, 50 and 100	The average optimization performance of the proposed DMPSADE algorithm was better than other DE variants in solving problems with 50 and 100- dimensional sizes.	The proposed method allowed each optimized variable to have different mutation control parameters and adaptive adjustment of mutation strategy via competition, leading to better preservation of population diversity.	The proposed algorithm did not show significant performance gains over other DE variants in solving benchmark problems with 30-dimensional size. Encoding strategy of the proposed method increased the complexity of algorithm, leading to longer execution time.

[41]	Multipopulation DE with balanced ensemble of mutation strategies (mDE-bES)	50, 100, 200, 500 and 1000	The proposed method has an excellent performance in solving persistent global problems efficiently	Multi-population strategy promoted exploitative strategy, whereas information sharing scheme among different subpopulations was adopted to reduce the drastic loss of population diversity over the generations. Different mutation strategies and control parameters were also assigned to each subpopulation to maximize the coverage of all individual solutions in search space.	The performance of proposed algorithm was not thoroughly verified with the adaptive adjustment of control parameters.
[42]	Constraint consensus mutation DE	10 and 30	The proposed algorithm was reported to have promising performance in solving constrained optimization problems in terms of fitness value and computational time. The computational time of proposed algorithm in solving 10 and 30 dimensional problems were 33.7% and 10.6% faster than those of conventional DE, respectively.	Faster computational time as compared to conventional DE in solving the constrained optimization problems.	High tendency of the proposed algorithm to suffer with rapid loss of population diversity during the initialization stage, leading to the inferior performance when dealing with multimodal constrained problems.

2) Modification in crossover operator and mechanism:

a) Modified crossover operator :

As we already know, the crossover operator constructs a new trial vector with the help of a mutant vector. Initially, exponential crossover was proposed in the original work of Storn (1996) [43] but later on it was mostly binomial variant that gained popularity among researchers (Storn and Price, 1997) [44]. An efficient comparative study of binomial and exponential crossover is given by Zaharie (2007) [45]. Zhao and Suganthan (2013)[46] showed the success of exponential crossover in the high dimensional optimization problem; Gong et al. (2014) [47] proposed a crossover rate repair technique for adaptive DE algorithms. In their scheme, the crossover rate in DE is repaired by its corresponding binary string, i.e. by using the average number of components taken from the mutant vector..Fister et al. (2016) [48] gave Epistatic arithmetic crossover based on the Cartesian graph product.. Fan and Zhang (2016) [49] proposed self-adaptive differential evolution with adaptive crossover strategies.. Guo and Yang (2015) proposed an eigenvector based crossover operator and showed that this concept can be applied to any crossover strategy. Zou and Gao (2012) [50] modified crossover rate by using a linear increasing strategy.

b) Modification in crossover mechanism:

Hui et al. [51] proposed an ensemble and arithmetic recombination-based speciation DE (EARSDE) to solve the multimodal optimization problems. In contrast to the conventional approach, the arithmetic recombination-based neighborhood speciation technique incorporated into EARSDE can enhance exploration without having to suffer from any radius parameterization issues. The proposed EARSDE was reported to outperform 11 peer algorithms in terms of efficiency and robustness when evaluated using 29 multimodal benchmark functions. The proposed speciation technique adopted by EARSDE was proven to be more generalizable in practical situations for being able to identify the peaks and troughs of highly irregular fitness landscape regions.

Fan et al. [52] presented a crossover adaptation strategy in self-adaptive differential evolution (CSA-SADE) to improve the performance of DE. Each CSA-SADE individual was envisioned to have a unique crossover strategy, mutation strategy and control parameters that can be changed adaptively by referring to its latest search progress. The proposed CASSADE was compared with eight advanced EAs, and it was proven competitive in solving the CEC 2005 benchmark functions and kinetic parameter estimation problem of mercury oxidation due to its enhanced exploitation capability.

Deng et al. [53] proposed a new DE variant that consists of rotating crossover operator (RCO) with multi-angle searching strategy, aiming to reduce the likelihood of generating inferior offspring solutions by expending the search space tactically. Unlike conventional binomial crossover scheme, the trial vector of RCO can be generated diversely within the circle regions around the donor and target vectors by referring to the self-adaptive crossover parameter and rotation control vectors that followed the Levy distribution. A comparison analysis was conducted between JADE-RCO with other five enhanced DE variants on a group of test functions in CEC 2013. Simulation results showed that DE-RCO outperformed the compared DE variants in terms of search accuracy and convergence rate, with performance gains in the range of 57% to 96%. DERCO implies the feasibility of developing different variants of multi-angle search strategy with an efficient parameter selection scheme to enhance the search performance of other DE variants. Table (II.2) illustrates the different enhancement in crossover factor.

Table(II.2): Summary of research on enhanced DE crossover scheme.

Author	**Technique introduced**	**Dimensional sizes**	**Results**	**Merits**	**Limitations**
[54]	Enhancing DE Utilising Eigenvector-Based Crossover Operator	30 and 50	Significant performance improvement was observed from proposed algorithm in dealing with the unimodal functions.	The proposed crossover operations allowed the offspring to be properly distributed corresponding to the fitness landscape, and to be directed towards the global optimum without affecting the search capabilities.	Lacking of clear explanations between the effect of dimensionality with population size of proposed algorithm.
[55]	Hybrid linkage crossover for DE (HLX-DE)	10, 30, 50, 100 and 200	High performance of HLX for the DE algorithms in terms of convergence speed as compared to four DE variants, original DE algorithm and advanced DE variants.	A group-wise binomial crossover and a group-wise orthogonal crossover were designed to guide the crossover process of DE more effectively, enabling the better balancing of exploration and exploitation strengths and the enhanced convergence rate of proposed algorithm.	Slightly performance degradations were observed when the hybrid linkage crossover mechanism was used by different DE variants to solve the hybrid composite functions.
[51]	Ensemble and arithmetic recombination based speciation DE (EARSDE)	1, 2, 3 and 10 .	EARSDE outperformed the compared optimisation algorithms, in term of efficiency and robustness, in solving multimodal functions.	Speciation was performed with the arithmetic recombination and ensemble strategy to improve the exploitative and explorative search behaviors of algorithm, respectively.	The performance of proposed algorithm to solve real-world optimization problems are unknown.
[56]	DE based superior-inferior (SI) and superior-superior (SS) crossover strategy	30, 50 and 100	The adoption of self-adaptive SI mechanism in DE variants can improve their optimisation performances in solving the unimodal, basic and expanded multimodal functions at 30-dimensional size.	Enhanced exploration and exploitation strengths of proposed algorithm by the SI crossover and SS crossover operators, respectively.	Performance degradation of the proposed algorithm can be observed when solving the hybrid composition functions. The ability of SI method to enhance performance of DE variants in solving large-scale and complex problems are questionable.

[52]	Crossover strategies adaptation with self-adaptive DE (CSA-SADE)	30 and 50	The proposed algorithm was reported to outperform five well-established DE variants and three non-DE algorithms, in terms of search accuracy.	The proposed self-adaptive mechanism can improve population diversity by allowing each individual to have the unique combination of crossover strategy, mutation strategy and control parameters.	The scalability of proposed method was not thoroughly analysed with different set of test functions at higher dimensions.

3) Modification in selection operator parameter and mechanism :

a) Parameters Modification:

DE has a unique selection mechanism that separates it out from the contemporary algorithms. Though modifications suggested in the selection scheme are limited to only a few papers, researchers have shown that suitable changes in it can further help in improving the performance of the algorithm. Yi et al. (2016) [57] proposed a fitness function value based p best selection mechanism. If the offspring is having better fitness value it means the p-best of that particular offspring is suitable for exploitation. By doing so, the population is not gathered near p-best, which results in the diversification of the population. Pan et al. (Gämperle et al., 2002) [58] proposed each target vector to be associated with a different strategy list (SL), a scaling factor F list (FL) and a crossover rate R list (CRL). When a trial vector is generated F and CR are selected from strategy lists and if an obtained trial vector is better than target vector then F and CR will enter into the winning strategy list (wSL), a winning F (wFL) and winning crossover (wCRL) respectively. After certain number of iterations, F and CR values are updated by selecting elements from wFL, wCRL and wSL.

b) Modification in mechanism :

Guo et al. [59] observed that the conventional one-to-one selection scheme tends to deteriorate the convergence speed of DE by unfairly rejecting the trial vectors with better fitness than most other current population members, especially if the corresponding target vector was even better. A novel subsetto-subset (STS) selection operator was proposed to enhance the convergence speed of DE by randomly partitioning the target and trial populations into several subsets of populations. For each subset, the best individual solutions among the subset of target and trial populations are identified by referring to their fitness values. Under the presence of the STS selection operator, the trial vectors with better fitness values were expected to have higher chances to survive in the next generation. Extensive simulation studies were conducted to compare the STS selection operator with four other survival selection schemes, and the proposed approach emerged as a more reliable selection scheme. Furthermore, the proposed STS selection improved the search accuracy and convergence speed of all DE variants significantly when it was incorporated into these algorithms.

Rakshit [60] proposed a DE integrated with noise handling policies (NDE) to enhance its optimization robustness in dealing with solution search spaces consisting of stochastic noise. A stochastic learning automata (SLA) was firstly incorporated into NDE to identify an appropriate sample size of solutions in largely noise-affected areas to achieve accurate fitness estimation without incurring additional computation complexity. A new fitness estimation strategy was also proposed by considering the weighted average of all fitness samples to reduce the influences of noisy minority fitness samples. An adaptive mutation rate was designed to select the solutions from relatively less noisy regions for the mutation process. Finally, a niching strategy was incorporated to address the deceptive effect of noise signal in the fitness landscape during the selection phase of NDE, hence ensuring proper trade-off between population diversity and quality. Two sets of benchmark functions, i.e. CEC 2013 functions contaminated with noise signals and CEC 2010 noisy benchmark functions, were used for performance evaluation, and NDE was reported to have better robustness and convergence speed against its competitors. Despite its promising performance, more effective strategies of state quantization and selection of reward functions were needed for NDE to provide more accurate sampling from noisy regions for fitness estimation. On the basis of the aforementioned reviews, Table.II.4 summarizes the research related to enhanced selection schemes. Table(II.3) gives the different enhancement in selection operator.

Table(II.3): Summary of research on enhanced DE selection scheme.

Author	Technique introduced	Dimensional sizes	Results	Merits	Limitations
[61]	Landscapedbased adaptive operator selection DE (LSAOSDE)	10, 30 and 50	LSAOS-DE outperformed others DE variants in solving majority benchmark functions from CEC 2014 and CEC 2015.	Fast convergence speed and good search accuracy	Optimal parameter settings of proposed algorithm were determined manually.

[62]	Improved individualbased parameter setting selection strategy (IDEI)	30	Competitive search performance in majority of complex optimisation problems.	Diversity-based selection strategy was designed as a secondary guidance of searching process by enabling the individuals with temporary inferior fitness values to be selected for survival in the next iteration.	The proposed diversitybased selection strategy is computationally expensive and has high tendency to promote excessive explorative search behavior that can lead to significant reduction of the convergence rate.
[59]	STS selection operator	30	STS emerged as a more reliable selection scheme compared to other 4 competitive selection schemes.	The proposed subset-tosubset selection is proven effective to improve the convergence rate of DE algorithm.	The scalability of the proposed method was not thoroughly investigated with different dimensional sizes.
[63]	Adopted multiobjective DE (MODE)	Not stated	The proposed algorithm generated good performance on the DEED problem	Provides better power emission value as compared to the other optimisation algorithm.	The computational times incurred by proposed method might be infeasible for
[60]	Improved DE for noisy optimisation	10, 20, 30, 40 and 50	NDE outperformed all its contesters in term of search capability in different dimensions and noise cancelation.	The proposed method can improve the convergence rate and search accuracy of DE.	Less efficient in solving the optimisation problems with complex fitness landscapes.

4) Initialization Modification techniques:

In population-based search algorithms, population is initialized through computer generated random numbers which usually follow a uniform distribution. Though, this is a simple method for initializing the population, researchers observed that customizing the initialization process may help in improving the performance of the algorithm. Consequently, a variety of initialization methods have been proposed in literature. Mostly these modifications are based on either contracting the search space in the beginning itself to encourage faster convergence or are based on dividing the population into smaller subgroups of populations that can perform in parallel or tries to adaptively tune the population. Some interesting initialization methods are discussed in the following paragraph.

Sun (2017) [64] proposed a novel symbiosis co-evolutionary model based on the population topology of DE, namely SCoPTDE, in which the population is divided into small species using specific topologies. Wang et al. (2016)[65] introduced Cooperative Differential Evolution (CDE) for multi-objective function using multi-population strategy. Di Carlo et al. (2015) [66] also used the multi-population technique. Aalto and Lampinen (2015) [67] proposed an adaptive mechanism population-based DE called cumu-DE in which a probability mass function mechanism is used for automatically adapting the population size. Awad et al. (2017b) [68] used niching based population reduction in a DE variant named sinusoidal differential evolution. Table(II.4) shows the different enhancement in the initialization procedure.

Table(II.4): the different enhancement in the initialization procedure.

Author	Technique introduced	Dimensional sizes	Results	Merits	Limitations
[69]	Opposition-based initialization method	2,3,4,5,6,10, 20,30. And 100	The convergence rate of the DE algorithm with opposition-based initialization was enhanced by 10% as compared with the DE algorithm with random population initialization	The proposed initialization method improved diversity level of initial population, hence increasing convergence rate of algorithm.	Significant performance degradation of acceleration rate can be observed in higher dimensional problems, i.e., for D > 10
[70]	Chaotically Initialized DE (CIDE)	2, 5 and 10	The presence of complex and yet dynamic initialization methods can improve the quality of solutions in solving optimization problems.	Reduced probability of premature convergence due to improvement of global search capability.	Only two out of seven chaotic maps can increase the solution quality in solving benchmark functions. The capability of other chaotic maps in generating solution with better quality to solve real-world applications were not investigated.
[71]	Smart Sampling DE (SSDE)	10, 20, 30, 40 and 60	Success rate and success performance of SSDE were proven to outperform original DE by 16% and 76%, respectively.	SSDE has better efficacy to find initial populations of superior quality when compared to three other DE variants with oppositional learning.	High computational cost was incurred due to the utilization of machine learning techniques to perform the smart sampling approach.
[72]	Adaptive population tuning scheme (APTS)	30, 100	The proposed JADE-APTS achieved a good performance in 30 dimensional problems and best performance in 100 dimensional problems.	Simplicity in implementation. Population size can be adaptively adjusted to increase the probability of locating global optimum.	Effect of control parameters were not carefully studied. High tendency of status monitor to discard the candidate solutions that have temporary inferior performance but can be potentially useful in long terms.
[73]	Cluster-Based	10, 30, 50,	The proposed algorithm is efficient and able to improve	The presence of intra-cluster and inter-cluster mutation strategies can	Additional computational cost can be incurred by clustering process. In

	population initialization (CBPI)	100 and 1000	the performance of DE framework consistently.	improve explorative behavior of algorithm.	addition, the results were not explicitly exploited.

5) Hybridization of DE :

Algorithm Hybridization is another popular approach used to enhance the search performance of DE by leveraging the strengths of search operators obtained from other computational intelligence algorithms. In this section, we focus on the growing trends in the past six years (i.e. 2016–2021) in research that hybridized DE with other computational intelligence algorithms. Some popular computational intelligence algorithms considered to hybridize with DE are artificial neural network (ANN), PSO, fuzzy logic (FL), WOA, FA, ACO and GA

a) DE with ANN :

Jiang et al. [74] proposed a simpler way to train the feedforward ANN by using the collective intelligence-based DE to optimize its network structure and parameters. With its ability to generate more diverse solution vectors by considering multiple best individuals from the current population via linear combination model, CIDE has better performance in training ANN. Majhi et al. [75] presented an evapotranspiration prediction model by hybridizing the DE with a radial basis function ANN, and it was then applied to predict the climate changes of a moist humid area in east– central India. Saporetti et al. [76] developed a hybrid surrogate model of DE and ANN to classify the petro-physical data automatically to improve the procedures of reservoir characterization in the oil industry. The optimal architecture and parameter settings of ANN (e.g. types of regularization, activation function and optimizer) were determined by DE to produce a robust classifier.

b) DE with WOA :

Xiong et al. [77] proposed a hybrid WOA and DE to solve the parameter estimation problem for solar model application. The performance of the proposed hybrid algorithm in this modeling problem was compared with that of the original WOA and DE algorithm under different environmental conditions such as weather, temperatures and irradiances. Dhabal et al. [78] presented a hybrid WOA and DE for image enhancement by improving the pixel intensity. It is realized using a cost function with global and local information.

c) DE with FA :

Anuradha et al. [79] hybridized FA with DE in developing a computationally efficient clustering technique for multi-agent systems. Rosic´ et al. [80] proposed an adaptive hybrid FA and DE to solve passive target localization with proper balancing local exploitation and global exploration searches during optimization problems

d) DE with ACO :

Zhang et al. [81] presented a hybrid algorithm of DE and ACO to learn the optimal structure of Bayesian network to enhance its convergence speed and learning accuracy. Xie et al. [82] proposed a hybrid algorithm of ACO and DE to address popular issues encountered in the cloud computing resource scheduling problem such as long processing time and uneven distribution of computing resources.

e) DE with GA :

Trivedi et al. [83] hybridized GA with DE as hGADE to solve the unit commitment scheduling problem. A heuristic was incorporated into the population initialization scheme to further enhance the performance of hGADE. Thakshaayene et al. [84] presented another hybrid algorithm of GA and DE to solve unit commitment problems, and the obtained solutions were compared with those of the conventional dynamic programming method. Li et al. [85] proposed a multi-objective optimization algorithm by hybridizing GA and DE to solve cloud computing applications.

Table.II. 5 presents a list of hybridized methods derived from the DE algorithm, and this trend analysis shows that the hybridization of DE with ANN, PSO and FL remains popular in the research community. Other computational intelligence algorithms such as WOA, FA, ACO and GA were observed as less favorable candidates to be hybridized with DE. Table II.5 summarizes the distributions of proposed hybrid DE variants according to year and the computational intelligence algorithms selected for hybridization. Accordingly, ANN, PSO, FL and FA are the most famous computational intelligence algorithms selected to be hybridized with DE, as indicated by recent works published in 2020 and 2021. In contrast, ACO and GA are less popular choices of candidates used for hybridization because no related works were published in 2020 or 2021. Table(II.5) indicates the different Hybridization forms of DE.

Table(II.5): Hybridization of DE algorithm with other AI algorithms.

Method hybridized	Years	Authors
DE with ANN	2018 2018 2019 2020	[86] [87] [88] [89]
DE with WOA	2018 2020	[77] [78]
DE with FA	2017 2018 2020 2021	[90] [91] [79] [80]
DE with ACO	2017 2018 2019	[92] [81] [82]
DE with GA	2016 2017 2018	[83] [84] [85]

II.4 Applications of DE algorithm :

The applications of the DE algorithm to solve different real-world engineering problems. Fifty-five articles related to the applications of original DE and its enhanced variants are covered [93]. Some of the published works will be summarized in another section for further performance analyses, whereas the remaining applications that do not fall in the scope of the study will be listed in this section only.

1) Prediction models:

Onan et al. [94] proposed a multi-objective weighted voting ensemble classifier to solve text sentiment classification problems, where DE was applied to determine the appropriate weight values of each individual classifier on the basis of their predictive performance. The machine learning models included in their proposed ensemble method were support vector machine, logistic regression, linear discriminant analysis, naı¨ve Bayes and Bayesian logistic regression. Hu et al. [95] applied DE to optimize the parameters and weights of least-square support vector machine. A multi-level regression model was then developed to predict the carbon efficiency to minimize the energy consumption incurred during the iron ore sintering process.

Peng et al. [96] presented a long short-term memory (LSTM) model optimized by DE to address the electricity price prediction task that can be formulated as time series and nonlinear regression problems. The LSTM optimized by DE can produce higher prediction accuracy. Al-Sudani et al. [97] proposed the application of DE to optimize the design of a multivariate adaptive regression spline developed using least square support vector regression. This prediction model was used to achieve more accurate forecasting of a streamflow pattern that plays crucial roles in effective planning and management of water resources. Both studies show the effectiveness of DE in optimizing the parameters of the machine learning model to develop useful prediction models in different areas of applications.

2) Industrial control :

Wang et al. [98] developed a new strategy to design a coaxial magnetic gear by referring to the theory related to magnet magneto-motive force. DE was applied to search for optimal combinations of key parameters that have a significant impact on the modulation effects of magnetic gears, i.e. the thickness of permanent magnet, ratio between magnet arc to permanent magnet pole pitch and ratio between air slot opening to pole pitch. Nadimi-Shahraki et al. [99] proposed three trial vector producers (TVPs), namely, the global best history based TVP, local random-based TVP and representative-based TVP, into their multi-trial vector-based DE (MTDE) to solve different numerical benchmark functions and four engineering design problems. Both the winner-based distribution policy and life-time archive were employed by MTDE to determine the most appropriate TVP for each subpopulation.

3) Computational systems:

By leveraging the benefits of cloud computing technologies, LaTorre et al. [100] applied DE in computational neuroscience model calibration. A simplified triggering technique with fixed diameter axon was used to increase the computation processes. The finding showed that the proposed model managed to address the complex framework of nerve damages. Houssein et al. [101] proposed to apply an adaptive guided DE (AGDE) in searching for the optimal quantum cloning circuit parameters through the minimization of cloning difference error value to improve the cloning fidelity. A new mutation operator was proposed to improve the convergence speed of AGDE by fully utilizing information brought by population members with good, average and poor fitness. A self-adaptive scheme was also introduced to adjust the crossover rate of AGDE to ensure the balancing of exploration and exploitation searches.

4) Electrical and power systems:

Biswas et al. [102] applied the linear population size reduction technique of success history-based adaptive DE (L-SHADE) presented to solve the parameter estimation problems of solar cells constructed from single-diode and double-diode models with minimum current–voltage errors. Ozyon et al. [103] produced the optimal solutions by using DE to solve the short-term operations of an electrical power system that consists of pumped-storage power generation units.

5) Feature selection :

Zhang et al. [104] proposed a self-learning multi-objective FS with binary DE (MOFS-BDE) to address the FS problems with the aim of maximizing classification accuracy and minimizing the number of selected features simultaneously. A novel binary mutation scheme based on probability difference was incorporated into MOFS-BDE to guide the solution members in locating the promising solution regions rapidly, and the self-learning capability of elite individuals in near-optimal solution regions was enhanced with a one-bit purifying search operator. The computational complexity of the selection process in MOFS-BDE was further reduced by using an efficient non-dominating sorting strategy based on crowding distance. Rivera-Lopez et al. [105] proposed a per-mutational-based DE algorithm to tackle feature subset selection problems without a fixed subset size to be defined in advance. The per-mutational-based mutation operator was designed to create new feasible solutions, and a repair-based recombination operator was proposed to maintain the population diversity during the evolution process.

6) Image processing:

Kaur et al. [106] applied a memetic DE to optimize the parameters of an intertwining logistic map to develop an image encryption technique with higher efficiency and security. The optimized intertwining map was able to produce encrypted images by generating appropriate secret keys used for encrypting the shuffled channels of color images. Sui et al. [107] developed a parallel computation of DE (pcDE) to solve the image threshold segmentation problem with better performance and stability even in the presence of different noise signals. Two communication schemes, namely, the optimal elite strategy and mean elite strategy, were incorporated into pcDE to promote information exchange between different subpopulations by replacing the local optimal solution with the global optimal solution and the mean value of the local optimal solution in all subpopulations, respectively.

7) Data clustering :

Mustafa et al. [108] designed an adaptive memetic DE to improve the quality of data clustering by optimizing the intra-cluster distance similarity measure. A neighborhoods selection heuristic and an adaptive DE mutation operator were integrated with a memetic algorithm to ensure that population diversity was maintained throughout the optimization process and to guarantee the consistency of clustering results. Wu et al. [109] presented a clustering DE method assembled with crowding factors to promote the populations to eliminate the local optima. A novel clustering method, namely, k-means special based DE, was proposed, which enhances the diversity of the populations in the earlier stage, whereas the solution accuracy was gradually improved in the later stage.

8) Health care:

Wang et al. [110] proposed a complex harmonic regularization with DE (CHR-DE) to address the biomarker selection problems that are crucial in combating cancer and genetic diseases. DE was used to optimize the hyper-parameters of CHR, enabling the latter method to have a strong ability to select relevant biomarkers from gene expression data. Kaur et al. [111] designed an e-health data prediction method by applying a multi-objective DE to fine-tune the parameters of random forest technique for different medical applications, such as the diagnosis of lung cancer, skin cancer, blood cancer, breast cancer, diabetes, brain tumour and Ebola.

9) Path planning :

Jain et al. [112] studied recent modifications made in DE to solve robotic path planning problems subjected to various constraints. Pan et al. [113] proposed a hybrid DE called CIJADE by combining the modified CIPDE (MCIPDE) and modified JADE (MJADE) to solve the path planning problem related to unmanned combat aerial vehicles. The main population of CIJADE was partitioned into inferior and superior subpopulations on the basis of fitness values and evolved independently using MJADE and MCIPDE, respectively. An external archive was incorporated into the mutation operator of MCIPDE to enhance its exploration capability, where a new crossover operator and dynamic strategy of determining elite size was designed in MJADE to achieve a better balancing of exploration and exploitation searches.

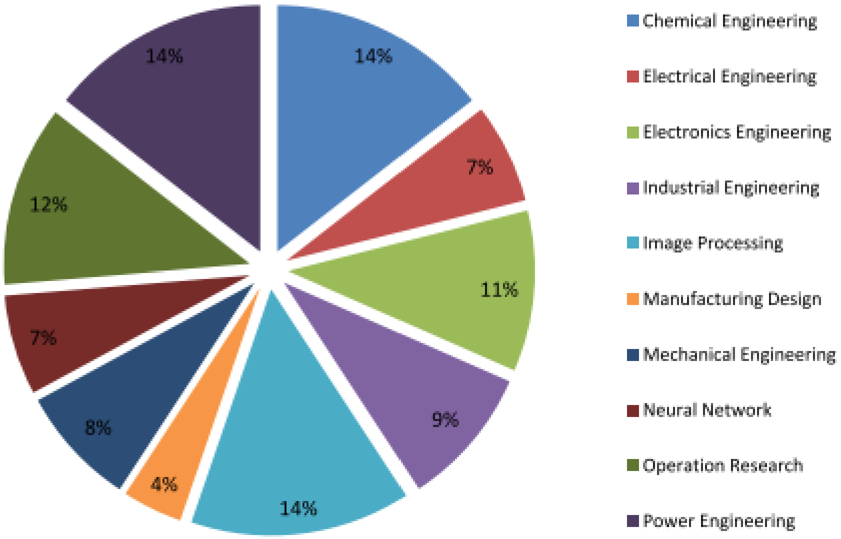
10) Differential equations resolution:

Fateh et al. [114] introduced a differential evolution-based solver to produce optimal solutions for elliptic PDEs. Second-order elliptic equations with homogeneous and nonhomogeneous forms were considered and formulated as the minimization problems. The proposed DE-based solver was effective in solving elliptic PDEs benchmark problems with linear and nonlinear characteristics. The proposed method has a high convergence speed, producing the best fitness value for each problem in approximately 500 generations, and was also reported to solve PDEs with low computational cost, as indicated by its low processing time.

In [115] a novel mesh-free approach was introduced to solve ODEs by combining the improved Fourier periodic expansion function with the weighted least square method to reduce the approximation errors. A weighted residual method was firstly applied to formulate the ODEs problems as an optimization problem. Then, an adaptive DE algorithm was used to minimize the ODEs' residuals and the boundary condition errors. The proposed method was implemented with five optimization algorithms to solve 20 types of ODEs with initial value problems and boundary value problems. The SHADE algorithm achieved the best accuracy in producing the optimal solutions for the majority of ODEs with average processing times.

12) Summary of applications:

Fig.II.5 summarizes the distributions of these studies according to their Percentage usage of DE with each application domains. Eleven applications are listed in this survey, namely, the prediction, industrial control, computational systems, electrical power systems, feature selection, image processing, clustering, health care, path planning, wireless sensor and differential equations. Feature selection is identified as the most popular application of DE, with nine research papers having been published in this research domain. Other applications also received notable attention among researchers, with works related to these research domains having been published in recent years



Figure(II.5): Percentage usage of DE in different reviewed applications areas.

II.5 Conclusion:

Differential Evolution (DE) is a powerful and widely used optimization algorithm that has proven to be effective in solving various real-world problems. DE belongs to the class of evolutionary algorithms and is particularly well-suited for continuous optimization tasks. DE operates on a population of candidate solutions, often referred to as individuals or agents, and iteratively improves them by applying a combination of mutation, crossover, and selection operations. The core idea behind DE is to maintain a diverse set of candidate solutions while exploring the search space efficiently.

CHAPTER III: PARAMETERS ESTIMATION AND RESULTS

III.1 Introduction:

The primary objective is to extract the unknown parameters of the choke gas flow rate models using a strong identification technique. For these purposes, Differential Evolution (DE) algorithm has been adopted to validate the parameter estimation.

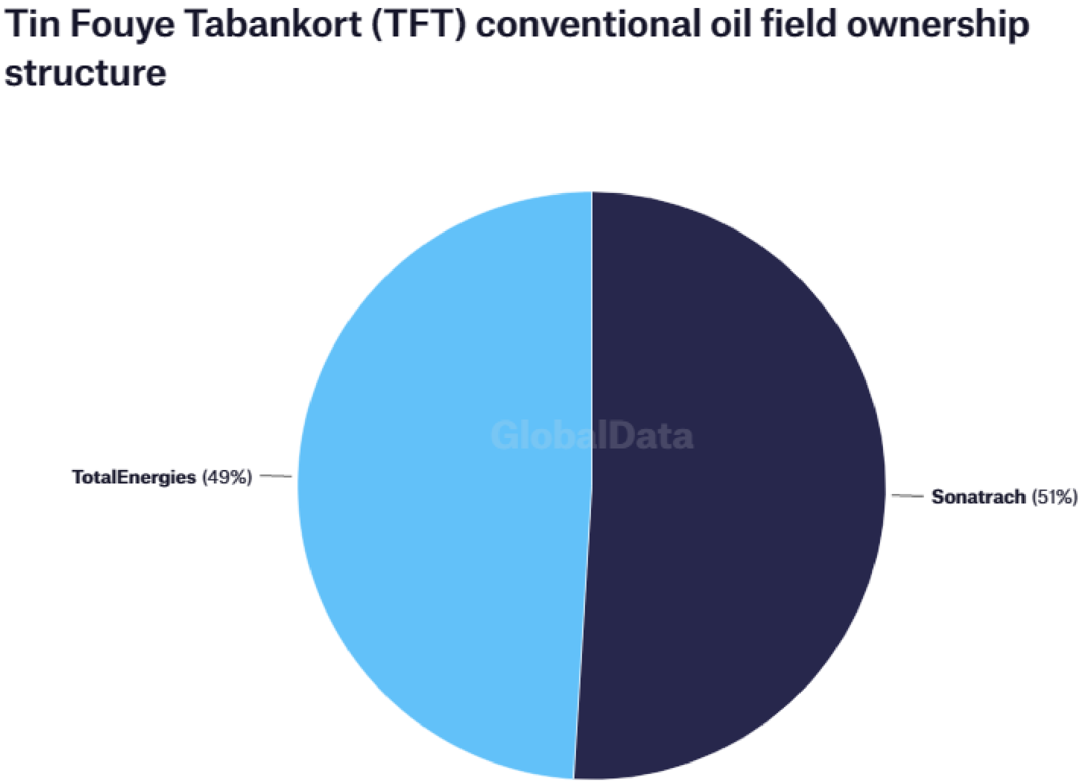
In this chapter several tests have been carried out to select the adequate DE variant for this considered problem. Therefore, a comprehensive examination of DE algorithm through rigorous testing and comparative analysis will be accomplished. To do these evaluations two models have been explored, the first one is developed by Leal et al. (2013) [22] and the second one has been proposed by S.Seidi et al. (2015) [1]. The tests have been carried out by means of available datasets from two regions GTFT field and Fars province of Iran.

To fully exploit the potential of the DE algorithm and unleash its power in real-world applications, it is crucial to determine the most effective configuration of its settings parameters. These parameters include the population size (NP), number of iterations (N_it), Crossover Probability (Cr) and Scaling Factor (F). Their values significantly impact the algorithm's convergence speed, solution quality, and ability to handle various problem characteristics.

III.2 Subject description:

1) GTFT reservoir:

Tin Fouye Tabankort (TFT) upstream field is located in Illizi, Algeria. The upstream field is owned by Sonatrach SpA (51%); TotalEnergies SE (49%). It is operated by Groupement TFT. The project started its operations in 1969.

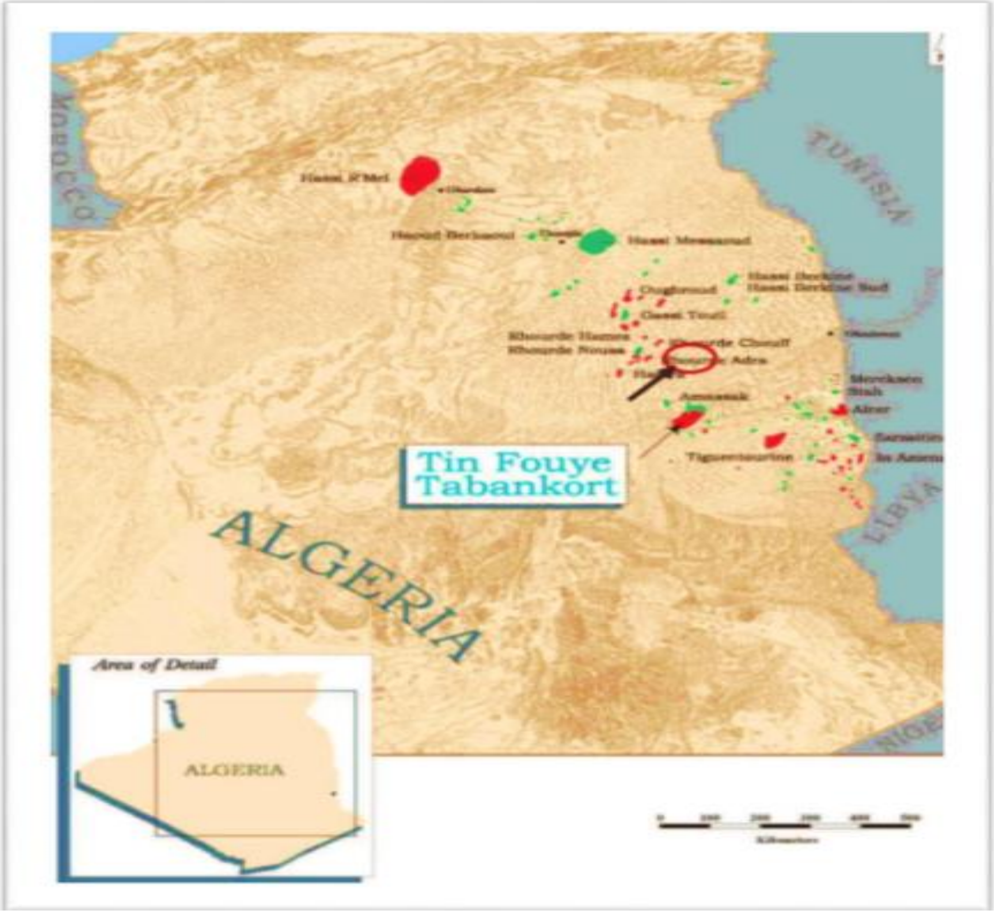


Figure(III.1): TFT conventional oil field ownership structure [116].

Tin Fouye Tabankort (TFT) Upstream Field profile includes core details such as name, resource type, asset status, stage, owner and equity stakes, operator, product specs (gravity, CO2, sulphur), location, as well as key operational data including production, start and end years, reserves and capital and operating costs. We also provide proprietary forecasts of production, capital and operating costs, and other key economic parameters, along-with relevant news, deals and contracts details [116].

2) GEOGRAPHICAL SITUATION:

The GTFT field is located in the north-west part of the Illizi basin, more precisely 300 km north-west of In-Aménas and 500 km south-east of HassiMassoud. It covers an area of 4000 km2 Figure(III.2). The TFT region is bounded by the following UTM coordinates: X1 = 310,000. X2 = 400,000. Y1 = 3,110,000. Y2 = 3,190,000 [117].



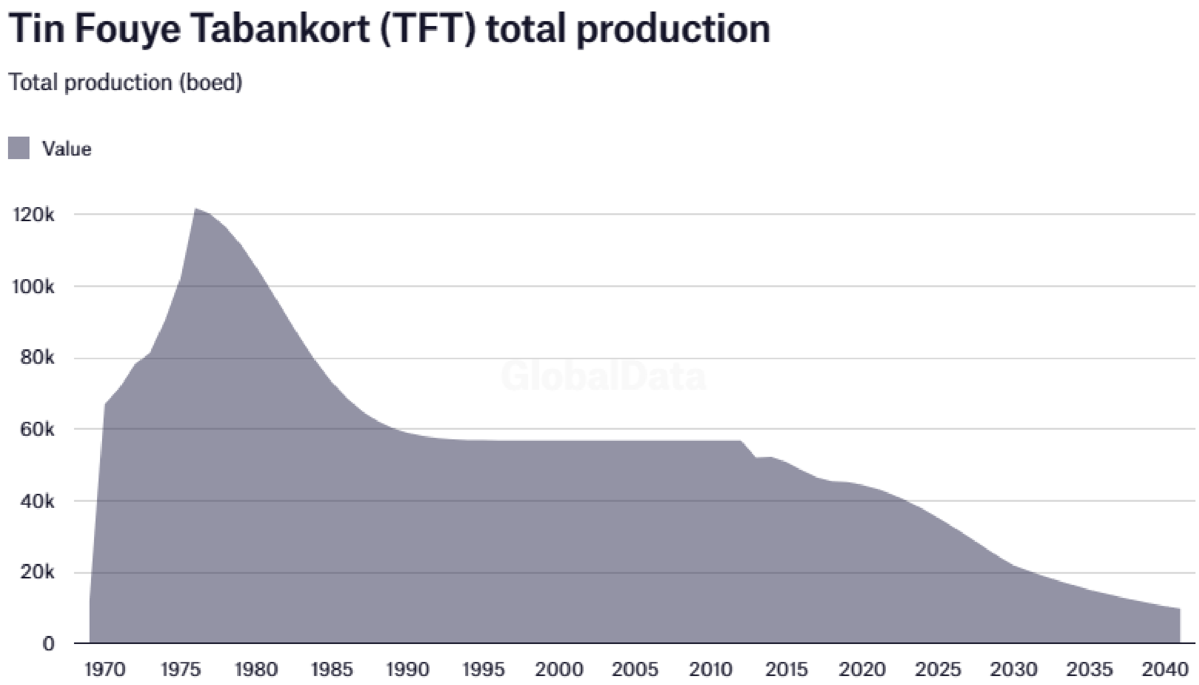
Figure(III.2): GTFT field geographical situation [117].

3) Production from Tin FouyeTabankort (TFT):

The Tin Fouye Tabankort (TFT) conventional oil field recovered 89.63% of its total recoverable reserves, with peak production in 1976. The peak production was approximately 122.05 thousand bpd of crude oil and condensate. Based on economic assumptions, production will continue until the field reaches its economic limit in 2041. The field currently accounts for approximately 2% of the country’s daily output.

4) Remaining recoverable reserves

The field is expected to recover 150.00 Mmboe, comprised of 150.00 Mmbbl of crude oil & condensate. Tin FouyeTabankort (TFT) conventional oil field reserves accounts 0.03% of total remaining reserves of producing conventional oil fields globally.[116]



Figure(III.3): TFT total production [116].

III.3 Descriptive Statistical summary of data sets:

The first database used in this study consists of 72 data points from wellhead choke tests related to the GTFT gas - condensate field in the Illizi Region, South East Algeria. The individual well test data is listed in Table (III.1). The second database used consist of 39 data points gathered from gas-condensate wells of Fars province of Iran [1]. The individual well test data is listed in Table (III.2).

Table(III.1) :Statistical analysis for GTFT field dataset.

Statistical Property	**Temperature (C°)**	**Qg (nmillm^3 /D)**	**D (1/64 in^2)**	**downstream pressure (PSI)**	**upstream pressure (PSI)**
Mean	119.4028	19.9452	46.7778	701.7917	1846.4
Minimum	80	0.2700	16	80	284
Maximum	163	52.3800	64	2595	6115

Table(III.2) :Statistical analysis for Fars province of Iran field dataset.

Statistical Property	LGR	Qg	S	PUP	PDOWN
Mean	8.0083	37.8419	54.7692	2285.8	1818.7
Minimum	5.2741	10.2166	32	1929.2	1639.1
Maximum	12.3618	54.6212	72	2480.3	2001.7

III.4 Differential Evolution parameter setting selection:

In this study the selected algorithm was standard DE which its strategy is:

Initialization: The algorithm begins by initializing a population of candidate solutions (also known as individuals) randomly within the defined search space. Each candidate solution is represented as a vector of real numbers (Eq.II.4). **Mutation:** DE uses a mutation operator to perturb the candidate solutions and create new trial solutions. For each candidate solution, three other random solutions, called the "base vector" and two "difference vectors," are selected from the population. The mutation operation is typically performed by adding the scaled difference vectors to the base vector (Eq.II.5). **Crossover:** After the mutation step, crossover is applied to combine the trial solutions with the original candidate solutions. The crossover probability determines the likelihood of a component from the trial solution being selected over the corresponding component from the original candidate solution (Eq.II.10). **Selection:** The trial solutions are compared with their respective original candidates based on a fitness evaluation function. If a trial solution has better fitness than its original candidate, it replaces the candidate in the population. This step ensures that only the better solutions survive and progress to the next generation (Eq.II.12). **Termination:** The algorithm continues to iterate through the mutation, crossover, and selection steps for a fixed number of generations or until a termination criterion is met. The termination criterion is typically based on reaching a maximum number of iterations or achieving a satisfactory solution.

The selection of the system settings of an algorithm is an important factor in determining its performance. To ensure that the algorithm setting parameter (CR, F and NP) are feasible for the considered optimization problem, a series of experiments have been conducted. By analyzing the results, the adequate setting parameters could be determined to achieve the best performances using the adopted optimization algorithm. The results from these experiments are presented in Table (III.3) for 15 tests (max_s = 15), while the number of iterations (N_it) is 700. The equation of the error calculated as follow:

$$\text{OF}=\frac{1}{N}\sqrt{\sum_{i=1}^{N}(Q_{ig.est}-Q_{ig.exp})^2} \qquad \text{(Eq.III.1)}$$

where:

Qig.est: the real gas flow rate measured in the wellhead tests.

Qig.exp: the calculated gas flow rate.

N: the number of data samples.

The objective function (OF) presented in (Eq.III.1) is based on Root Mean Square Error (RMSE).

Table(III.3): DE parameter setting selection results.

Case	Parameters		Reached iteration to get the best value	Time (sec)	OF Mean
1 [118]	F=0.5 Cr=0.9	NP=50	WORST: > 700 BEST: 63 with 0.5247 sec.	WORST: 6.1431 BEST: 5.8305 MEAN: 5.9259	0.3092
		NP=100	WORST: > 700 BEST: 73 , (1.1225 sec)	WORST: 10.7637 BEST: 10.5844 MEAN: 10.6488	0.2884
		NP=150	WORST: 105 , (2.4918 sec) BEST: 66 , (1.5660 sec) MEAN: 86 , (2.0222 sec)	WORST: 16.7101 BEST: 15.9626 MEAN: 16.4599	**0.2875**
2 [119]	F=0.7 Cr=0.5	NP=30	WORST: > 700 BEST: 227 , (1.2029 sec)	WORST: 3.8419 BEST: 3.6988 MEAN: 3.7665	0.3655

		NP=50	WORST: > 700 BEST: 247 (2.0827 sec)	WORST: 7.1138 BEST: 5.7818 MEAN: 6.0221	0.3269
		NP=100	WORST: > 700 BEST: 268 (4.2771 sec)	WORST: 11.8100 BEST: 10.9034 MEAN: 11.4819	0.2994
3 [120]	F=0.4 Cr=0.75	NP=100	WORST: > 700 BEST: 91 , (1.4045 sec)	WORST: 10.9131 BEST: 10.7884 MEAN: 10.8264	0.2913
		NP=150	WORST: > 700 BEST: 98 , (2.1888 sec)	WORST: 16.7027 MIN: 15.4325 MEAN: 15.6956	0.2890
		NP=50	WORST: > 700 BEST: 71 , (0.5761 sec)	WORST: 5.7783 BEST: 5.5661 MEAN: 5.6493	0.4457
4 [121]	F=0.9 Cr=0.8	NP=256	WORST: 237 (9.2774 sec) BEST: 148 (5.7736 sec) MEAN: 193	WORST: 27.4018 BEST: 26.2996 MEAN: 27.0273	0.2875
		NP=150	WORST: > 700 BEST: 130 , (2.8835 sec)	WORST: 15.8563 BEST: 15.3584 MEAN: 15.5419	0.3414
		NP=100	WORST: > 700 BEST: 138 , (2.0723 sec)	WORST: 10.7214 BEST: 10.3466 MEAN: 10.5078	0.3412
5 [122]		NP=50	WORST: > 700	WORST: 5.8669	0.2903

	F=0.7 Cr=0.9		BEST: 75 (0.6101 sec)	BEST: 5.6946 MEAN: 5.7504	
		NP=100	WORST: > 700 BEST: 91 , (1.3466 sec)	WORST: 11.4461 BEST: 10.3225 MEAN: 10.5218	0.3148
		NP=150	WORST: > 700 BEST: 89 , (1.9456 sec)	WORST: 16.9124 BEST: 15.1664 MEAN: 15.3264	0.3142

The best results according to the previous tests are presented in Table (III.4).

Table(III.4): The best DE parameter setting.

Parameters	**Iteration**	**Time (sec)**	**OF Mean**

F=0.5
Cr=0.9
NP=150
	WORST: 105 , (2.4918 sec)
	BEST: 66 , (1.5660 sec)
	MEAN: 86 , (2.0222 sec)
		WORST: 16.7101
		BEST: 15.9626
		MEAN: 16.4599
			0.2875

III.5 Results description and discussion:

The performances of DE algorithm have been tested and compared with Ant Colony Optimization (ACO) algorithm ones [25]. Where, an investigation on the effect of changing the maximum number of iterations and the population number of both algorithms (DE and ACO) has been carried out. While, the computer performances are : Inter(R) Core(TM) i3-101110U CPU @ 2.10 2.59 GHz and 8.00 GB of RAM, using MATLAB 2013 environment.

Four statistical metrics measuring performance have been used in terms of uncertainty error and correlation are used here to assess the performance of the DE algorithm in identifying optimal values. These statistical metrics are:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} abs(OF - OF_{best}) \qquad \text{(Eq.III.2)}$$

$$STD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(OF - \ OF_{mean})^2} \qquad \text{(Eq.III.3)}$$

$$RE = \frac{1}{OF_{best}} \sum_{i=1}^{N} (OF - OF_{best}) \qquad \text{(Eq.III.4)}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (OF - OF_{best})^2} \qquad \text{(Eq.III.5)}$$

The results of the comparison are listed in the tables (III.5-III.9) where the setting parameters of DE are similar for all the presented results: Cr = 0.9, F = 0.5 and max_s = 30.

Table (III.5) depicts the comparison performance results for both algorithms under using a maximum NFE (Number of function evaluations) of 24200 (NP = 150 & N_it = 160). From this Table, It could be noticed that the RMSE value goes to 3.6811e-06 using DE algorithm; meanwhile, it is clearly high (0.0493) using ACO. Which means that, the selected setting parameter are suitable for DE algorithm. However, the CPU run times, for both algorithms, is this case are almost similar.

Table(III.5): Performance test using a maximum NFE of 24200 (NP = 150 & N_it = 160).

Parameters	**DE**	**ACO**
RMSE	3.6811e-06	0.0493
RE	7.9051e-05	4.01092
STD	3.6023e-06	0.02750
MAE	7.5764e-07	0.040
Best OF	0.28752	0.30655
Mean OF	0.28752	0.34754
Worst OF	0.28757	0.42201
Best Run Time	4.3783	5.3004
Mean Run Time	4.7568	5.3836
Worst Run Time	5.1294	6.0676
Run Time STD	0.2433	0.1336

Table (III.6) depicts the comparison performance results for both algorithms under using a maximum NFE of 24200 (NP = 75 & N_it = 320).

Table(III.6): Performance test using a maximum NFE of 24200 (NP = 75 & N_it = 320).

Parameters	**DE**	**ACO**
RMSE	0.0164	0.0263
RE	0.4469	1.6108
STD	0.0158	0.0210
MAE	0.0042840	0.0158320
Best OF	0.28752	0.29485
Mean OF	0.29180	0.31068
Worst OF	0.3730	0.36498
Best Run Time	4.6190	5.2025
Mean Run Time	4.9187	5.3788
Worst Run Time	5.4226	5.5051
Run Time STD	0.3048	0.0719

Table (III.7) depicts the comparison performance results for both algorithms under using a maximum NFE of 24200 (NP = 40 & N_it = 600).

Table(III.7): Performance test using a maximum NFE of 24200 (NP = 40 & N_it = 600).

Parameters	**DE**	**ACO**
RMSE	0.05481	0.0098
RE	2.2169	0.8841
STD	0.0505	0.0050
MAE	0.0212475	0.0085040
Best OF	0.28752	0.28853
Mean OF	0.30877	0.29704
Worst OF	0.54954	0.31166

Best Run Time	5.0334	4.9229
Mean Run Time	5.6931	5.2465
Worst Run Time	6.1582	5.6262
Run Time STD	0.4049	0.1698

Table (III.8) depicts the comparison performance results for both algorithms under using a maximum NFE of 50000 (NP = 40 & N_it = 1250). From this Table, It could be observed that the RMSE value goes to 0.0067 using ACO algorithm; meanwhile, it is relatively higher 0.1069 by means of DE. Which means that, the selected setting parameter are suitable for ACO algorithm. However, the CPU run time goes to approximately 10 sec for both algorithms. Which is not suitable for online applications.

Table(III.8): Performance test using a maximum NFE of 50000 (NP = 40 & N_it = 1250).

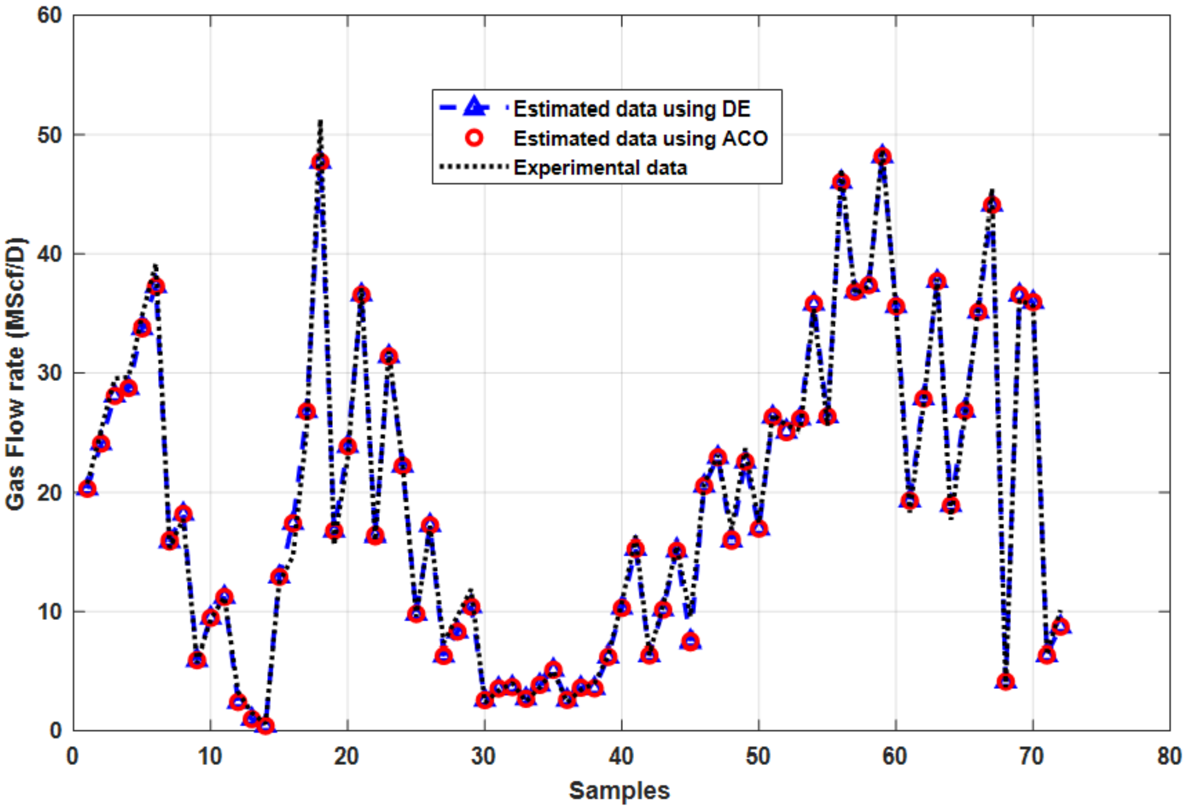
Parameters	**DE**	**ACO**
RMSE	0.1069	0.0067
RE	5.0270	0.6375
STD	0.0955	0.0028
MAE	0.0481805	0.00611349
Best OF	0.28752	0.28766
Mean OF	0.33570	0.29378
Worst OF	0.68579	0.30291
Best Run Time	9.7508	10.1757
Mean Run Time	11.4809	11.0449
Worst Run Time	13.0707	11.5779
Run Time STD	1.0506	0.37842

Table (III.9) depicts the comparison performance results for both algorithms under using a maximum NFE of 24200 (NP = 200 & N_it = 120).

Table(III.9): Performance test using a maximum NFE of 24200 (NP = 200 & N_it = 120).

Parameters	**DE**	**ACO**
RMSE	7.36119622947345e-05	0.0631
RE	0.00257	5.15402
STD	6.93512010379310e-05	0.03386
MAE	2.468059e-05	0.0533
Best OF	0.28752	0.31038
Mean OF	0.28754	0.36371
Worst OF	0.28788	0.43410
Best Run Time	4.1770	5.4145
Mean Run Time	4.6239	5.5449
Worst Run Time	5.0591	6.0203
Run Time STD	0.2686	0.1098

The results show that increasing NP allows DE to give the best performances. Conversely, when NP is decreased, ACO yields better results. Otherwise, DE can reach a satisfactory solution within a smaller number of iterations or NFE. It can explore the search space more efficiently and exploit potential solutions more effectively, leading to faster optimization.



Figure(III.4): Measured and calculated gas flow rates presentation using DE and ACO algorithms.

In order to verify and assess the performances of Differential Evolution (DE) algorithm in comparison to Ant Colony Optimization (ACO) algorithm, the Absolute Error of gas flow rates has been computed and provided in Table (III.10).

Table(III.10): Absolute Error of (DE&ACO) algorithms.

Qg GTFT	Qg (DE)	Qg(ACO)	Absolute Error Qg (DE)	Absolute Error Qg (ACO)
23.07	20.74075	21.0895991	2.329254996	1.980400894
25.19	25.14991	24.9153203	0.040090592	0.274679725
30.9	29.54607	28.6647653	1.353929314	2.235234677
28.431	29.65255	28.9749332	1.221547777	0.543933159
32.498	34.80622	33.439393	2.308223025	0.941393024
41.28	39.17901	36.8330615	2.100990049	4.446938537
17.56	15.40256	15.8772125	2.15744067	1.682787521
18.44	17.72715	17.8092098	0.712849723	0.630790192
6.24	5.76879	6.58146583	0.471210081	0.341465834
8.369	9.335869	10.0352579	0.966869229	1.666257864

9.986	11.2382	11.7558693	1.252202845	1.769869293
1.307	3.296382	3.0859957	1.989382063	1.778995697
0.52	1.531498	1.38111383	1.011497712	0.861113832
0.27	0.605388	0.61968815	0.335388274	0.349688146
11.38	12.478	13.5223473	1.098000825	2.142347283
13.12	14.69512	18.7457585	1.575124161	5.625758489
22.26	25.18652	28.9948111	2.926515851	6.734811065
43.25	51.29029	50.9184816	8.040293246	7.668481581
17.25	15.60966	17.0208597	1.640338206	0.229140308
23.76	22.92695	23.6010742	0.833045831	0.158925755
37.85	37.09721	33.0968225	0.752788602	4.753177504
18.659	15.90781	17.1952427	2.751189369	1.463757315
32.6	31.83672	31.2594987	0.763277361	1.340501346
24.21	22.41472	22.0287798	1.79528352	2.181220151
16.246	9.583223	10.3324461	6.66277718	5.913553869
22.824	17.16936	16.8798042	5.654640246	5.944195805
7.109	7.368456	7.75906958	0.259456084	0.650069575
9.192	9.632501	9.89440558	0.440500711	0.702405582
11.301	11.90355	12.0831164	0.602550968	0.782116382
2.42	2.385001	2.56036954	0.034998859	0.140369538
3.8	3.416804	3.4736558	0.383196491	0.326344202
4.89	4.047909	3.86753128	0.842090948	1.022468724
2.67	2.488708	2.66670739	0.181291514	0.003292611
3.97	3.697054	3.72097608	0.272946347	0.249023919
5.29	5.068569	4.86618	0.221430823	0.423820005
3.58	2.404957	2.59464716	1.175042894	0.985352843
4.9	3.48129	3.56100357	1.418709697	1.338996431
7.25	4.037325	3.81044064	3.212675133	3.439559359
9.375	6.364182	7.06914438	3.010817783	2.305855623
13.786	10.78014	11.2683564	3.005863691	2.517643597
21.605	16.31933	16.2781762	5.285673393	5.326823834
8.78	6.420929	7.17317152	2.359071227	1.606828476
13.23	10.49389	11.0435505	2.736106427	2.18644945
19.83	15.54935	15.7813648	4.280646834	4.048635192
14.04	9.586693	8.90586086	4.453306789	5.134139136
20.178	20.71475	20.356119	0.536748972	0.178118961
21.66	23.16068	21.6765198	1.500679669	0.016519785
19.836	16.83651	16.9016357	2.999492242	2.934364324
23.959	23.735	23.0507818	0.223999562	0.908218241
19.558	17.38525	17.6370523	2.172746614	1.920947716
25.137	26.49683	26.0375573	1.35982777	0.900557261
25.02	25.56404	25.0741512	0.544041009	0.054151242
27.278	25.16484	25.932167	2.113159005	1.345832971

38.667	35.31019	34.6375953	3.356806403	4.029404678
25.982	25.54106	26.331659	0.440940663	0.349658975
44.559	47.06739	44.4086721	2.508391169	0.150327947
34.767	36.59432	35.968662	1.827321228	1.201661987
34.18	37.24224	36.6148707	3.062243248	2.434870652
52.38	48.70316	45.7590259	3.676836652	6.620974057
32.53	35.17075	34.5263839	2.640747229	1.996383937
17.93	18.2923	19.9110353	0.36229812	1.981035345
27.44	27.27507	28.0901506	0.164929487	0.650150642
35.02	37.87923	37.2333834	2.859230876	2.213383354
15.65	17.66829	19.265898	2.018293178	3.615897952
26.83	26.12256	26.9230138	0.70744236	0.093013758
35.68	35.46913	34.8382617	0.210873103	0.841738308
46.23	45.43449	42.8677519	0.795513056	3.362248075
0.753	3.695515	4.03235064	2.942515266	3.279350639
34.73	36.10296	35.4342213	1.372959266	0.704221333
33.724	35.57344	34.9378045	1.849440286	1.213804538
8.78	6.420929	7.17317152	2.359071227	1.606828476
13.106	10.12223	9.97657415	2.983774536	3.129425846
Sum =			**134.5108496**	**144.5827043**

It is clear in Table (III.10) that the sum of absolute error (AE) by using DE is lower than the AE sum by means of ACO.

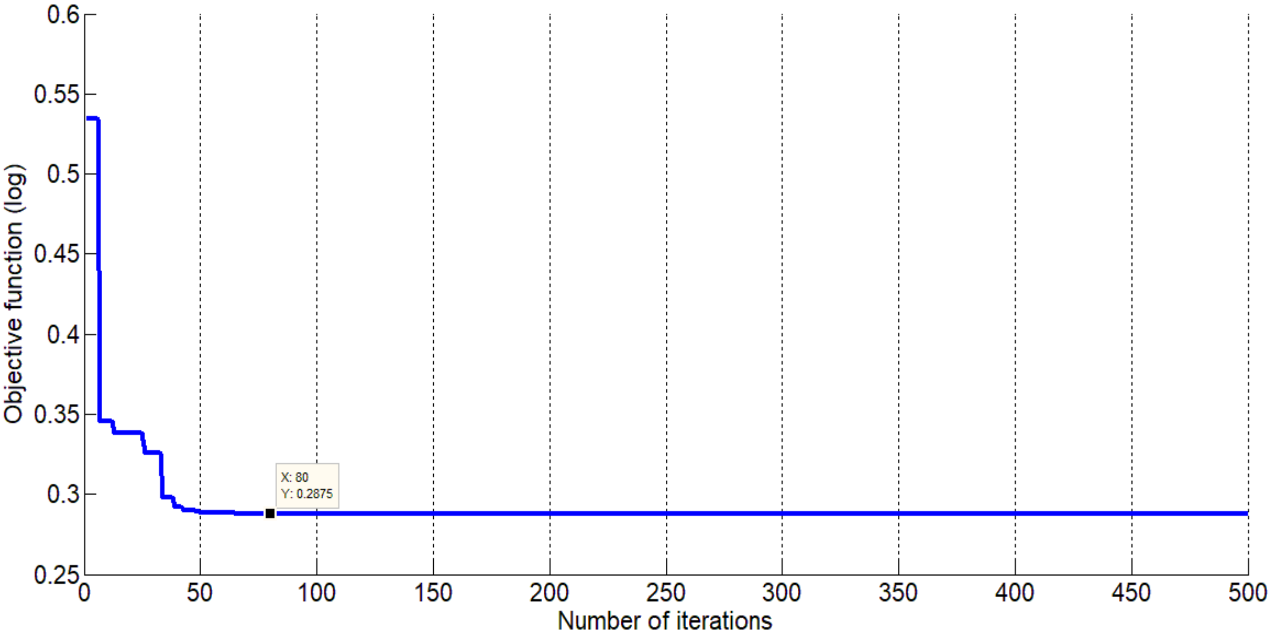
Table (III.11) compares the values of five statistical parameters for the $\beta1$ to $\beta5$ constants derived applying DE and ACO.

The limits chosen for these parameters are: **Xmin.j=1e-7, Xmax.j=5**

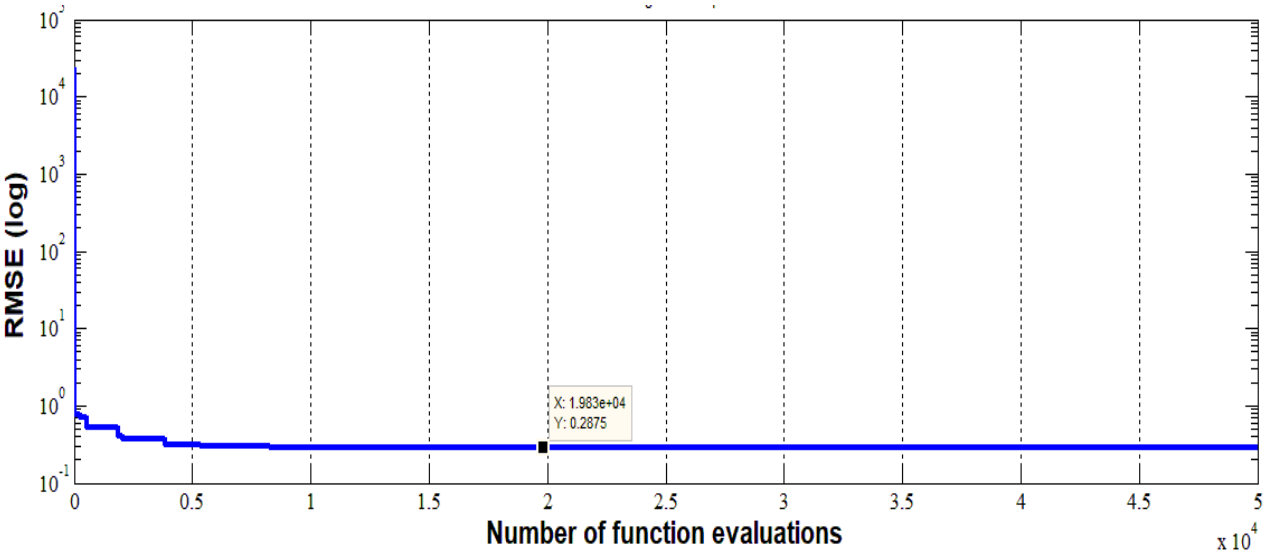
Table(III.11): Optimal Correlations by (this study &Thesis ACO) of GTFT field dataset.

Optimal coefficient value	$\beta1$	$\beta2$	$\beta3$	$\beta4$	$\beta5$	OF
ACO	0.001	2.2643	0.4358	0.4738	3	0.3042
DE	2.1926	2.2434	1.0e-07	5	0.5514	0.2875

The used model to obtain correlations in Table(III.11) is the model of Leal et al (2013) presented in (Eq I.8), where five parameters have to be identified.



Figure(III.5): OF evolution by DE of GTFT field data.



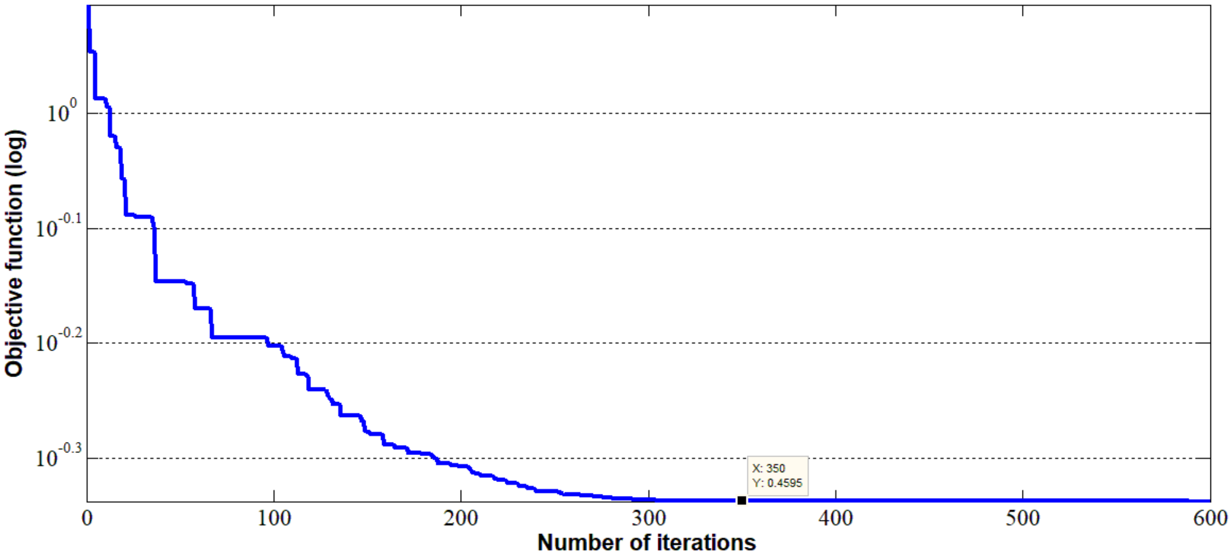
Figure(III.6): OF evolution by ACO of GTFT field data.

Table (III.12) compares the values of four constants derived applying DE, ACO and those derived from S.Seidi (2015).

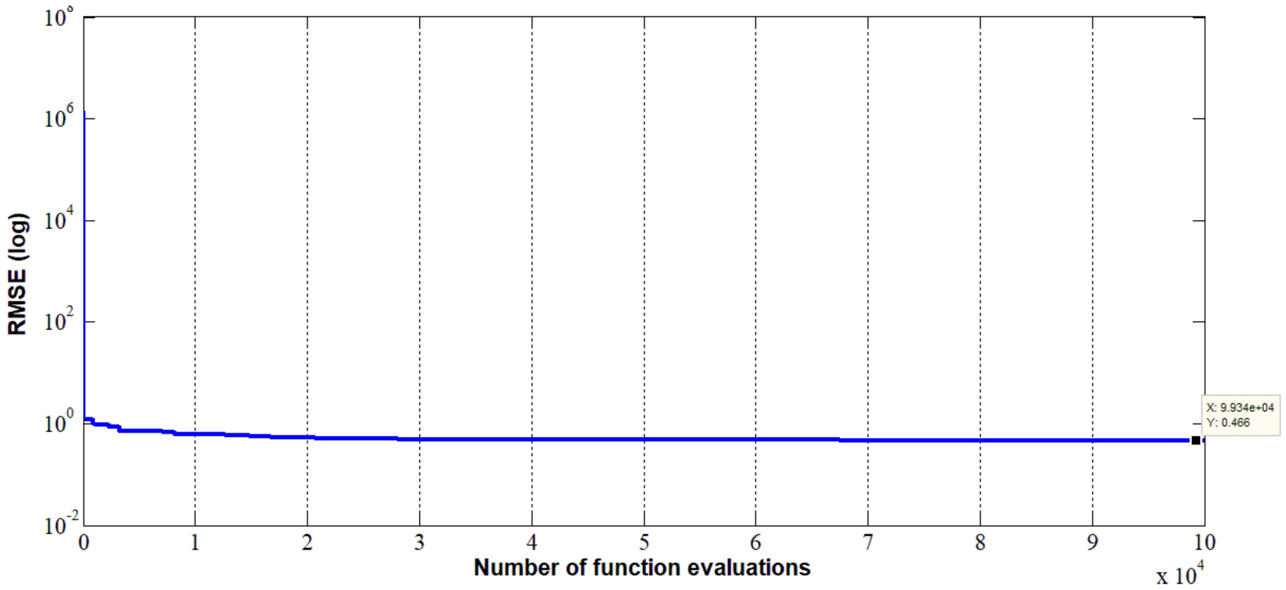
Table(III.12): Optimal Correlations by (this study & Thesis ACO & S.Seidi) of Fars province of Iran data.

Optimal coefficient value	a	b	c	d	OF
S.Seidi [1]	0.0164	0.3931	1.2624	0.556	0.9700
ACO	0.0116	0.2040	1.6040	0.3371	0.4660
DE	0.0043	0.1343	1.7582	0.3722	0.4595

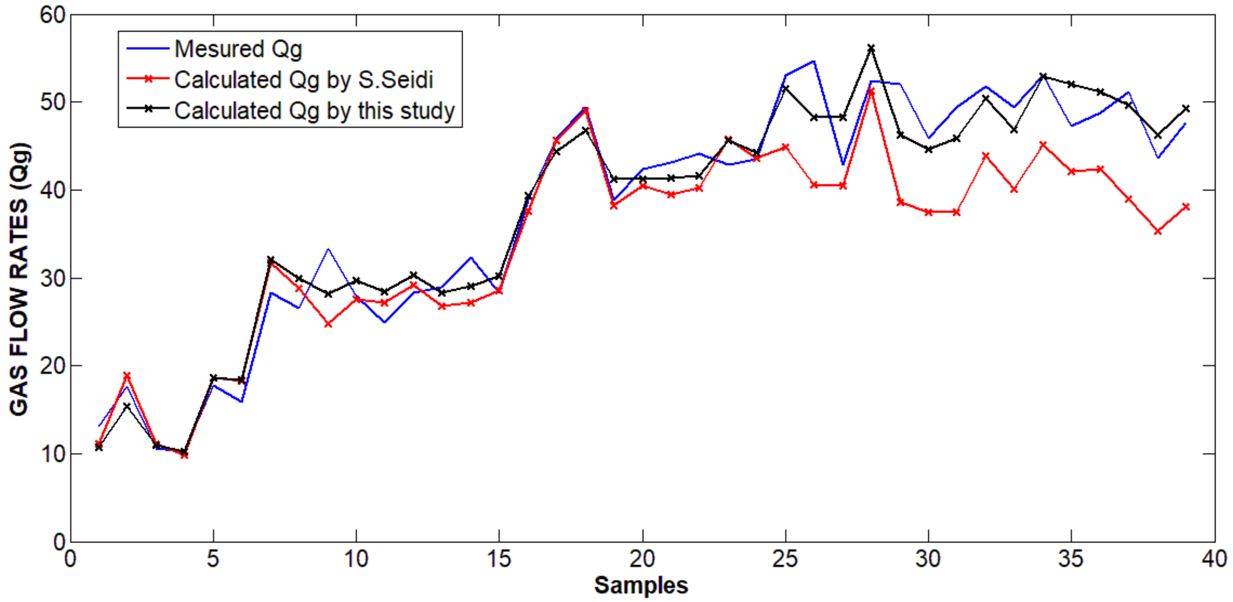
The used model to obtain correlations in Table (III.12) is the model of S.Seidi (2015) presented in (Eq I.3), while four parameters should be extracted.



Figure(III.7): OF evolution by DE of Fars province of Iran data.



Figure(III.8): OF evolution by ACO of Fars province of Iran data.



Figure(III.9): Comparison of measured and estimated flow rates of gas applying Optimal Correlations by (this study & S.Seidi) of Fars province of Iran data.

The curve illustrated in Figure(III.9) represents the prediction of gas flow rates using optimal correlations suggested by Seidi and those found in this study, the correlations using DE algorithm give better performances and appropriate fitting.

Data sets presentation:

Table(III.13): GTFT field dataset (detailed).

Number	Qg	T	PUP	PDOWN	G	D
1	23.07	589.67	2009	680	0.6754	48
2	25.19	585.67	2007	670	0.6563	52
3	30.9	591.67	2000	680	0.658	56
4	28.431	559.67	1893	700	0.6456	56
5	32.498	559.67	1878	750	0.6525	60
6	41.28	559.67	1847	710	0.6533	64
7	17.56	576.67	1490	670	0.789	48
8	18.44	576.67	1420	670	0.79	52
9	6.24	541.67	1317	410	0.634	32
10	8.369	539.67	1262	425	0.6326	40
11	9.986	539.67	1221	420	0.6336	44
12	1.307	591.67	1094	140	0.7554	32
13	0.52	583.67	1082	92	0.7619	24
14	0.27	577.67	1052	90	0.7577	16
15	11.38	565.67	1920	710	0.82	40
16	13.12	585.67	6115	2595	0.6251	24
17	22.26	601.67	5910	1989	0.628	32
18	43.25	622.67	5160	1452	0.626	48
19	17.25	599.67	2207	978	0.7108	40
20	23.76	603.67	2115	1005	0.7008	48
21	37.85	608.67	1762	1218	0.6895	64
22	18.659	569.67	2181	780	0.6357	40
23	32.6	563.67	2073	840	0.6971	56
24	24.21	583.67	1431	600	0.6575	56
25	16.246	575.67	1353	470	0.6573	40
26	22.824	578.67	1073	480	0.6539	56
27	7.109	611.67	3005	509	0.7072	28
28	9.192	617.67	2877	520	0.7087	32
29	11.301	620.67	2646	530	0.7011	36
30	2.42	565.67	311	175	0.6931	40
31	3.8	566.67	297	165	0.695	48
32	4.89	567.67	284	85	0.6892	56
33	2.67	547.67	319	182	0.6933	40
34	3.97	548.67	314	187	0.6917	48
35	5.29	551.67	306	182	0.6941	56
36	3.58	565.67	315	170	0.6931	40
37	4.9	567.67	305	160	0.695	48

38	7.25	569.67	290	80	0.6892	56
39	9.375	603.67	1666	435	0.6809	32
40	13.786	575.67	1645	450	0.6771	40
41	21.605	576.67	1636	475	0.6825	48
42	8.78	579.67	1648	445	0.6941	32
43	13.23	579.67	1611	460	0.6964	40
44	19.83	576.67	1534	510	0.7098	48
45	14.04	575.67	1693	270	0.6754	40
46	20.178	568.67	1325	553	0.676	56
47	21.66	563.67	1046	551	0.677	64
48	19.836	577.67	1646	500	0.6637	48
49	23.959	577.67	1592	547	0.6653	56
50	19.558	583.67	1670	553	0.6635	48
51	25.137	584.67	1650	688	0.6229	56
52	25.02	587.67	1680	663	0.6726	56
53	27.278	591.67	2264	1038	0.6705	48
54	38.667	594.67	2227	1078	0.6687	56
55	25.982	593.67	2306	1004	0.6584	48
56	44.559	593.67	2215	997	0.6597	64
57	34.767	584.67	2306	1050	0.6629	56
58	34.18	567.67	2330	1040	0.6674	56
59	52.38	589.67	2252	1130	0.6683	64
60	32.53	595.67	2217	1055	0.6626	56
61	17.93	569.67	2490	998	0.6627	40
62	27.44	570.67	2455	1005	0.662	48
63	35.02	571.67	2393	1013	0.6602	56
64	15.65	576.67	2401	1046	0.6778	40
65	26.83	581.67	2372	1002	0.6706	48
66	35.68	585.67	2319	950	0.6799	56
67	46.23	590.67	2216	946	0.696	64
68	0.753	569.67	2201	1998	0.7162	24
69	34.73	579.67	2258	1080	0.6716	56
70	33.724	592.67	2235	1050	0.6583	56
71	8.78	579.67	1648	445	0.6941	32
72	13.106	568.67	1651	335	0.6657	40

Table(III.14): Fars province of Iran dataset (detailed) [1].

Number	**Qg (MMscf/D)**	**PUP (psia)**	**PDOWN (psia)**	**LGR (bbl/MMscf)**	**S (1/64inch)**
1	13.15733	2103.238	1885.682	8.15525182	32
2	17.65733	2335.299	1769.652	8.15525182	32
3	10.5944	2074.231	1827.667	9.81629046	32
4	10.21658	1929.193	1726.14	10.1642977	32
5	17.75733	2175.757	1842.17	8.1591533	40
6	15.81957	2161.254	1842.17	8.12246389	40
7	28.25173	2306.291	1668.125	9.58317309	48
8	26.486	2335.299	1813.163	9.19353791	48
9	33.31467	2277.284	1784.155	12.3618401	48
10	27.95173	2219.269	1682.629	10.6397664	48
11	24.92027	2248.276	1813.163	8.19165919	48
12	28.25173	2233.772	1682.629	9.59713855	48
13	28.886	2248.276	1813.163	8.52957077	48
14	32.31467	2262.78	1769.652	9.83619523	48
15	28.25173	2233.772	1682.629	10.1493755	48
16	38.84613	2422.321	1958.201	6.45991707	56
17	45.90907	2378.81	1755.148	6.01406257	56
18	49.44053	2364.306	1639.118	6.17104993	56
19	38.84613	2451.329	1871.178	8.46016682	56
20	42.3776	2349.803	1827.667	6.32590077	56
21	43.10907	2335.299	1784.155	7.25589071	56
22	44.14333	2422.321	1871.178	6.937611	56
23	42.92445	2480.337	1755.148	7.37425227	56
24	43.43704	2451.329	1784.155	7.37425227	56
25	52.972	2335.299	1784.155	8.03917264	64
26	54.6212	2393.314	1932.704	8.03917264	64
27	42.90907	2364.306	1900.186	8.20099442	64
28	52.44053	2335.299	1648.976	7.84653346	64
29	51.972	2378.81	1972.704	7.6842555	64
30	45.90907	2277.284	1929.193	6.64752115	64
31	49.44053	2407.818	2001.712	8.30026472	64
32	51.772	2349.803	1842.17	7.62098402	64
33	49.44053	2306.291	1900.186	6.95830082	64
34	53.00627	2262.78	1856.674	6.27312956	68
35	47.20627	2306.291	1885.682	7.85792725	68
36	48.72508	2248.276	1871.178	6.64273911	68
37	51.20627	2219.269	1842.17	8.19235665	68
38	43.6748	2088.735	1885.682	5.27413503	72
39	47.6748	2074.231	1827.667	5.71836411	72

III.6 Conclusion:

In this chapter, the parameter extraction of gas flow rate model problem has been discussed. The illustrated results indicate that the performance of the optimization algorithms, DE (Differential Evolution) and ACO (Ant Colony Optimization) is meanly influenced by the NP (Population Size) parameter. It has been observed that, increasing NP for DE application leads to the best performances, while decreasing NP favors ACO in terms of results. However, DE demonstrates its strength in reaching satisfactory solutions with fewer iterations or NFE (Number of Function Evaluations) which means less computational time.

General conclusion:

In conclusion, the gas flow rates from condensate reservoirs through wellhead choke has been successfully predicted by this study using differential evolution (DE) where satisfactory results were extracted.

In this study we have seen many models of gas flow rates prediction but only two of them have been selected Leal et al (2013) and S.Seidi et al (2015) according to the available data's which are the 72 dataset from GTFT (Ilizi) and 39 dataset from several wells (province of Iran), where both of models have been extracted from the first model Gilbert (1954).

A lot of optimization strategies have been explored as well while the selected one was standard (DE), the reason behind selecting it is its rapidity in reaching the optimal result which means saving more time. The algorithm parameter setting were chosen after experiments where five proposal parameter setting by several studies have been tested and the best results were Cr=0.9, F=0.5 and NP=150.

DE algorithm efficiently explores the search space and effectively exploits potential solutions, ultimately resulting in faster optimization. These findings highlight the importance of selecting the appropriate algorithm and parameter values based on the specific optimization problem at hand, considering the trade-off between exploration and exploitation. After experiments by using (DE) in this study, we reached good results represented in the performance of the algorithm (choosing the best parameter setting) and finding the best constants values for both models (Leal and Seidi) using for each model (GTFT and Fars province of Iran) data's, respectively. A noticed improvement in the results of the values obtained comparing by the values proposed in the previous studies, which indicates the success of this study.

The obtained results which represented in extracting the unknown parameters (β1 to β5) for Leal model and (a,b,c and d) for Seidi model, gave a better performance in predicting the gas flow rates where the error (OF) has minimized from 0.3042 to 0.2875 for Leal model comparing by the ACO study and from 0.9700 to 0.4595 for Seidi model comparing by Seidi and ACO studies.

References:

[1] Siamak Seidi , Tofigh Sayahi ,"A new correlation for prediction of sub-critical two-phase flow pressure drop through large-sized wellhead chokes" 2021.

[2] Two-Phase Flow Through Chokes R. Sachdeva; October 1986.
https://doi.org/10.2118/15657-MS

[3] Multiphase Flow Through Chokes - An Evaluation of Frozen, Equilibrium, and Nonequilibrium Flow Models Carl-Martin Carstensen One Subsea, a Schlumberger Company, Sandslikroken 140, N-5254.

[4] H.Ghorbani et al., "Development of a new comprehensive model for choke performance correlation in iranian gas condensate wells," Advances in Environmental Biology, vol. 8, no. 17, pp. 308-313, 2014.

[5] B. Guo, W. Lyons, and A. Ghalambor, "Petroleum Production Engineering. Elsevier Science and Technology Books," 2007.

[6] Kaydani H et al,"Wellhead choke performance in oil well pipeline systems based on genetic programming. J. Pipeline Syst". (2014).

[7] Omid Hazbeh,Hamzeh Ghorbani,"Proposing a New Model for Estimation of Oil RatePassing Through Wellhead Chokes in an IranianHeavy Oil Field",November 16, 2022, http://dx.doi.org/10.2139/ssrn.4278876.

[8] F. Latif, S. Castrup, and A. Al Kalbani, "Field evaluation of MOV adjustable steam chokes,"2012: OnePetro.

[9] Hamzeh Ghorbani , Hossein Shojaei Barjouei1,"Prediction performance advantages of deep machine learning algorithms for two-phase flow rates through wellhead chokes" 23 February 2021.

[10] W.R. (1954). Analysis of Two-Phase Flow in Arbitrary Configurations. AIChE Journal, 1(4), 511-514.

[11] Hagedorn, A.R., and Brown, K.E. "Experimental Study of Two-Phase Flow in Vertical Wells." Journal of Petroleum Technology, 17(5), 607-617, (1965).

[12] Beggs, H.D., and Brill, J.P. (1973). A Study of Two-Phase Flow in Inclined Pipes. Journal of Petroleum Technology, 25(5), 607-617.

[13] Taitel, Y., and Duckler, A.E. (1976). "A Model for Predicting Flow Regime Transitions in Horizontal and Near Horizontal Gas-Liquid Flow." AIChE Journal, 22(1), 47-55.

[14] Zhang, H.Q., et al. (1998). "A New Model for Predicting Flow Pattern Transitions in Gas-Liquid Two-Phase Flow." Chemical Engineering Science, 53(7), 1379-1396.

[15] Abdul-Majeed, G.H., and Sarica, C. (2004). "A New Model for Predicting Flow Regime Transitions in Pipes." Journal of Energy Resources Technology, 126(2), 96-104.

[16] Oliemans, R.V.A., et al."State-of-the-Art Multiphase Flow Models for Wells." SPE Production & Facilities, 23(1), 15-29. (2008).

[17] Hidrobo, D., et al. (2012). "A New Methodology for Modeling Gas-Liquid Flows in Wellbores." SPE Production & Operations, 27(2), 134-145.

[18] Oliemans, R.V.A., et al. (2017). "Multiphase Flow Modeling in the 21st Century." SPE Journal, 22(3), 866-881.

[19] Mariella Leporini1 ,Alessandro Terenzi1 ,Barbara Marchetti ,"Improvement of a multiphase flow model for wellhead chokes under critical and subcritical conditions using field data",26 February 2021.

[20] Giacchetta G, Leporini M, Marchetti B, Terenzi A (2014) Numerical study of choked two-phase flow of hydrocarbons fluids through orifices. J Loss Prev Process Ind 27:13–20.

[21] Kargarpour, M.A., 2019. Oil and gas well rate estimation by choke formula: semianalytical approach. Journal of Petroleum Exploration and Production Technology (9), 2375–2386.

[22] Leal, J., Al-Dammen, M., Villegas, R., Bolarinwa, S., Aziz, A., Azly, A., Buali, M., Garzon, F., 2013. A new analytical model to predict gas rate volume measurement through well head chokes. In: Paper IPTC 17046 Presented at the InternationalPetroleum Technology Conference Held in Beijing, China, 26 – 28 March 2013.

[23] Jumaah, H.A., 2019. Modify chock performance equation for tertiary reservoir wells in khabaz oil field. 1st international conference on petroleum technology and petrochemicals. IOP Conf. Ser. Mater. Sci. Eng. 579, 012023, 2019.

[24] Hamzeh Ghorbani, Jamshid Moghadasi, David A. Wood ,"Prediction of gas flow rates from gas condensate reservoirs through wellhead chokes using a firefly optimization algorithm" Volume 45, September 2017,https://doi.org/10.1016/j.jngse.2017.04.034.

[25] morsli Kacem,"Prediction of Gas Flow Rate Through Wellhead Chokes In Condensate Reservoirs Using Ant Colony Optimization",2021.

[26] K. Price, Differential Evolution: a fast and simple numerical optimizer, in: Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, 1996, pp. 524–527, https://doi.org/10.1109/NAFIPS.1996. 53479– R. Storn, K. Price, Differential Evolution –- a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359, https://doi.org/10.1023/A%3A1008202821328

[27] K. Price, Differential Evolution: a fast and simple numerical optimizer, in: Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, 1996, pp. 524–527, https://doi.org/10.1109/NAFIPS.1996. 53479– R. Storn, K. Price, Differential Evolution –- a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359, https://doi.org/10.1023/A%3A1008202821328

[28] M. Epitropakis, D. Tasoulis, N. Pavlidis, V. Plagianakos, M. Vrahatis, Enhancing Differential Evolution utilizing proximity-based mutation operators, IEEE Trans. Evol. Comput. 15 (1) (2011) 99–119, https://doi.org/10.1109/TEVC.2010. 2083670l

[29] W. Gong, Z. Cai, Differential Evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081, https://doi.org/10.1109/TCYB. 2013.2239988

[30] A. Zhang, J. Sanderson, JADE: adaptive Differential Evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[31] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Transactions on evolutionary computation 13 (3)

[32] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79

[33] Cui, L., Li, G., Lin, Q., Chen, J., Lu, N., 2016. Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. Comput. Oper. Res. 67, 155–173.

[34] Sun, G., Cai, Y., 2017. A novel neighborhood-dependent mutation operator for differential evolution. In: Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, Vol. 1, pp. 837–841

[35] Choudhary, N., Sharma, H., Sharma, N., 2017. Differential evolution algorithm using stochastic mutation. In: Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016, pp. 315–320.

[36] Yu, X., Yu, X., Lu, Y., Yen, G.G., Cai, M., 2018. Differential evolution mutation operators for constrained multi-objective optimization. Appl. Soft Comput. 67, 452–466.

[37] S. Wang, Y. Li, H. Yang, Self-adaptive mutation differential evolution algorithm based on particle swarm optimization, Appl. Soft Comput. 81 (2019), https://doi.org/10.1016/j. asoc.2019.105496 105496.

[38] L. Deng, L. Zhang, H. Sun, L. Qiao, DSM-DE: a differential evolution with dynamic speciation-based mutation for singleobjective optimization, Memetic Comput. 12 (1) (2020) 73–86, https://doi.org/10.1007/s12293-019-00279-0.

[39] S.M. Islam, S. Das, S. Ghosh, S. Roy, P.N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2) (2011) 482–500, https://doi.org/10.1109/ TSMCB.2011.2167966.

[40] Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with discrete mutation control parameters, Expert Syst. Appl. 42 (3) (2015) 1551–1572, https://doi.org/10.1016/j. eswa.2014.09.046.

[41] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. 33 (2015) 304–327, https://doi.org/10.1016/j. asoc.2015.04.019.

[42] N.M. Hamza, D.L. Essam, R.A. Sarker, Constraint consensus mutation-based differential evolution for constrained optimization, IEEE Trans. Evol. Comput. 20 (3) (2015) 447– 459, https://doi.org/10.1109/TEVC.2015.2477402

[43] Storn, Rainer, 1996. On the usage of differential evolution for function optimization.In: Proceedings of North American Fuzzy Information Processing. IEEE.

[44]] Storn, R., Price, K., 1997. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.

[45] Zaharie, D., 2007. A comparative analysis of crossover variants in differential evolution.Comput. Sci. Inf. Technol. 171–181.

[46] Zhao, S.Z., Suganthan, P.N., 2013. Empirical investigations into the exponential crossover of differential evolutions. Swarm Evol. Comput. 9, 27–36.

[47] Gong, W., Cai, Z., Wang, Y., 2014. Repairing the crossover rate in adaptive differential evolution. Appl. Soft Comput. J. 15, 149–168.

[48] Fister, I., Tepeh, A., Fister, I., 2016. Epistatic arithmetic crossover based on Cartesiangraph product in ensemble differential evolution. Appl. Math. Comput. 283, 181–194.

[49] Fan, Q., Zhang, Y., 2016. Self-adaptive differential evolution algorithm with crossoverstrategies adaptation and its application in parameter estimation. Chemom. Intell. Lab. Syst. 151 (1550), 164–171.

[50] Zou, D., Gao, L., 2012. An efficient improved differential evolution algorithm. In: Chinese Control Conference, CCC, pp. 2385–2390

[51] S. Hui, P.N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, IEEE Trans. Cybern. 46 (1) (2015) 64–74, https://doi.org/10.1109/TCYB.2015.2394466

[52] Q. Fan, Y. Zhang, Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation, Chemometr. Intell. Lab. Syst. 151 (2016) 164–171, https://doi.org/10.1016/j. chemolab.2015.12.020

[53]] L.-B. Deng, S. Wang, L.-Y. Qiao, B.-Q. Zhang, DE-RCO: rotating crossover operator with multiangle searching strategy for adaptive differential evolution, IEEE Access 6 (2017) 2970–2983, https://doi.org/10.1109/ACCESS.2017.2786347

[54] S.-M. Guo, C.-C. Yang, Enhancing differential evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (1) (2014) 31–49, https://doi.org/10.1109/ TEVC.2013.2297160.

[55] Y. Cai, J. Wang, Differential evolution with hybrid linkage crossover, Inf. Sci. 320 (2015) 244–287, https://doi.org/10.1016/ j.ins.2015.05.026.

[56] Y. Xu, J.-A. Fang, W. Zhu, X. Wang, L. Zhao, Differential evolution using a superior–inferior crossover scheme, Comput. Optimiz. Appl. 61 (1) (2015) 243–274, https://doi.org/10.1007/ s10589-014-9701-9.

[57] Yi, W., Zhou, Y., Gao, L., Li, X., Mou, J., 2016. An improved adaptive differential evolution algorithm for continuous optimization. Expert Syst. Appl. 44, 1–12.

[58] Gämperle, R., Müller, S.D., Koumoutsakos, P., 2002. A parameter study for differential evolution. Adv. Intell. Syst. Fuzzy Syst. Evol. Comput. 10, 293–298.

[59] J. Guo, Z. Li, S. Yang, Accelerating differential evolution based on a subset-to-subset survivor selection operator, Soft. Comput. 23 (12) (2019) 4113–4130, https://doi.org/10.1007/ s00500-018-3060-x

[60] P. Rakshit, Improved differential evolution for noisy optimization, Swarm Evol. Comput. 52 (2020) 100628, https://doi.org/10.1016/j.swevo.2019.100628

[61] K.M. Sallam, S.M. Elsayed, R.A. Sarker, D.L. Essam, Landscape-based adaptive operator selection mechanism for differential evolution, Inf. Sci. 418 (2017) 383–404, https://doi. org/10.1016/j.ins.2017.08.028.

[62] M. Tian, X. Gao, C. Dai, Differential evolution with improved individual-based parameter setting and selection strategy, Appl. Soft Comput. 56 (2017) 286–297, https://doi.org/ 10.1016/j.asoc.2017.03.010.

[63] B. Qu, J.J. Liang, Y. Zhu, P.N. Suganthan, Solving dynamic economic emission dispatch problem considering wind power by multi-objective differential evolution with ensemble of selection method, Nat. Comput. 18 (4) (2019) 695–703,https://doi.org/10.1007/s11047-016-95986.

[64] Sun, Y., 2017. Symbiosis co-evolutionary population topology differential evolution. In: Proceedings - 12th International Conference on Computational Intelligence and Security, No. 1, CIS 2016, pp. 530–533

[65] Wang, J., Zhang, W., Zhang, J., Member, S., 2016. Cooperative differential evolution with multiple populations for multiobjective optimization. IEEE Trans. Cybern. 46 (12), 2848–2861.Wei, W., Wang, J., Tao, M., 2015. Constrained differential evolution with multiobjective

[66] Di Carlo, M., Vasile, M., Minisci, E., 2015. Multi-population inflationary differential evolution algorithm with adaptive local restart. pp. 632–639.

[67] Aalto, J., Lampinen, J., 2015. A population adaptation mechanism for differential evolution algorithm. In: Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015, pp. 1514–1521

[68] Awad, N.H., Ali, M.Z., Suganthan, P.N., 2017a. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc., pp. 372–379

[69] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, A novel population initialization method for accelerating evolutionary algorithms, Comput. Math. Appl. 53 (10) (2007) 1605–1614, https://doi.org/10.1016/j.camwa.2006.07.013.

[70] A.B. Ozer, CIDE: chaotically initialized differential evolution, Expert Syst. Appl. 37 (6) (2010) 4632–4641, https://doi.org/ 10.1016/j.eswa.2009.12.045.

[71] V.V. de Melo, A.C.B. Delbem, Investigating smart sampling as a population initialization method for differential evolution in Differential evolution (DE) is a popular evolutionary algorithm 3867 continuous problems, Inf. Sci. 193 (2012) 36–53, https://doi. org/10.1016/j.ins.2011.12.037.

[72] W. Zhu, Y. Tang, J.-A. Fang, W. Zhang, Adaptive population tuning scheme for differential evolution, Inf. Sci. 223 (2013) 164–191, https://doi.org/10.1016/j.ins.2012.09.019.

[73] I. Poikolainen, F. Neri, F. Caraffini, Cluster-based population initialization for differential evolution frameworks, Inf. Sci. 297 (2015) 216–235, https://doi.org/10.1016/j.ins.2014.11.026.

[74] R. Jiang, J. Zhang, Y. Tang, C. Wang, J. Feng, A Collective Intelligence Based Differential Evolution Algorithm for Optimizing the Structure and Parameters of a Neural Network, IEEEAccess8(2020)6960169614,https://doi.org/10.1109/Access.628763910.1109/ ACCESS.2020.2986398

[75] B. Majhi, D. Naidu, Differential evolution based radial basis function neural network model for reference evapotranspiration estimation, SN Applied Sciences 3 (1) (2021) 1–19, https://doi.org/10.1007/s42452-020-04069-z.

[76] C.M. Saporetti, L. Goliatt, E. Pereira, Neural network boosted with differential evolution for lithology identification based on well logs information, Earth Sci. Inf. 14 (1) (2021) 133–140, https://doi.org/10.1007/s12145-020-00533-x

[77] G. Xiong, J. Zhang, X. Yuan, D. Shi, Y. He, G. Yao, Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm, Sol. Energy 176 (2018) 742–761, https://doi.org/ 10.1016/j.solener.2018.10.050.

[78] S. Dhabal, D.K. Saha, Image enhancement using differential evolution based whale optimization algorithm, Emerging Technology in Modelling and Graphics, Springer, 2020, pp. 619–628. 10.1007/978-981-13-7403-6_54

[79] M. Anuradha, V. Ganesan, S. Oliver, T. Jayasankar, R. Gopi, Hybrid firefly with differential evolution algorithm for multi agent system using clustering based personalization, J. Ambient Intell. Hum. Comput. 12 (6) (2021) 5797–5806, https://doi.org/10.1007/s12652-020-02120-w.

[80] M.B. Rosic´, M.I. Simic´, P.V. Pejovic´, An improved adaptive hybrid firefly differential evolution algorithm for passive target localization, Soft. Comput. 25 (7) (2021) 5559–5585, https:// doi.org/10.1007/s00500-020-05554-8.

[81] X. Zhang, Y. Xue, X. Lu, S. Jia, Differential-evolution-based coevolution ant colony optimization algorithm for Bayesian network structure learning, Algorithms 11 (11) (2018) 188, https://doi.org/10.3390/a11110188

[82] X. Xie, K. Xu, X. Wang, Cloud computing resource scheduling based on improved differential evolution ant colony algorithm, in: Proceedings of the 2019 International Conference on Data Mining and Machine Learning, 2019, pp. 171–177

[83] A. Trivedi, D. Srinivasan, S. Biswas, T. Reindl, A genetic algorithm–differential evolution based hybrid framework: case study on unit commitment scheduling problem, Inf. Sci. 354 (2016) 275–300, https://doi.org/10.1016/j.ins.2016.03.023.

[84] J. Thakshaayene, D. Kavitha, Unit commitment using hybrid genetic algorithm with differential evolution, 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), IEEE, 2017, pp. 1-6

[85] Y. Li, S. Wang, X. Hong, Y. Li, Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm, 2018 37th Chinese Control Conference (CCC), IEEE, 2018, pp. 4489- 4494.

[86] K. Mason, J. Duggan, E. Howley, A multi-objective neural network trained with differential evolution for dynamic economic emission dispatch, Int. J. Electr. Power Energy Syst. 100 (2018) 201–221, https://doi.org/10.1016/j. ijepes.2018.02.021.

[87] J. Kumar, A.K. Singh, Workload prediction in cloud using artificial neural network and adaptive differential evolution, Future Generation Comput. Syst. 81 (2018) 41–52, https://doi. org/10.1016/j.future.2017.10.047.

[88] A. Dahou, M.A. Elaziz, J. Zhou, S. Xiong, Arabic sentiment classification using convolutional neural network and differential evolution algorithm, Computat. Intell. Neurosci. 2019 (2019), https://doi.org/10.1155/2019/2537689.

[89] R. Li, H. Zhang, Q. Zhuang, R. Li, Y. Chen, BP neural network and improved differential evolution for transient electromagnetic inversion, Comput. Geosci. 137 (2020), https://doi.org/10.1016/j.cageo.2020.104434 104434.

[90] S. Sarbazfard, A. Jafarian, A hybrid algorithm based on firefly algorithm and differential evolution for global optimization, Int. J. Adv. Comput. Sci. Appl. 7(6) (2016) 95–106. 10.14569/ ijacsa.2016.070612.

[91] T.K. Ghosh, S. Das, A novel hybrid algorithm based on firefly algorithm and differential evolution for job scheduling in computational grid, Int. J. Distributed Syst. Technol. (IJDST) 9 (2) (2018) 1–15, https://doi.org/10.4018/IJDST.2018040101.

[92] N. Rahmat, N. Aziz, M. Mansor, I. Musirin, Optimizing economic load dispatch with renewable energy sources via differential evolution immunized ant colony optimization technique, Int. J. Adv. Sci. Eng. Inf. Technol 7(6) (2017) 2012. 10.18517/ijaseit.7.6.2328.

[93] Z. Zhang, Y. Cai, D. Zhang, Solving ordinary differential equations with adaptive differential evolution, IEEE Access 8 (2020) 128908–128922, https://doi.org/10.1109/ ACCESS.2020.3008823.

[94] A. Onan, S. Korukogˇlu, H. Bulut, A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification, Expert Syst. Appl. 62 (2016) 1–16, https://doi.org/10.1016/j.eswa.2016.06.005.

[95] J. Hu, M. Wu, X. Chen, S. Du, P. Zhang, W. Cao, J. She, A multilevel prediction model of carbon efficiency based on the differential evolution algorithm for the iron ore sintering process, IEEE Trans. Ind. Electron. 65 (11) (2018) 88787https://doi.org/10.1109/TIE.2018.2811371

[96] L. Peng, S. Liu, R. Liu, L. Wang, Effective long short-term memory with differential evolution algorithm for electricity price prediction, Energy 162 (2018) 1301–1314, https://doi.org/ 10.1016/j.energy.2018.05.052

[97] Zainab Abdulelah Al-Sudani, Sinan Q. Salih, Ahmad sharafati, Zaher Mundher Yaseen, Development of multivariate adaptive regression spline integrated with differential evolution model for streamflow simulation, J. Hydrol. 573 (2019) 1–12, https:// doi.org/10.1016/j.jhydrol.2019.03.004

[98] Y. Wang, M. Filippini, G. Bacco, N. Bianchi, Parametric design and optimization of magnetic gears with differential evolution method, IEEE Trans. Ind. Appl. 55 (4) (2019) 3445– 3452, https://doi.org/10.1109/ICELMACH.2018.8507160

[99] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, H. Faris, MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems, Appl. Soft Comput. 97 (2020), https://doi. org/10.1016/j.asoc.2020.106761 106761

[100] A. LaTorre, M.T. Kwong, J.A. Garcı´a-Grajales, R. Shi, A. Je´rusalem, J.-M. Pena, Model calibration using a parallel differential evolution algorithm in computational neuroscience: Simulation of stretch induced nerve deficit, J. Computat. Sci. 39 (2020), https://doi.org/10.1016/j. jocs.2019.101053 101053.

[101] E.H. Houssein, M.A. Mahdy, M.G. Eldin, D. Shebl, W.M. Mohamed, M. Abdel-Aty, Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm, J. Adv. Res. 29 (2021) 147–157, https://doi.org/ 10.1016/j.jare.2020.10.001

[102] P.P. Biswas, P.N. Suganthan, G. Wu, G.A. Amaratunga, Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm, Renew. Energy 132 (2019) 425–438, https://doi. org/10.1016/j.renene.2018.07.152.

[103] S. O¨ zyo¨n, Optimal short-term operation of pumped-storage power plants with differential evolution algorithm, Energy 194 (2020), https://doi.org/10.1016/j.energy.2019.116866 116866.

[104] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, X.-Y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inf. Sci. 507 (2020) 67–85, https://doi.org/ 10.1016/j.ins.2019.08.040

[105] R. Rivera-Lo´pez, E. Mezura-Montes, J. Canul-Reich, M.A. Cruz-Cha´vez, A permutational-based Differential Evolution algorithm for feature subset selection, Pattern Recogn. Lett. 133 (2020) 86–93, https://doi.org/10.1016/j.patrec.2020.02.021

[106] M. Kaur, V. Kumar, L. Li, Color image encryption approach based on memetic differential evolution, Neural Comput. Appl. 31 (11) (2019) 7975–7987, https://doi.org/10.1007/ s00521-018-3642-7

[107] X. Sui, S.-C. Chu, J.-S. Pan, H. Luo, Parallel compact differential evolution for optimization applied to image segmentation, Appl. Sci. 10 (6) (2020) 2195, https://doi.org/ 10.3390/app10062195.

[108] H.M. Mustafa, M. Ayob, D. Albashish, S. Abu-Taleb, Solving text clustering problem using a memetic differential evolution algorithm, PLoS ONE 15 (6) (2020), https://doi.org/10.1371/ journal.pone.0232816 e0232816.

[109] J. Wu, L. Bie, N. Jin, L. Guo, J. Zhang, J. Tao, V. Sna´sˇel, Dual-frequency output of wireless power transfer system with single inverter using improved differential evolution algorithm, Energies 13 (9) (2020) 2209, https://doi.org/10.3390/ en13092209

[110] S. Wang, H.-W. Shen, H. Chai, Y. Liang, Complex harmonic regularization with differential evolution in a memetic framework for biomarker selection, PLoS ONE 14 (2) (2019), https://doi.org/10.1371/journal.pone.0210786 e0210786.

[111] M. Kaur, H.K. Gianey, D. Singh, M. Sabharwal, Multiobjective differential evolution based random forest for ehealth applications, Mod. Phys. Lett. B 33 (05) (2019) 1950022, https://doi.org/10.1142/S0217984919500222

[112] S. Jain, V.K. Sharma, S. Kumar, Robot path planning using differential evolution, Advances in computing and intelligent systems, Springer, 2020, pp. 531–537. 10.1007/978-981-15- 0222-4_50.

[113] J.-S. Pan, N. Liu, S.-C. Chu, A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning, IEEE Access 8 (2020) 17691–17712, https://doi.org/10.1109/ACCESS.2020.2968119

[114] M.F. Fateh, A. Zameer, S.M. Mirza, N.M. Mirza, M.S. Aslam, M.A.Z. Raja, Differential evolution based computation intelligence solver for elliptic partial differential equations, Front. Inform. Technol.ElectronicEng.20(10)(2019)1445–1456, https://doi.org/10.1631/FITEE.1900221.

[115] Z. Zhang, Y. Cai, D. Zhang, Solving ordinary differential equations with adaptive differential evolution, IEEE Access 8 (2020)128908–128922,https://doi.org/10.1109/ ACCESS.2020.3008823

[116] Global Data Oil & Gas Intelligence Center (www.globaldata.com),Report Code: GDOGAP02526-MP.

[117] Conybeare, C. E. B. (2011). Geomorphology of oil and gas fields in sandstone bodies. Elsevier.

[118] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, A novel population initialization method for accelerating evolutionary algorithms, Comput. Math. Appl. 53 (10) (2007) 1605–1614, https://doi.org/10.1016/j.camwa.2006.07.013.

[119] I. Poikolainen, F. Neri, F. Caraffini, Cluster-based population initialization for differential evolution frameworks, Inf. Sci. 297 (2015) 216–235, https://doi.org/10.1016/j.ins.2014.11.026.

[120] N.M. Hamza, D.L. Essam, R.A. Sarker, Constraint consensus mutation-based differential evolution for constrained optimization, IEEE Trans. Evol. Comput. 20 (3) (2015) 447– 459, https://doi.org/10.1109/TEVC.2015.2477402.

[121] E.H. Houssein, M.A. Mahdy, M.G. Eldin, D. Shebl, W.M. Mohamed, M. Abdel-Aty, Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm, J. Adv. Res. 29 (2021) 147–157, https://doi.org/10.1016/j.jare.2020.10.001.

[122] T. Hamdi, J.B. Ali, V. Di Costanzo, F. Fnaiech, E. Moreau, J.- M. Ginoux, Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm, Biocybernet. Biomed. Eng. 38 (2) (2018) 362–372, https://doi.org/10.1016/j.bbe.2018.02.005.