# Optimization of Rate of Reaction Using Nature Inspired Algorithms

*Thesis submitted in partial fulfillment of the requirements for the award of the degree of*

**Master of Engineering**

in

**Computer Science and Engineering**

submitted by

**Akash Shrivastava**
(Roll no: 801532003)

Under Supervision of

**Dr. Prashant Singh Rana**
**Mr. Shatrughan Modi**

**Thapar University, Patiala-147001, Punjab, India**
**June 2017**

# Candidate's Declaration

I hereby certify that the work, which is being presented in the thesis, titled **Optimization of Rate of Reaction Using Nature Inspired Algorithms**, in partial fulfillment of the requirements for the award of the degree of **Master of Engineering** and submitted to Thapar University is an authentic record of my own work carried out under the supervision of **Dr. Prashant Singh Rana** and **Mr. Shatrughan Modi**. I have also cited the source(s) of text(s)/figure(s)/table(s).

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any institution.

Akash

**Akash Shrivastava**

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Rana 11/8/2017

**(Dr. Prashant Singh Rana)**
Assistant Professor
CSED

Shatrughan Modi

**(Mr. Shatrughan Modi)**
Lecturer
CSED

The M.E. Viva-Voce examination of **Akash Shrivastava**, has been held on

11/08/2017

([redacted])

# Abstract

Chemical reaction involves transformation of reactant into product. Rate of chemical reaction is the measure of how fast these changes are taking place. Some reactions occur very rapidly, others very slowly. For example, ionic reactions are very fast, while those taking place in water treatment plant may last up to few days. Rate of reaction can also be defined as the speed at which a reaction happens. If a chemical reaction has high rate, it shows that molecules combine at a higher rate than the reaction which has a slow rate. The rate of chemical reaction can also depend on different types of molecules that are combining. If there is low concentration of an essential element or compound, the reaction will be slower.

Earlier Arrhenius equation was used to calculate the rate of reaction with some parameters. However, to know the exact reaction mechanism, actual reaction needed to be executed, which was not feasible and was time consuming. There are some simulators available for chemical kinetics which can compute the reaction rate though not in an optimal way.

This thesis focuses on the optimization of rate of chemical reaction using four different nature inspired optimization algorithms, i.e, Random Algorithm, Genetic Algorithm, Differential Evolution and Particle Swarm Optimization, in order to maximize the rate of chemical reaction. In addition, these algorithms aim to get the best possible reaction rate along with the input tuning parameters. Test results show that different algorithms perform significantly better for different reactions and have different convergence rate. In this thesis, a simulator is used which is a object-oriented software tool named Cantera to calculate the rate of reaction which uses modified Arrhenius equation.

***Keywords:*** Rate of chemical reaction, Nature Inspired Algorithms, Optimization, Genetic Algorithm, Differential Evolution, Particle swarm optimization, Convergence.

# Acknowledgements

First, I would like to express my deep gratitude towards my supervisors **Dr. Prashant Singh Rana and Mr. Shatrughan Modi** for their invaluable advice and encouragement at every step of my Master's program. Without their unfailing support and belief in me, this thesis would not have been possible. Their contribution to this thesis goes well beyond their role as an academic supervisor and includes constant support on a personal level, without which this journey may never have been completed, and for this, I am truly grateful. They are great mentors for my life as well.

I am also thankful to **Dr. Ashutosh Mishra**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work. He has always been supportive and provided us required information at regular intervals.

I would like to acknowledge **Dr. Maninder Singh**, Head, Computer Science and Engineering Department, Thapar University for setting good standards for his students and providing all the help and facilities that were essential throughout the journey. His encouragement time and again has helped students to achieve the set goals.

I will be failing in my duty if I do not express my gratitude towards **Dr. S.S. Bhatia**, Dean of Academic Affairs, Thapar University for making provisions of infrastructure such as library facilities, computer labs equipped with internet facilities, immensely useful for the learners to equip themselves with the latest knowledge in the field. I want to express my sincere thanks to **Dr. Niyati Baliyan**, visiting Assistant Professor, Thapar University for her support and motivation.

Above all, thanks to the Almighty, my family and friends for always being there for me and staying calm at times required.

**Akash Shrivastava**

# Table of Contents

**Title** **Page No.**

# List of Figures

**Figure No.** **Title** **Page No.**

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **GA** | Genetic Algorithm |
| **DE** | Differential Evolution |
| **PSO** | Particle Swarm Optimization |
| **ACO** | Ant Colony Optimization |
| **NP** | Non Polynomial |
| **NPC** | Non Polynomial Completeness |
| **NSGA** | Non dominated Sorting Genetic Algorithm |
| **CR** | Crossover Rate |
| **MR** | Mutation Rate |

# Chapter 1

# Introduction

This chapter is introductory of the work done in this thesis. Apart from stating the definitions, it underlines the objectives to carry out this work and the research motivation. The chapter is closed with thesis organization.

The work presented here, mainly focuses on the optimization of rate of chemical reaction. The speed at which a chemical reaction occurs is the rate of a reaction. If the molecules involved in chemical reaction combined slowly as compared to molecules combined at a higher speed then we can say that chemical reaction has a high rate[4]. This variation in speed of chemical reaction can be seen in various natural processes, such as combustion of fossil fuel and chemical reaction of some acids. These have different rates of chemical reaction. Some are slow processes such as carbonization of fossil fuels and growth of plants. These reactions take hundreds or thousands of years to complete while some reactions can happen in a second or less. The types of molecules in a chemical reaction which are combining to other molecules, can also affect the rate of reaction. If the amount (concentration) of essential element is low, then reaction will have slow rate. The process of transforming unstable substances into stable substance can also be called as chemical reaction. If we see how chemical reaction starts at microscopic view, first the unstable molecules having excessive energy interact with each other followed by some elementary reactions. Later, these molecules convert into stable molecules, i.e., with minimum energy. Rates of chemical reactions depend on nature of the reactants, temperature, presence of catalyst(s), and concentration.

$$Rate\ of\ reaction\ =\ \frac{Amount\ of\ reactant\ used\ or\ amount\ of\ product\ formed}{Time\ taken} \tag{1.1}$$

## 1.1 Factors That Affect Rate of Reaction

Following are the factors that affect rate of reaction.

- Pressure
- Temperature

- Concentration
- Presence of catalyst
- Solid reactants are broken into smaller pieces
- Nature of reactants
- Orientation of reacting species
- Surface area
- Intensity of light
- Nature of solvent

### 1.1.1 Pressure

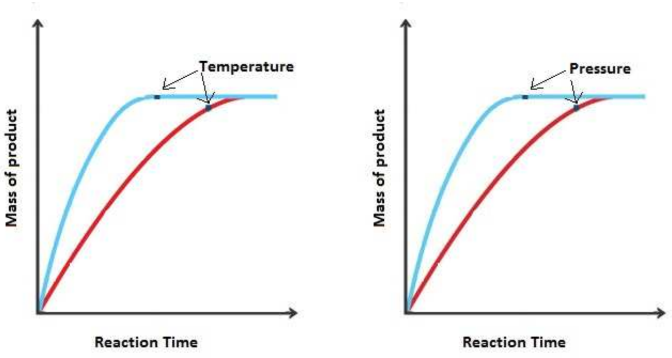
Reactions involving gas molecules have a significant effect of pressure. In these reactions, increase in pressure results in an increase in the rate of reaction as shown in Figure 1.1b. If we increase the pressure of a gas, that means the particles of gas are squeezed into smaller volume. However, in the case of reactions with only solids and liquids involved, there is no direct effect of pressure[5].

### 1.1.2 Temperature

For a reaction to happen, it is required that particles should collide[1]. If the substance is heated up, then its particles start to collide frequently and move faster which results in increasing the rate of reaction[6]. Thus, we can say that rate of chemical reaction is directly proportional to temperature shown in Figure 1.1a.

### 1.1.3 Concentration and Presence of Catalyst

A substance can react to another substance when their constituent particles such as atoms, molecules or ions contact each other. The reaction will not happen if there is no contact of these constituent particles. Hence, the possibility of reaction to occur is more if more reactant particles collide, per unit time. Consequently, the concentration of reactant particles increases, and the rate of reaction also increases.
A catalyst is a substance used in chemical reaction to increase the rate of reaction but the only thing which needs to be ensured is that, catalyst should never change the net



(a) Temperature (b) Pressure

Figure 1.1: Effect of Temperature and Pressure [1]

chemicals of the reaction[4]. Presence or absence of catalyst in decomposition of hydrogen peroxide is a good example for this. By accelerating some intermediate reaction, catalysts can increase the net reaction rate.

### 1.1.4 Presence of Different Type of Substances

The substances which take part in a reaction can also affect the rate of reaction individually. These substances have varying properties but we can generalize them as bond strength, bond type, molecular size and state of matter. Reactions involving solid substances can have effect of surface area as it is the exposed matter of a solid substance. The surface area of a solid substance is related to rate of reaction. Number of collisions per second increases as the surface area of solid reactant increases, as a result the rate of reaction increases[5].

### 1.1.5 Surface Area and Phase

If two reactants are in different phases that can not be mixed or in the case that one of them is in non soluble state then their particles collide less frequently as compared to the same reactants in fluid state. Number of collisions increase if the reactants involve in reaction are uniformly dispersed in single homogeneous solution. If the reactants are not in same phase, i.e., if they are heterogeneous then the collision occurs at interfaces among the phases, i.e., in some time interval. Rate of reaction directly depends on the number of collisions per unit time. In case of heterogeneous reactions, reaction rate can depend on surface area of the phases.

### 1.1.6 Presence of Solvent

The rate of reaction can also depend on the nature of solvent. Viscosity of solvent is important for rate of reaction. Particles diffuse at a slower rate if viscosity of solvent is high, whereas particles diffuse rapidly and collide frequently per unit time in less viscous solvents. Thus we can say that reaction rate is inversely proportional to the rate of reaction.

### 1.1.7 Medium or State of Matter

It is a matter of great concern that in which medium a reaction in taking place. There are chances that it can make difference according to the reaction whether the medium is solid, liquid and gaseous; organic or aqueous; or polar or non-polar. Reaction rate of the reactions involving solid reactants have high dependency on surface area, size and shape of reactants.

### 1.1.8 Interaction of Reactants

Reaction rate depends on how easily reactants can interact to each other, this interaction can increase or decrease the reaction rate. Table 1.1 gives a collective summarization of some factors that can affect the reaction rate. We can see that this dependency of any factor is not consistent throughout the process (from initial particles to final product). Sometimes, a factor which has high effect on a reaction till intermediate stage, does not affect the reaction, after a change in the parameter.

In this thesis, two factors from the above list are used, they are temperature and pressure. With the help of these two factors, we calculate the rate of chemical reaction. Cantera [7] tool provides facility to calculate the rate of chemical reaction by passing temperature and pressure as its parameters. Cantera has provided its library in Python environment. Optimization algorithms are implemented on Python2.7. For these algorithms, temperature and pressure are passed as their tuning parameters.

Table 1.1: Factors affecting rate of reaction

| Factors | Effect on Rate of Reaction |
|---|---|
| Temperature | Rate of reaction increases with increase in temperature but upto a limited extent |
| Pressure | Rate of reaction increases with increase in pressure |
| Catalyst | Catalyst lowers activation energy but increases the rate of reaction |
| Mixing | Mixing fastens the interaction of particles and hence the rate of reaction |
| Concentration | It depends on the type of reactant of which the concentration is increasing, it helps in regulating the rate of reaction |
| Medium | Solids have least reaction rate followed by liquids and gaseous particles can react rapidly |

## 1.2 Arrhenius Equation and Activation Energy

There are some reactions which take place faster as compared to others. To get the better understanding of this phenomenon, we need to get to microscopic level of chemical reaction. This theory is given by Svante Arrhenius[8].

### a. Reaction Mechanism

The series of events that take place as reactant molecules involved in reaction are converted into final product, is defined as the reaction mechanism. Every event consists of elementary steps which can be defined as dissociation (breaking) of molecule into smaller units or as the collision (combining) of the molecules. The newer molecules that are formed in every step might be the required product or might be the intermediate product. These intermediate products formed in elementary steps may not necessarily be present till the formation of final product, they can be destroyed in some other step and may not affect the present in reaction rate.

Chemical reaction mechanism describes events at molecular level in a sequence through which reactant change into products. Every step in a reaction involve 1, 2 or sometimes even 3 reactants. Reactions involving 1 reactant are known as unimolecular, whereas reaction involving 2 reactants are known as bimolecular and reactions which have 3 reactants are known as termolecular reactions.

### b. Collision theory of chemical change

Collision theory helps us to understand why there are not more than 3 reactants in a reaction and even why reactions which involve 3 reactants are extremely rare. It states that if there are 1000 collisions in a reaction then there will be only 1 collision which will involve 3 molecules colliding with each other. Reactions involving 4 reactants are so improbable that no one ever observed a reaction which involve 4 reactants.

Consider the bimolecular reaction: X+Y. If X and Y have to react then they must come close enough so that their existing bonds can be broken and new bonds can be formed.

The rate of collisions depends on concentration of each of reactants and if we double the concentration of either of reactants then rate of reaction will get doubled as well. If all the collisions happening in reaction will lead to products, then the rate of reaction will be of first-order in X and Y and $2^{nd}$ order overall rate = k[X][Y]

All collisions are not the same, when collision occurs between two balls, they will bounce back usually. A similar phenomenon will happen in case of a chemical reaction only if bonds between their atom is weaken significantly. To initiate a reaction effectively, collisions between two molecules should have enough energy such that the existing bonds can be broken and new ones can be formed. Moreover, the reactants should be oriented correctly during collision. Cantera uses Arrhenius equation for calculating rate of reaction[9].

Arrhenius equation is given by 5.1

$$r = Pe^{-E_a/(GK)} \tag{1.2}$$

where,

- r is the rate constant
- P is the pre-exponential factor
- $E_a$ is the activation energy for the reaction (in Joules)
- K is the absolute temperature (in Kelvin)
- G is the Universal Gas Constant

**Activation Energy:**

For a reaction to start, an energy is required which helps in interaction of the molecules of the particles involve in the reaction, this energy is called activation energy. Figure 1.2 shows that starting from the initial reactants, activation energy is required up to some point where it reaches and converted into some intermediate product, then molecules of the particles start colliding with each other after which we can observe, how the requirement of this activation energy decreases as it tends towards the formation of final product. Some of the reactions in which neutral molecules are involved can not take place or even start without getting the desired energy. Activation energy is also known as critical energy.

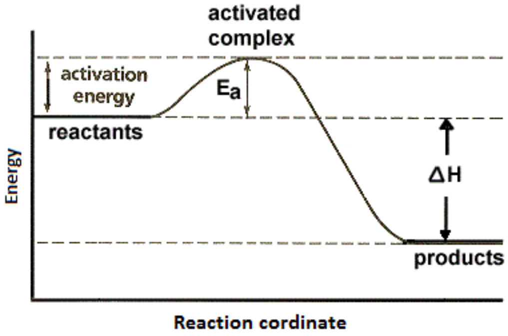


Figure 1.2: Activation Energy Diagram [2]

## 1.3 Research Motivation

Optimization is defined as the process to find the best alternative(s) among available solutions, which a given function can achieve in its domain. For example, one is interested to get best possible mixture ratio of a material to make a good structure of building. These type of problems aim to get desirable response that can be scientifically resolved through modeling and optimization. Modeling refers to converting the original problem into mathematical structure and to do so, the model must be designed in such a way that it includes all key characteristics of the original system. The created models are mainly written as procedures, called objective functions, in single or more options which corresponds to adaptable parameters of the system. In this thesis, main objective is to get the best possible rate of reaction using some optimization algorithms. These algorithms have input parameters, which are used as tuning parameters.

## 1.4 Organization of Thesis

This Thesis is structured into 7 chapters. A brief review of each chapter is provided below:

- **Chapter 1:** This chapter discusses about basic introduction to chemical reactions, rate of chemical reaction, factors that affect rate of reaction. It provides a preliminary information on the topic that is being accomplished by this Thesis. This section also provides a brief summary of all the chapters and acts as a guide for the reader as he/she can know briefly about each chapter and directly move to the chapter which is of interest.

- **Chapter 2:** This chapter describes the literature review of the problem. In this, Cantera, basics of optimization, different types of optimization problems (single objective and multi objective) and different approaches to solve these problems is presented. Subsequently, all the nature inspired algorithm are discussed.
- **Chapter 3:** This chapter focuses on the problem that is being tackled by the Thesis. It includes the research motivation and the key research area to be focused. This chapter talks about the gaps introduced through the research and also the objectives are formulated to carry out this research in a systematic way.
- **Chapter 4:** This chapter describes all the optimization algorithms which are applied in this work. Here, three Nature Inspired Optimized Algorithms are applied along with simulation using Cantera. Next, the methodology which is followed throughout the work is described.
- **Chapter 5:** This chapter describes the simulator named Cantera, Mainly how simulation is achieved to obtain the reaction rate, How to create some input files which are further passed to Cantera in order to fulfill the basic requirements for simulation.
- **Chapter 6:** This chapter describes the environmental setup to achieve the goal, parameter settings and tuning. Additionally, all the key results along with graphs are discussed.
- **Chapter 7:** This chapter contains the conclusion and future scope. There are possibilities that can be explored further to get some more specific and better results, they are explained here.

# Chapter 2

# Literature Review

## 2.1 Cantera

Cantera 2.3.0 [7] is a software tool to solve the problems in the fields of transport processes, chemical kinetics and thermodynamics among others. It provides classes of objects and interfaces among these objects representing phases of matter, reaction managers and time dependent reactor networks. Problems related to combustion, electrochemical energy conversion, storage and rate of chemical reaction can be solved using Cantera currently.

We can use Cantera from Forton 90 or C++ based application, Matlab and Python interfaces. In this thesis Python interface is used because there are many advantages of Python over any other language such as it covers almost all the features of basic C++ and object oriented concepts, in addition, it provides a flexible environment. The reason for choosing Python over Matlab is that Python can be downloaded free from the Internet but Matlab is commercial. Moreover Python module is commonly used nowadays in almost all the fields. Helps on various issues and queries can also be resolved easily on Internet.

Cantera uses modified Arrhenius equation for getting value of rate constant.

$$r = PK^n e^{-E_a/(GK)} \tag{2.1}$$

The above equation is the original Arrhenius equation where $n = 0$, which results in the value of rate constant between -1<n<1. There are several predictions on n. It has been pointed out that it is not feasible to establish, on the basis of temperature studies of the rate constant, whether the predicted $T^{1/2}$ dependence of the pre-exponential factor is observed experimentally. However, if additional evidence is available, from theory and/or experiment (such as density dependence), there is no obstacle to incisive tests of the Arrhenius law.

SK Khaitan and M Raju used Cantera to read and interpret the reaction mechanisms and set up a framework for evaluating the chemical reaction rates in dynamic simulation of air storage-based gas turbine plants[10].

S Vakalisa, D Prandoa, F Patuzzia, and M Baratieria created a multi box[11] approach in Cantera and they assessed the performance of the multi-box, by inserting real operating data (i.e. biomass composition, equivalent ratio, operation temperatures), collected during a monitoring activity that has been performed by the authors. The values which were returned from the model was compared with the actual measurements.

L Tao and Z Dongmei coupled OpenFOAM and Cantera to provide better simulation of combustion[12]. In this context, they coupled OpenFOAM and Cantera, and calculated the same H2-O2 premixed combustion case with steady solver coupled with Cantera and different transient solvers coupled and not coupled with Cantera. Through the comparison and analysis of different calculation results of the same case, they concluded that Cantera coupled with OpenFoam was better at simulating combustion. C Togb used Cantera for modeling 1-pentanol oxidation he simulated laminar flame speeds and also he used Cantera to evaluate densities[13].

## 2.2 Optimization

Optimization is the method to get a set of optimal points within the search space which makes the value of fitness function to be high or low depending upon the objective to be achieved and specification of problem. The process of getting the best collection of optimized values to satisfy an objective defined under constraints and presented as mathematical formulas is called Global optimization. Global optimization problems comes in the class of NP-Complete so there exists not even a single algorithm which can solve these problems in polynomial time[14].

## 2.3 Types of Optimization

Optimization problems are generally divided into two sections viz. continuous optimization and discrete optimization.

### 2.3.1 Continuous Optimization

In this type of optimization every variable is permitted to have values form sub-intervals of $(-\infty, \infty)$, i.e., real number line and these values are equally spaced. Global optimization algorithms require to get that value of selected variables which maximize/minimize the fitness function regarding all the allowed values. The continuous optimization is further classified into two parts, Unconstrained optimization and Constrained optimization. In unconstrained optimization there is no other condition specified to be fulfill and any relation can exist among decision variables for eg, non linear least squares, non linear equations etc. In constrained, optimization, a constraint condition should be fulfilled by any decision variable, if it is not fulfilled then this variable can not be selected. It includes the problems of quadratic, linear and network programming etc.

### 2.3.2 Discrete Optimization

This optimization makes the variables to take selected points from the interval $(-\infty, \infty)$ and these are required not to be equally spaced. Comparing with continuous problems, discrete problems are tough to solve. Two general techniques are taken into account to handle discrete optimization problems and are described as follows:

- Integer programming which is selected for problems where the solution is feasible only if some of variables are integer only.
- Stochastic programming which is also implemented for continuous optimization. In some real life problems the data can not be determined due to a number of limits. These kind of problems are perfectly handled by stochastic optimization [15].

## 2.4 Single Objective Optimization and Algorithms

A lot of real world problems can be formulated as optimization problems with only one objective under limited resources. In a multi solution condition, it is difficult to find global optimal solution using the conventional methods. Optimization as a key research topic can devise several approaches to solve these problems.

### 2.4.1 Optimization Problems

This section provides insight into common concepts of optimization in which Section 2.4.1.1 discuss the basic idea about optimization and Section 2.4.1.2 discusses the principle of single objective optimization.

#### 2.4.1.1 Introduction and Overview

Optimization as a key area in research is the process of getting best solution to the requested problem. Many real world problems can be formulated as optimization problem having global minima/maxima that is covered by local minima/maxima. For example

- In financial investment analysis, finding the best return over investment.
- In distribution system, to find the shortest route to travel between two points with an aim to minimize the transit time.
- In CPU scheduling, minimize the processor idle time while planning the execution of the processes.

For these types of problems, all feasible solutions can be sorted or ranked according to their objective values. Thus the goal of such optimization is well known beforehand and it is to find the best ranked solution within the complete set of feasible solutions[16].

#### 2.4.1.2 Single Objective Optimization

A single objective optimization problem can be formulated to find the distinct optimal solution of a function known as objective or fitness function. Thus optimization needs to find a vector of decision parameters that optimize the given objective function f(x) of a problem.

$$For\ a\ minimization\ case : f(\alpha) > f(x)$$

$$For\ a\ maximization\ case : f(\alpha) < f(x)$$

where X=[$x_1$,$x_2$,$x_3$,......,$x_n$] is the vector of n dimension decision variables of function f and $\alpha$=[$\alpha_1$,$\alpha_2$,$\alpha_2$,.......,$\alpha_n$] is the vector of decision variable such that objective function f is optimized. The final purpose of global optimization is to find vector of decision variables ($\alpha$) such that the function is optimized [16].

### 2.4.2 Single-Objective Optimization Algorithms

In support of the discussion about optimization problem in Section 2.4.1, this section describes the review and analysis of different kinds of algorithms developed to solve these problems. Classical techniques are presented in Section 2.4.2.1 and population based stochastic search methods are discussed in Section 2.4.2.2.

#### 2.4.2.1 Classical Techniques

Classical optimization techniques are many including linear programming, non linear programming, and dynamic programming. Heuristic technique is another optimization problem solving approach used by researchers. A heuristic is expressed as rule of thumb which provides a good solution for a difficult problem. The heuristic mainly ignores whether the solution is practically the best solution. This approach is mainly applied when there is no proper method to find the optimal solution or the method is not practically achievable using the present computing resources. This approach involves having some information about the problem being solved in order to design specific rules for solving the problem. Such need restricts the usage of general heuristic approach as it requires certain prior knowledge about the target problem. A metaheuristic technique is defined as good if it directs the process of searching in perfect direction. This reduces the time for repeated search in search process of good solution. Local or neighborhood search is the simplest technique used for metaheuristic. A neighborhood approach initializes with a randomly generated initial solution and then goes on generating neighborhood solutions. If a better solution is found, then it replaces the original solution and process is iterated. Such approach is similar to hill climbing process and generally covers local optimal solution. Several types of local search techniques have been proposed including the Tabu search and gradient approach [17].

#### 2.4.2.2 Population Based Stochastic Search Approach

Classic heuristic based approach is a very simple concept of problem solving. The rule based procedure fails most of the heuristic approach to solve some general problems. Heuristic approach solves only specific problems with specific characteristics and cannot be used to solve all problems related to optimization. Hence, population based stochastic search method is designed to solve all types of problems with a little change in the algorithmic procedure. In previous literature, a number of population based stochastic search methods are developed for optimization problems. Mainly GA algorithm, DE algorithm, PSO algorithm and ACO algorithm are the popularly known optimization algorithms [17]. Genetics is considered as one of the biological process that is attracting the researchers towards itself. As it is originated from nature, its optimization power has made it the basic of all optimization algorithms. Genetic algorithm which uses the process of gene evolution has always a high priority for developing most of the computation algorithms.

## 2.5 Multi-objective Optimization and Optimization Algorithms

This Section discusses multi-objective optimization. The difference between single objective optimization and multi objective optimization lies in the fact that the definition of optimum has to be re-stated. Algorithms used to solve multi-objective optimization produce a set of solutions in accordance with the multiple objectives of the problem.

### 2.5.1 Optimization Problems

This Section enlightens the general concepts of optimization, with the basic idea being discussed in Section 2.5.1.1. The principle behind multi-objective optimization is given in Section 2.5.1.2 and Section 2.5.1.3 discusses the fundamental concepts of multi-objective optimization.

#### 2.5.1.1 Introduction and Review

Real world problems require satisfaction of more than one objective at a time. In science, engineering, finance and many other domains problems are having more than one aims to satisfy. For example:

- In financial investment we have to balance risk and return simultaneously.
- In traveling problem a route is to be selected with short traveling time and minimize the relative cost.

- In scheduling, the minimization of idle time and maximization of output is required. These problems requires concurrent optimization of more than one objective. Moreover it is impossible to find an optimal solution which satisfies every objective of a problem at same time. A solution can be optimal to one objective but at the same time may not be optimal for another objective.

#### 2.5.1.2 Multi-Objective Optimization

For a multi-objective optimization more than one objective needs to be satisfied at the same time. A multi-objective optimization problem thus needs to find the vector of decision variables that optimize all the objective functions:

$$Minimize/Maximize f'(x) \tag{2.2}$$

$\acute{f}(x) = f_1(x), f_2(x), f_2(x), \ldots\ldots.., f_n(x)$ where $x = [x_1, x_2, x_3, \ldots\ldots.., x_n]$ is the n-dimensional decision vector representing the variables of objective function f. In multi-objective optimization it is almost impossible to find a single optimum solution to satisfy every objective as these objectives most of the time related to each other. Thus mostly a set of solutions is found instead of single solution to satisfy the requirements of different objectives [5].

#### 2.5.1.3 Concept of Multi-Objective Optimization

**a. Concept of Pareto Dominance:**

It is not necessary that multi-objective optimization provides a solution that is better than other solutions for objective function i.e. value of objective function can not be used for comparison. The comparison is done by using concept of Pareto dominance, i.e., two solutions are compared on the basis that which solution dominates the other solutions. Pareto dominance requires comparison of pair of solutions (s and t) with respect to the following conditions.

- In all objectives, t is worse than s

$$gi(s) > gi(t), \forall i \in 1, 2, ...., n \tag{2.3}$$

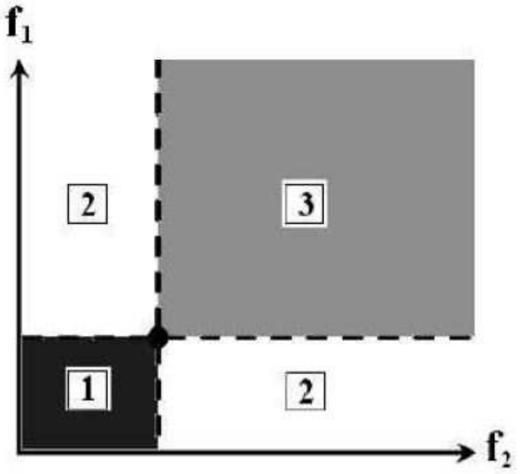


Figure 2.1: Relationship of solutions in the objective space [3]

- s is strictly better than t in some cases

$$gi(s) > gi(t), i \in [1, n] \tag{2.4}$$

If both the conditions are met then s is dominating over t and if any of the condition is violated then neither solution is dominating.

Figure 2.1 shows different types of solutions. In the above Figure black dot shows a feasible solution. Region 1 shows the solution which dominated the solution under observation, region 3 shows the solution which are dominated by observed solution and region 2 contains the solutions that are indifferent with observed solution.

**b. Pareto Optimality:**

Pareto optimality is the optimality criterion for multi-objective optimization and is summarized as the best solution that can be obtained without degrading at least one objective using the criterion of Pareto dominance. All solutions in the solution set are compared in pairs to get the dominating solutions. Non dominated solutions are those which dominate others but are not dominated by others. No improvement can be possible to these solutions without sacrificing the optimality of any objective function. Hence, they are the Pareto optimal solution set. Figure 2.2 gives an idea of Pareto optimality in which shaded region provides entire solution set and a black line at left bottom part provides the Pareto optimal solution set [18].

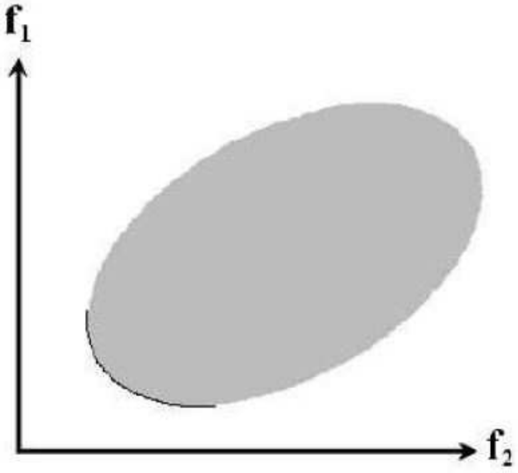


Figure 2.2: An objective space with Pareto optimality[3]

### 2.5.2 Multi Objective Optimization Algorithm

This Section discusses the classic concept of solving multi-objective optimization in section 2.5.2.1. Evolutionary algorithms are presented in Section 2.4.2.2 and NSGA II is discussed in Section 2.5.2.3.

#### 2.5.2.1 Classical Approach

Multi-objective optimization problems are solved using the two classical approaches. First one is the aggregate approach which aggregate all the objectives to form a single objective function. Second is the constraint approach which redefines the multi-objective problem to optimize one of the objective function and put constraint on residual objectives. This approach also convert the problem into single objective problem.

**a. Aggregate Approach** This approach merge all the objective functions with weighted addition of each of the objectives with user defined weights.

$$Minimize\ H(s) = \sum_{j=0}^{n} w_j h_j(x) \tag{2.5}$$

where w is the defined weight of objectives. The weights are selected on the basis of the relative importance of objectives. Hence single objective optimization techniques can be used to solve the problem. When this aggregated objective is optimized it is possible to get a particular solution for pre-defined weights. This is one of the simplest and widely used classical approaches but suffers a difficulty to assign weights to some objectives.

**b. Constraint Method** This method use the procedure to optimize one of the objective and put constraints on the residue objectives. It reduce the problem into single objective optimization as:

$$Minimize\ g(x) = gi(x) \geq j_i, i = 1, 2, ......, n\&i \neq z \tag{2.6}$$

where j is the upper bound of objectives and is converted into constraints. This method results in only a single solution instead of a set of solutions. The right selection of objective function for constraint method is difficult to be accurate. Another difficulty is the choice of the value of $j$. Too high a value results in poor solution, but too low a value leads to no solution found by the method [7].

#### 2.5.2.2 Evolutionary Approach

In classical approach, the multi-objective problem is solved by converting it into single objective problem. However, practically merging of objectives does not give the desired result as objectives are mainly inseparable. In this situation evolutionary algorithms can serve our purpose as they work with all objectives and produce a group of solution which is highly uniform by distributed and near to pareto optimal set.

**a. Fundamentals of Evolutionary Algorithms** Evolutionary algorithm are best suited to solve multi-objective problems due to their ability to produce many solutions in just one run. Solutions to multi-objective problem require a set of solution from the feasible search space and population based process of evolutionary algorithm is suited to these problems. The first advantage over classical method is the supported search in the search space that decrease the redundancy among solutions and less number of iterations are needed. Second advantage is that evolutionary algorithm finds similar solution rapidly using crossover and mutation and evaluates them quickly. Hence, these algorithms need very less computational resources in getting full set of solutions. Because of need of less computational resources, ease of implementation and less knowledge requirement, evolutionary algorithms are more suitable for multi-objective optimization problems.

#### 2.5.2.3 Non Dominated Sorting Genetic Algorithm II(NSGA)

NSGA II is a fast and elitist multi-objective optimization algorithm. It is a very famous algorithm among research community to solve multi-objective optimization problems due to its simplicity of operations and fast computation. The main features of NSGA II are:

1. A procedure for non dominated sorting where solutions are sorted regarding the non dominance level.
2. Elitism is implemented by storing all non dominated solutions hence increasing convergence among solutions.
3. A satisfactory self directing phenomenon is used using crowding distance to get diversity among solutions.

## 2.6 Nature Inspired Optimization Algorithms

Optimization algorithm is an iterative method, beginning with underlying assumptions. After a specific (adequately huge) number of cycles, it might tends towards a steady solution, or the ideal solution for the problem[19] [20]. Here states are solutions and attractors are converged solutions. Mathematical rules and equations are responsible for the evolvement of these self-organizing and iterative systems. Initially they are very complex, but with the help of some mathematical rules and equations, they interact and organize into some desired converged state with some self organization attributes. Thus the design of an optimum optimization algorithm is similar to searching an optimal way of evolving a system into self-organizing system[21].

**Evolutionary Operators** To know the exact working of an algorithm we need to go through all operators which are involved in the algorithm. In 1960s and 1970s J. Holland explained about Genetic Algorithms, that is based on Darwinian Evolution Theory. Genetic Algorithms have operators like selection, crossover, mutation and recombination[22]. Compared to previous traditional algorithms, GA has some advantages like parallelism, gradient-free and highly explorative nature. GA can solve some discontinuous and complex problems as it does not need any gradient/derivative information. The search space of GA is globally optimal and has more effective reach due to stochastic nature of crossover and mutation. Moreover GA can be implemented on parallel and complex problems as it is population-based algorithm and uses multiple set of solutions or chromosomes[23, 24].

The three main operators of genetic algorithms are:

**a. Crossover -** The combination of two previous solutions or chromosomes by altering the part of first solution into second and vice versa to get the new chromosome/solution.

**b. Mutation -** By changing some bits of a previous solution to generate the solution. Using binary encoding i.e. by swapping the bit 0 and 1, mutation can be achieved. Single site and multi-site mutation can be achieved simultaneously.

**c. Selection -** As it is discussed that GA is based on Darwinian theory of survival of the fittest, these algorithms select the best solution by comparing the fitness values of previous set of solutions and newly generated solutions, which implies they let best genes to pass on the next generations.

Mathematically, the process of crossover can be regarded as a mixing process where subspace is searched locally in an extensive nature. This can be better understood by the following example. Consider a problem of 6 dimensions with total search space of $\Omega = \Re^8$ , if we have got some parents say, p1=[aaaabb] and p2=[aaaaaa] where a and b can either be real values or even binary digits for any $j$th dimension of our problem's solution. No matter which crossover technique you use their child will be one of possible four combinations which are [aaaaba],[aaaaab],[aaaaaa] or [aaaaab]. As one can observe in any case first four variable are always [aaaa] here we can say that crossover can give a new solution only in $5^{th}$ and $6^{th}$ dimension and never a new solution will be different in first four dimensions. If we speak technically then we can say that crossover will give us a new solution in subspace S = [aaaaaa] $\bigcup \Re^2 \subset \omega$ in the present case. Crossover is basically a local search operator, however, if our subspace is large enough then it can become global operator too. When we talk about mutation, then it is always global search operator. For example, in the above case, if we mutate p1 then a new solution can be found which is not in our subspace. For example, if in p1 the second variable becomes b then we get a new solution, i.e., [abaabb] and this new solution is not present in our subspace. Sometime with mutation we can get a very different solution than our already existing solution this is why mutation is regarded as global operator. However mutation can become local search if we keep mutation rate very low. Both mutation as well as crossover provide variability so that we can get new solution. However when we talk about crossover it provide mixing only in subspace so its diversity is little whereas mutation provides better diversity. Nevertheless sometimes solutions which are away from our solution may make whole population away from converged/evolved characteristics.

### 2.6.1 Simulated Annealing

The simplest stochastic algorithm is probably the so-called simulated annealing, developed by S Kirkpatrick, C D Gelatt and M P Vecchi, in 1983 [25] based on the characteristics of the metal annealing process. From the current solution or state $x^i$ , a new solution $x^j$ is accepted with a probability

$$p(x_i \rightarrow x_j | x(t) = x_i) = \frac{1}{Z} e^{-1/T(t) max(0, f(x_j) - f(x_i))} \tag{2.7}$$

where f is the objective function to be minimized. Here, Z is a normalization factor. The original article by S Kirkpatrick et al . demonstrated how to solve very challenging problems. However, generating new solutions x j from the current solution may depend on the implementation and problem of interest. Whatever the ways of generation might be, such generations of new solutions form a Markov chain, or more specifically, a random walk. Therefore, the main operator is to generate new solutions by random walks, and consequently randomization acts as a mutation or explorative search mechanism. Selection is achieved by testing whether a solution is getting better (smaller for a minimization problem). Strictly speaking, simulated annealing is not an evolutionary algorithm, and thus there is no crossover operator in this algorithm. In addition, exploitation is relatively weak because the acceptance is carried out by a probability condition. That is why simulated annealing often converges very slowly in practice, though it is good at exploration and often has a good property of finding the global optimality at the expense of a large number of function evaluations.

### 2.6.2 Genetic Algorithms

Genetic algorithms (GA), developed by John Holland[26], essentially form the foundations of modern evolutionary computing. GA has three key genetic operators: crossover, mutation, and selection, as discussed earlier. Though there are no explicit mathematical equations in the original GA, it did provide detailed procedures and steps on how to generate offspring from parent solutions/strings. Crossover helps exploit and enhance the convergence. From empirical results and theoretical studies, all suggest a relatively higher probability $p_c$ for crossover in the range of 0.6 to 0.95, whereas the mutation probability $p_m$ is typically very low, around 0.001 to 0.05. These values correspond to a high degree of mixing and exploitation and a relatively lower degree of exploration. In practice, this means that genetic algorithms can often converge well and in many cases the global optimality can be achieved easily. The selection or survival of the fittest provides a good mechanism to select the best solution; elitism can guarantee that the best solution will remain in the population, which will enhance the convergence of the algorithm. However, the global optimality will be reachable under certain conditions, and mutation can be considered a double edged sword, it can increase the probability of finding the global optimality and at the same time slow down the convergence.

### 2.6.3 Differential Evolution

In 1997 R. Storn developed Differential Evoluion (DE)[27]. In previously used traditional pattern search algorithms, mutation operator was used to generate new solutions. Modern DE Algorithm is similar to this. Here we can see the generalized pattern search in Differential Algorithm by:

$$x_m = x_n + K(x_a - x_b) \tag{2.8}$$

Here a,b,m,n are four different integers generated by randomly. K is the differential weight in the range [0, 2].

### 2.6.4 Ant Algorithm and Ant Colony Optimization

M. Dorigo developed the Ant Algorithm[28], and the Ant colony optimization. It is based on social behaviour of ants. Initially, all algorithms related to ant theory use a chemical messenger pheromone for the indicator of quality, they use pheromone concentration. To implement the problems of interest, pheromone concentration helps to find the path and route by marking it. Higher concentration leads to better solution of the problem. If we compare Ant Algorithms with GA and DE, initial random route generation is quite similar to initial population generation in GA and DE. However, there is no explicit Mutation and Crossover applied.

### 2.6.5 Particle Swarm Optimization

In 1995, J Kennedy and R C Eberhart developed Particle Swarm Optimization (PSO)[29]. It is based on bird and fish schooling and swarm behavior. The set of solution or chromosome are called as particle. PSO uses the operators such as velocity and position of a particle and expressed as $v_i$ and $x_i$.

$$v_i^{t+1} = v_i^t + \alpha\epsilon_1[g^* - x^t] + \beta\epsilon_2[x_i^* - x_i^t] \tag{2.9}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2.10}$$

Here $\epsilon_1$ and $\epsilon_2$ are vectors and they take values 1 and 0 in every entry. $\alpha$ and $\beta$ are parameters for acceleration constants and learning parameters, for example they can be taken as $\alpha \approx \beta \approx 3$. We can see that new position is computed by applying pattern search based mutation by capmaring it with parent set of solutions. Global best of the present solution $g^x$ is and best solution of individual $x_i^6$ set are used to compute the selection process.

The global best of present set is more important than the individual best of current set solution as it has effect of particle swarm acceleration. We can see that it is computed after the accelerated PSO. On camparing with Genetiv algorithm, PSO does not have crossover in its operators, it consists of mutation and selection based on its own computations, which implies that PSO can have high exploration degree and very high mobility. In selection process, it uses $g^*$ which improves the performance efficiently. As it uses the current best $g^*$ for selection process so we can get the convergence at early stage of time but there are chances of premature convergence which means we may not get the true optimal results.

# Chapter 3

# Description of the problem

Chemical reactions are very essential part of our daily life. Fuel burning, iron smelting, making wine, cheese or drugs are some active examples that we might face but not acknowledge. These reactions have played integral part in our lives for centuries. On coming to the deep analysis of a chemical reaction, we need to understand some terminologies related to chemical reaction, such as rate of reaction, activation energy, factors that affect rate of reaction and how fast or slow they reactions can take place depends on various factors. Rate of reaction can be defined as how fast or slow a particular reaction take place. Our requirement for this rate of reaction depends on what type of reaction we want to perform, i.e. whether we want to maximize the rate of reaction or we want to minimize it.

## 3.1 Problem Statement

In this thesis, the main focus is on maximizing the rate of chemical reaction. It is not feasible to check the rate of chemical reaction on all possible set of inputs by performing the actual chemical reaction. Cantera allows to simulate these chemical reactions by applying some inputs in the form of its parameters by which we can get the rate of reaction without performing the actual reaction.

## 3.2 Research Direction

This section focusses on the research gaps found in literature survey and objectives that are formulated.

### 3.2.1 Research Gaps

Following are the gaps that are identified during literature survey.

1. To check the reaction rate, we need to perform the actual reaction[30].
2. To know the reaction mechanism, available techniques use Arrhenius equations and its parameters. It is very costly and time consuming.
3. Simulation can be done to calculate the reaction rate from available simulators but they are not optimized[7].
4. There is no research presently available related to the optimization of rate of chemical reaction.

### 3.2.2 Research Objectives

The following research objectives are formulated.

1. To read and understand previous research that has been done in the field of chemical reaction kinetics ad optimization.
2. To check the reaction rate without performing the actual reaction.
3. To optimize the reaction rate (maximize in this case) according to our requirements using Nature Inspired optimization Algorithms.
4. For a desired reaction rate, determine values of input parameters of the reaction.

### 3.2.3 Research Methodology

1. Literature survey will be done to get the understanding of chemical kinetics and optimization.
2. Rate of reaction will be determined by using simulator Cantera.
3. By applying optimization algorithms, rate of reaction will be maximized.
4. To get the desired rate of reaction, we can set the desired rate in the optimization algorithm with number of iterations.

# Chapter 4

# Algorithms and Methodology

## 4.1 Algorithms

Following algorithms are implemented in this thesis.

### 4.1.1 Random Algorithm

Random optimization (RO) is a family of numerical optimization methods that does not use any type of operators in its functioning. These optimization are also called direct search algorithms.

**Algorithm 4.1** : Random Algorithm

```
Function RA():
    Initialize $x$ with random position in search space
    Calculate its fitness (distance from the actual solution)
while termination criteria is met do
    Sample a new position $y$ by adding a normally distributed random vector to the current position x
    if fitness(y) is better than fitness(x) then
        replace the solution, i.e., x=y
    end if
end while
```

### 4.1.2 Genetic Algorithm

A simple Genetic Algorithm (GA) works on the principle of *Select the best and discard the rest*[31]. This algorithm has some initial set of solutions or we can call them as chromosomes. These chromosomes can be altered for getting better solutions with the help of Crossover and Mutation.

To implement GA[32], it has to be started with a random population of set of chromosome. Then evaluate these chromosome and allocates reproductive opportunities with respect

**Algorithm 4.2** : Genetic Algorithm

```
Function GA():
    Initialize population
    Calculate Fitness
while $Fitness != DesiredValue$ do
    Selection
    Crossover
    Mutation
    Calculate Fitness
end while
```

to its result. More chances of reproduction should be given to those chromosomes which gives better solution towards the given target problem as compared to the chromosomes which gives poorer results. The goodness or fitness of chromosome is defined with the current population . The term GA's strict interpretation refers to a model introduced and investigated by J Holland [26] and his students. Most of the recent case also refers primarily to this model introduced by Holland as well as its variations.

### 4.1.3 Differential Evolution

Differential Evolution(DE) is an evolutionary algorithm, it is Stochastic, population-based optimization algorithm[33]. The basic difference between GA and DE is that GA depends on crossover, which uses the results of two previous chromosomes, while evolutionary strategies use mutation as the primary search mechanism. DE [34] is a parallel direct search method which utilizes N dimensions parameters for population P. It is an evolutionary algorithm, which allows functional parameters to be represented as floating points by which continuous optimization problems can be simplified. Here parameters are mutated using floating point arithmetic operations. Initial population is generated randomly from the specific range and should cover whole parameter space. It

**Algorithm 4.3** : Differential Evolution

```
Function DE():
    Initialize population
    Calculate Fitness
while $Fitness != DesiredValue$ do
    Mutation
    Recombination
    Calculate Fitness
    Selection
end while
```

consist of process called mutation in which we generate new parameter by adding the weighted difference between previous two parameter vectors[35]. If the new vector yields a better solution then it replaces the best available vector, this process is selection, this whole process has to be done on every population.

### 4.1.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a parallel evolutionary computation algorithm based on social behaviour personification. It is developed by J Kennedy and Eberhart[36]. It is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. It solves a problem by having a set of solutions or particles, and altering these particles in the search-space according to some formulae over the particle's position and velocity. All particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles[36].

**Algorithm 4.4** : Particle Swarm Optimization

```
Function PSO():
for each particle do
    Initialize particle
end for
while $maxiteration$ or $minimumerrorcriteria$ do
    for each particle do
        Calculate fitness value
        if current fitness value is better than best fitness then
            set current fitness as best fitness
        end if
    end for
    Choose the particle with best fitness
    for each particle do
        Calculate particle velocity
        Update particle position
    end for
end while
```

## 4.2 Methodology

We applied four optimization algorithms for maximizing rate of chemical reaction on following four chemical reactions.

Table 4.1: List of Reactions

| Reaction Id | Reaction |
|---|---|
| CR1 | $CO + O(+M) \Leftrightarrow CO_2(+M)$ |
| CR2 | $CO + O_2 \Leftrightarrow CO_2 + O$ |
| CR3 | $CO + OH \Leftrightarrow CO_2 + H$ |
| CR4 | $CO + HO_2 \Leftrightarrow CO_2 + OH$ |

For these reactions we first created inputs to Cantera from the given range of parameters, then simulate the rate of reaction by which we got some set of inputs (temperature, pressure) and outputs (rate of chemical reaction) then applied various optimization algorithms in order to get maximum rate of chemical reaction. Finally, we perform result analysis of different chemical reactions on different optimization algorithms.

Figure 4.1 shows the methodology followed to perform the above mentioned process.

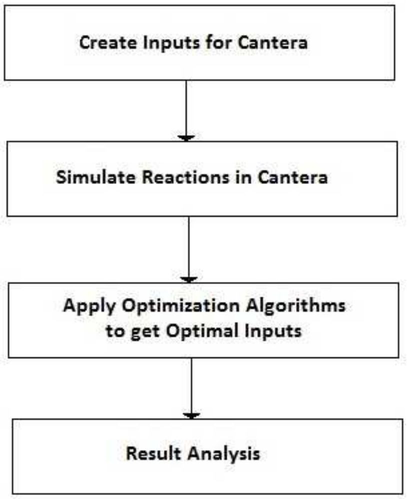


Figure 4.1: Methodology.

# Chapter 5

# Simulation

## 5.1 Cantera

This section describes the simulation process using Cantera and how to generate the input file along with its specifications and syntax. Cantera 2.3.0 [7] is a software tool that can enhance problem solving in the areas of kinetics, thermodynamics and transport phenomena. Various components can be inserted in Cantera like gas mixtures, specific reactor designs, kinetics mechanisms, reactivity of surfaces and other relevant files. Cantera is open-source software, thus it allows a wide range of interference from the user. Moreover Cantera can run on various environments like Matlab, FORTRAN, Python and C++. In our case the model is developed in a Python - Cantera environment although in the future it could be the case that Cantera will be also applied by the authors in other interfaces in order to process more complex mechanisms. Cantera may apply both stoichiometric and non-stoichiometric methods in order to calculate the thermodynamic equilibrium. The method of element potential minimization (non-stoichiometric) and the Villars CruiseSmith algorithm (stoichiometric) can be applied depending on the preference of the user. Cantera can be used to get rate of progress of chemical reaction at different temperatures and pressures.

Cantera uses modified Arrhenius equation for getting value of rate constant.

$$k = AT^n e^{-Ea/(RT)} \tag{5.1}$$

In the above equation, if we will put n = 0 then it will become same as original Arrehnius equation. Typically value of n lies between - 1 and 1. Various predictions for n are derived using theoretical analysis. It has been pointed out that, it is not feasible to establish, on the basis of temperature studies of the rate constant, whether the predicted $T1/2$ dependence of the pre-exponential factor is observed experimentally. However, if we can get some evidence, from theory and/or from excrement then we can perform incisive tests of Arrhenius law.

### 5.1.1 Input file

Input file contains directives and all the entries, syntax of both are quite similar to syntax of functions. Entry in input file can be defined as object like a phase, a species or a reaction. Directive set gives options, which illustrate the parameters of entry, for example how any error can be handled or default unit system.

Input file for Cantera is based on Python and follows the basic rules and syntax of Python. It would be very easy to understand the details of this if we are familiar with Python syntax and its terminologies by which we can directly jump to the portion of dimensions values.

A value can be assigned to the fields of entry of the input file. Following is the example of entry of species which has atoms and fields name and some more details.

species(name='C20', atoms='C:20')

There are some mandatory fields of an entry and a value must be assigned to it otherwise the process will stop and an error message will be shown. Some fields have default values if not given and some fields are optional.

#### 5.1.1.1 Species Name

The name field can not have the characters which are reserved keywords of XML, but can contain embedded parenthesis. Generally - or + signs are for indication of charge, or anything that can be treated as the name but not the keyword of XML. Following is the name specification example.

name = 'Carbon di Oxide'

name = 'CO2'

name = 'CO2(singlet)'

name = 'ar_2 +'

#### 5.1.1.2 Elemental Composition

In atoms entry, elementary composition is specified as follows.

atoms = 'O:2' # O2

atoms = 'S:1, O:2' # SO2 optional comma

atoms = ’H:2 S:1 O:4 ’ # Sulfuric acid H2SO4

atoms = ” # an empty specie representation

atoms = ”Xe:1 E:-2” # Xe++

For solid species solution, there is not any well defined way to define the species, it is same for surface area solution and liquid solution. It can be defined in many possible ways for above mentioned three types. In case of aqueous species, these can be defined according to solvation cage, i.e., without including water molecules or with including water molecules. However, for gaseous species, the species represent distinct molecules so the elementary composition is fixed.

In case of surface species, if the species represent empty surface site and is composed of nothing, it can leave the atoms field entirely. Emptiness in solids can be represented by this. Composition solely electrons can be described as a charged vacancy.

species(name = ’Yttria-Stabilized Zirconia–oxygen-vacancy’,

atoms = ’O:0, E:2’,

# ..., )

Here we can see that, ignoring an element is equivalent to giving that element an atom number of zero. Negative numbers can only be given to some element E that is for electron, otherwise the number of atoms should be zero or any positive number.

#### 5.1.1.3 Thermodynamic Properties

The ideal interface and phase entries implement particular models for which they are suitable according to their thermodynamic properties. For computation of these properties, every phase may use different expression. Thermodynamic properties are required for individual species. Following is an example of properties needed for this particular phase type implementation.

- The molar heat capacity at constant pressure $c^0p(T)$ for a range of temperatures and a reference pressure $P_0$
- The molar enthalpy $h^(T_0, P_0)$ at $P_0$ and a reference temperature $T_0$
- The absolute molar entropy $s^(T_0, P_0)$at $(T_0, P_0)$

#### 5.1.1.4 Species Transport Coefficients

There is a coefficient that defines the effect of species on phase transport properties, this coefficient is required in transport property model. An embedded entry can be assigned to transport field which gives species coefficient. Presently, gas transport type is the only entry type which supplies the parameters required in gas transport property model. Gas transport entry unit and the field values are compatible with transport parameters. No unit conversion is required to use the entries in transport database, they can be used directly as per the format explained. Only numeric values should be given without unit strings.

#### 5.1.1.5 Thermodynamic Property Models

Following is the entry type explanation to provide data for the thermodynamic field. Here each of the entry type implements different functional form or parametrization for heat capacity. Here we can see, it is not necessary that same parametrization is used by all species. Every species can explain the most specific dependency of temperature on heat capacity. To provide species properties specific for model of pure compound, ideal solution and ideal gas mixture, there are many types implemented.

#### 5.1.1.6 The NASA 7-Coefficient Polynomial Parameterization

The NASA 7-coefficient polynomial parameterization is used to compute the species reference-state thermodynamic properties $\hat{c}_p^0(T)$ , $\hat{h}^0(T)$ and $\hat{s}^0(T)$

The NASA parameterization represents $\hat{c}_p^0(T)$ with a fourth-order polynomial:

$c_p^0(T)/R = a_0{+}a_1T{+}a_2T^2{+}a_3T^3{+}a_4T^4$

$h^0(T)/RT{=}a_0 + \frac{a_1}{2}T + \frac{a_2}{3}T^2 + \frac{a_3}{4}T^3 + \frac{a_4}{5}T^4 + \frac{a_5}{T}$

$s^0(T)/R{=}a_0 \ln T + a_1T + \frac{a_2}{2}T^2 + \frac{a_3}{3}T^3 + \frac{a_4}{4}T^4 + a_6$

In original NASA equilibrium program, the old polynomial form which uses 7 coefficient in every temperature factor as described above, but the most recent versions of NASA equilibrium program with 9 coefficients for every temperature region are not compatible with the older version.

An embedded NASA entry defines a NASA parameterization. For two contiguous temperature ranges, two different NASA parametrization are used.

```
species(name = ”H2O”,

atoms = ” H:2 O:1 ”,

thermo = (

NASA( [ 300.00, 2000.00], [ 4.56892314E+00, -4.9865468E-02,

5.23568744E-4, -6.964454555E-05, 5.6848555E-7,

-3.26574155E+07, 6.23514788E+03] ),

NASA( [ 1200.00, 6000.00], [ 5.32587455E+02, 3.215455E-03,

-7.26565656E-09, 2.33545544E-08, -5.23265689E-08,

-3.26598987E+03, 5.68956556E+03] ) ) )
```

#### 5.1.1.7 The NASA 9-Coefficient Polynomial Parameterization

The NASA 9-coefficient polynomial parameterization [2] (NASA9) for short is an extension of the NASA 7-coefficient polynomial parameterization which includes two additional terms in each temperature region, as well as supporting an arbitrary number of temperature regions.

The NASA9 parameterization represents the species thermodynamic properties with the following equations:

$C_p^0(T)/R = a_0 T^2 + a_1^{T1} + a_2 + a_3 T + a_4 T^2 + a_5 T^3 + a_6 T^4$

$H^0(T)/RT = a_0 T^2 + a_1 lnT/T + a_2 + a_3/2T + a_4/3T^2 + a_5/4T^3 + a_6/5T^4 + a_7/T$

$s^0(T)/R = a_0/2T^2 a_1 T^1 + a_2 lnT + a_3 T + a_4^2 T^2 + a_5^3 T^3 + a_6/4T^4 + a_8$

The following is an example of a species defined using the NASA9 parameterization in three different temperature regions:

```
species(name=u’H2O’,

atoms=’H:2 O:1’,

thermo=(NASA 9([300.00,1200.00],

[ 5.1659752E+03, -5.82255675E+03, 6.26565956E+01,

3.25646454E-02, -1.954534432E-08, -8.52661516E-09,

3.15681121E-09, -3.2545441544E+03, -6.35454154E+01]),
```

```
NASA 9([1200.00, 5600.00],
[ 2.35656884E+04, -2.36478955E+02, 7.3655878844E+01,
-8.9652214535E-04, 5.235142587E-10, -2.159862147E-09,
7.235661488E-09, -2.65841278E+03, -3.25656417E+02]),
NASA9([5600.00, 19000.00],
[-2.1589321723E+08, 2.035698847E+07, -3.25614286E+03,
2.2566632541E-03, -1.2325641556E-07, 7.12526478E-09,
-9.0221515214E-09, -9.265614499E+08, 3.365695455E+04])),
```

NASA Thermo build provides data of thermodynamic for range of species. There are some input files available using web interface for a set of species. Later, these files, are converted into thermo NASA9 readable form.

For simulation, we first created an input file for Cantera. We got data about chemical reaction in above reaction from GRI 3.0 Then we fed this input file in Cantera and simulated different chemical reactions with different mole fractions of chosen chemical species. To generate data, random values of temperature, pressure and concentration of reactants were taken and these values were provided to Cantera to compute net production rate. The range of temperature and pressure is shown in Table 5.1.

Table 5.1: Description of Cantera Parameters

| Parameter | Lower bound | Upper bound |
|---|---|---|
| Temperature(Kelvin) | 200 | 5000 |
| Pressure(Pascal) | 8000 | 1000000 |

# Chapter 6

# Results and Analysis

## 6.1 Environmental Setup

The Cantera 2.2.1 package is used for calculating rate of chemical reaction. All the algorithms are implemented in Python 2.7 in Windows environment and executed on Intel core i3 with 1.7GHz processor and 4GB RAM.

## 6.2 Parameter Setting

The default parameter setting for each algorithm is given by Table 6.1.

Table 6.1: Parameter settings for maximization of rate of reaction

| **Algorithm** | **Parameters** |
|---|---|
| GA | Crossover Rate=0.9, Mutation Rate=0.1 |
| DE | Crossover Rate=0.9, F (Inertia) =0.5 |
| PSO | c1,c2=1.5,$w$=0.8 |

## 6.3 Results and Analysis

After applying optimization algorithms on four different chemical reactions, we found that every optimization algorithm can have a different effect on every different chemical reaction in order to get the maximum rate of chemical reaction.
On every single chemical reaction, we have applied our optimization algorithms with different number of iterations and their results are shown in the graphs.
In Figure 6.1, which shows the results of chemical reaction 1 over different algorithms with different number of iterations. Here we can see that, on increasing number of iterations, every algorithm has different rate of convergence. Here Genetic Algorithm performs best

Table 6.2: Results for CR1

| Algorithm name | Rate of reaction (kmol/m$^3$/s) | Temperature (Kelvin) | Pressure (Pascal) |
|---|---|---|---|
| Random Algorithm | 19422.56 | 201 | 239526 |
| Genetic Algorithm | **20323.36** | 578 | 997826 |
| Differential Evolution | 18757.3 | 760 | 995765 |
| Particle Swarm Optimization | 17223.55 | 716 | 945390 |

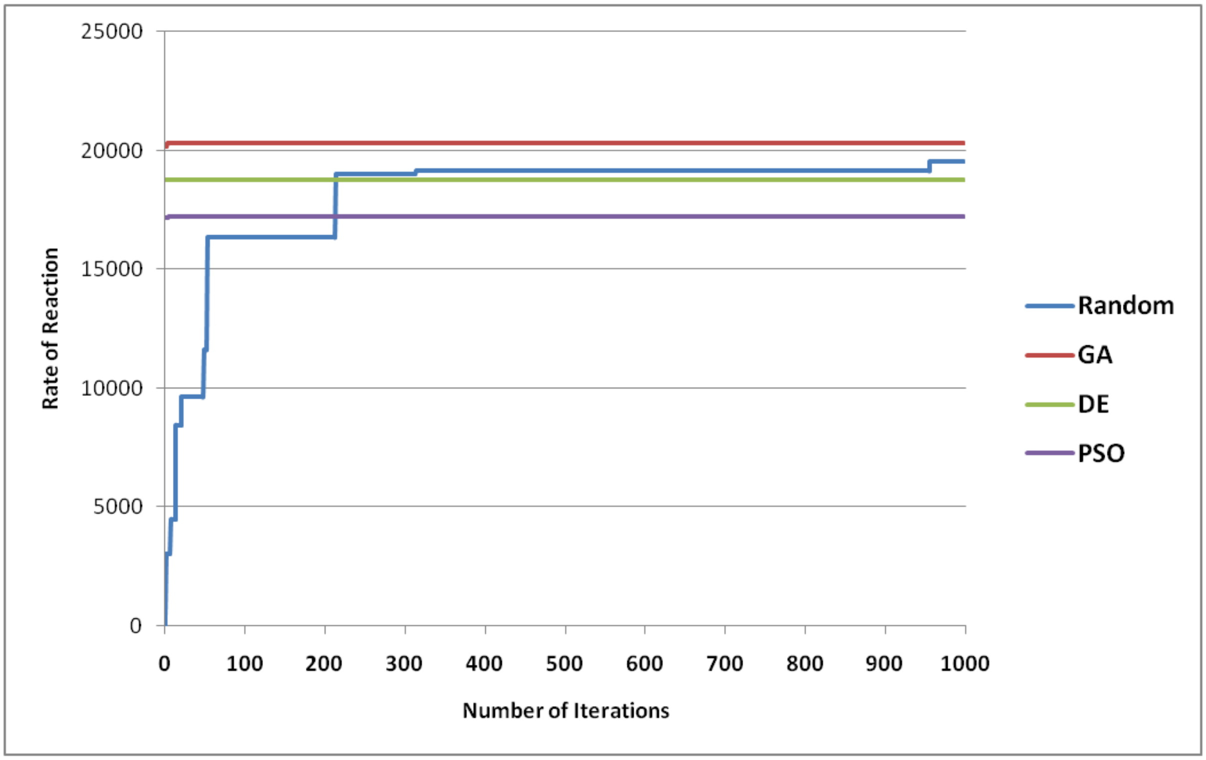


Figure 6.1: Getting maximum reaction rate on CR1 from different algorithms.

among all four algorithms as it converges faster than others.

Figure 6.2 shows the results of chemical reaction 2 over different algorithms with different number of iterations. Here Differential Evolution performs best among all four algorithms as it converge faster than others.

Figure 6.3 shows the results of chemical reaction 3 over different algorithms with different number of iterations. Here Genetic Algorithm performs best among all four algorithms as it converge faster than others.

Table 6.3: Results for CR2

| Algorithm name | Rate of reaction (kmol/m$^3$/s) | Temperature (Kelvin) | Pressure (Pascal) |
|---|---|---|---|
| Random Algorithm | 2693.35 | 4177 | 890799 |
| Genetic Algorithm | 2840.83 | 5000 | 982277 |
| Differential Evolution | **2944.27** | 5000 | 1000000 |
| Particle Swarm Optimization | 2801.89 | 4965 | 985123 |

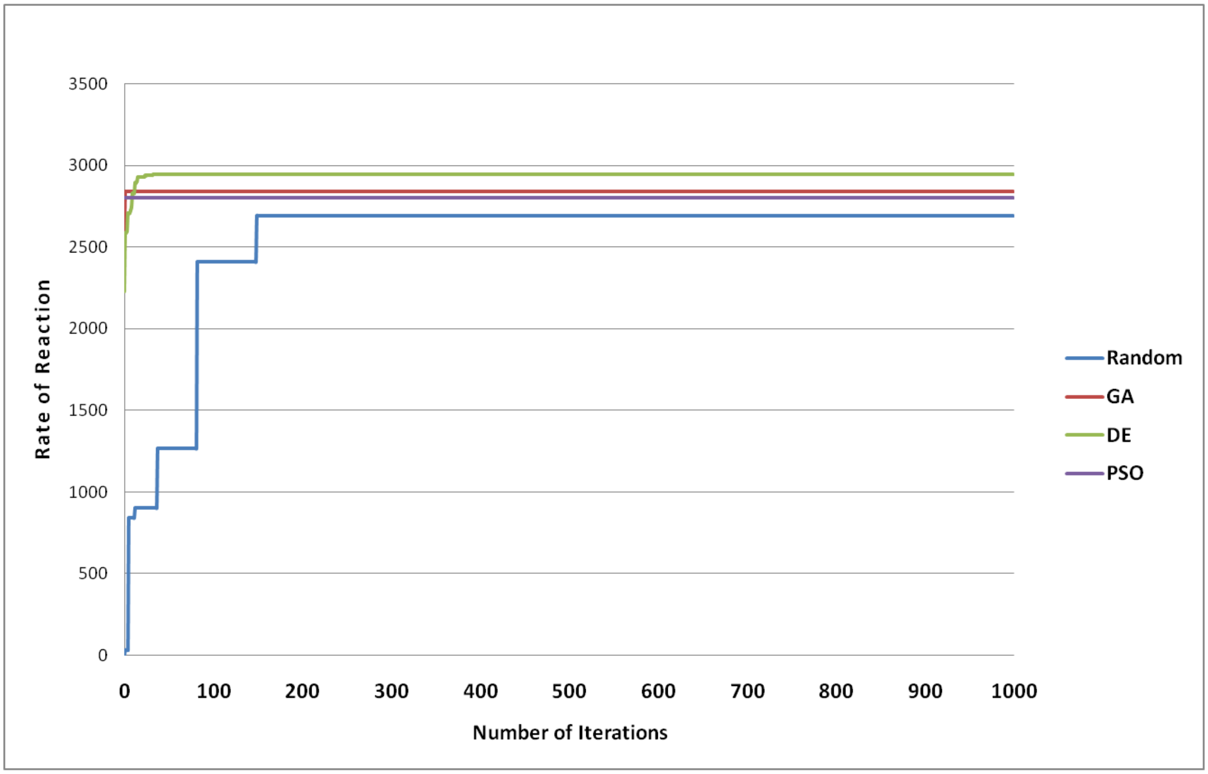


Figure 6.2: Getting maximum reaction rate on CR2 from different algorithms.

Figure 6.4 shows the results of chemical reaction 4 over different algorithms with different number of iterations. Here Genetic Algorithm performs best among all four algorithms as it converge faster than others.

Table 6.4: Results for CR3

| Algorithm name | Rate of reaction ($kmol/m^3/s$) | Temperature (Kelvin) | Pressure (Pascal) |
|---|---|---|---|
| Random Algorithm | 1888000 | 387 | 154572 |
| Genetic Algorithm | **2406799** | 200 | 998214 |
| Differential Evolution | 2134192 | 241 | 995119 |
| Particle Swarm Optimization | 1519841 | 259 | 859456 |

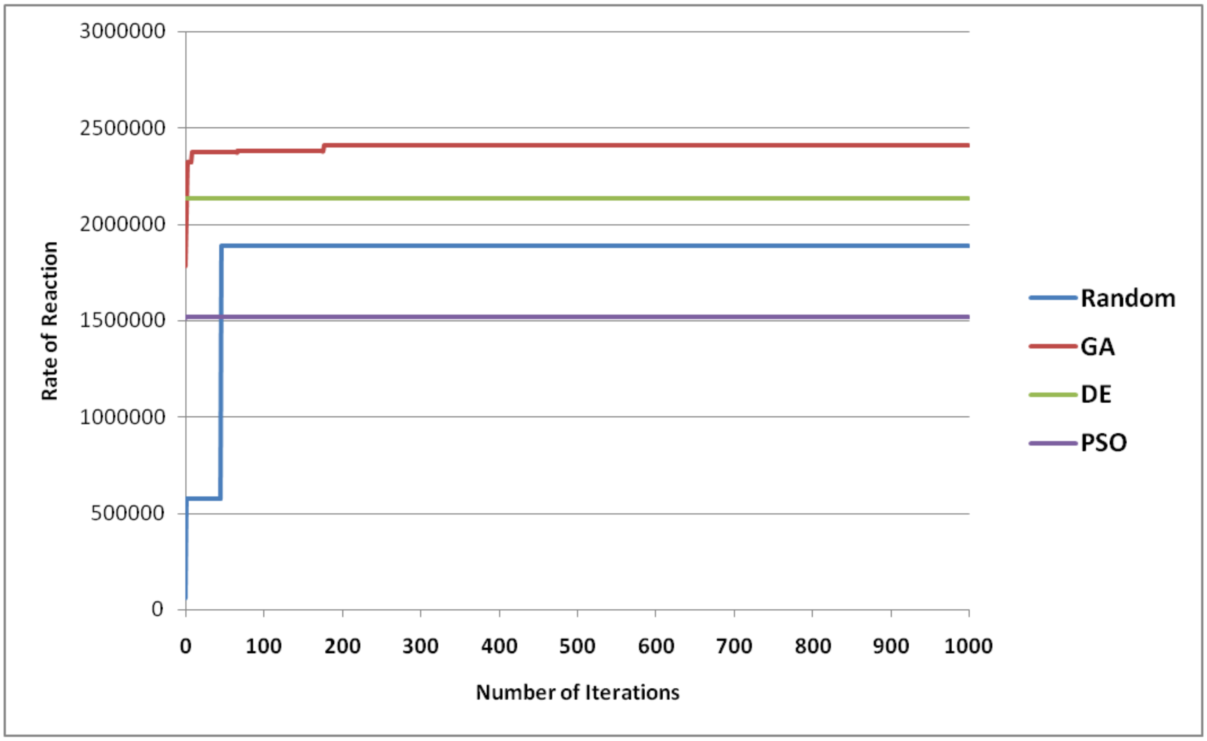


Figure 6.3: Getting maximum reaction rate on CR3 from different algorithms.

## 6.4 Comparative Performance

From the results shown above in all the tables and graphs, we can conclude the overall performance as the maximum rate of reaction obtained from each optimization algorithm applied over all four chemical reactions. Out of the four, Genetic Algorithm gives best maximum rate of reaction. Table 6.6 shows the collective performance from all the algorithms.

Table 6.5: Results for CR4

| Algorithm name | Rate of reaction ($kmol/m^3/s$) | Temperature (Kelvin) | Pressure (Pascal) |
|---|---|---|---|
| Random Algorithm | 1986938 | 3699 | 344542 |
| Genetic Algorithm | **1989877** | 5000 | 993045 |
| Differential Evolution | 1836383 | 4385 | 988266 |
| Particle Swarm Optimization | 1774615 | 4901 | 941506 |

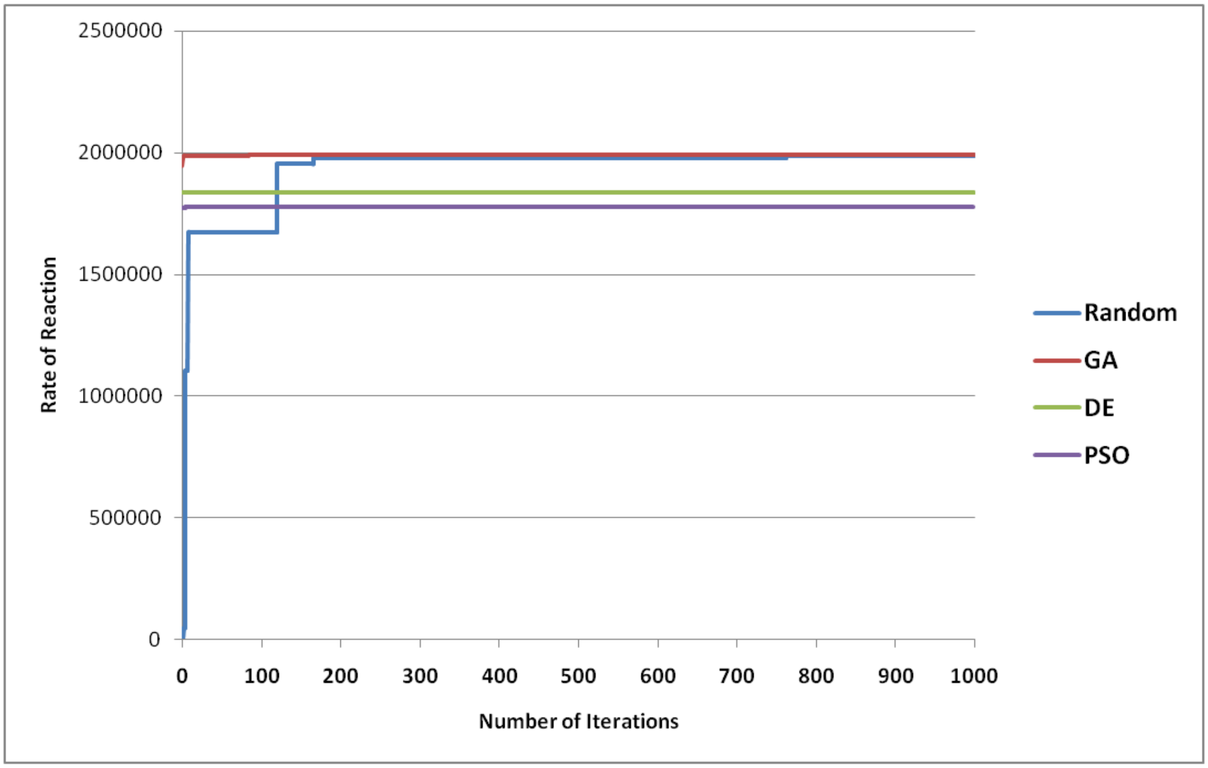


Figure 6.4: Getting maximum reaction rate on CR4 from different algorithms.

Table 6.6: Final result from all algorithms

| **S.No.** | **Reaction** | **Performance** |
|---|---|---|
| 1 | Reaction 1 | GA > PSO > RA > DE |
| 2 | Reaction 2 | DE > GA > PSO > RA |
| 3 | Reaction 3 | GA > DE > RA > PSO |
| 4 | Reaction 4 | GA > RA > DE > PSO |

# Chapter 7

# Conclusion and Future Scope

## 7.1 Conclusion

Optimization of rate of reaction basically states that we are trying to get the maximum rate of reaction within limited number of iterations by applying various optimization algorithms. In these algorithms, temperature and pressure are treated as the tuning parameters of algorithms. Here rate of reaction can be defined as the amount consumed or produced per unit volume and per unit time or per unit mass of reactants to a final product.

With the help of simulators, there is no need to perform the actual reaction to know the reaction mechanism. Available simulators can give the reaction rate but not in optimal way. Optimization algorithms are applied along with the simulation in order to get the best possible reaction rate. With the help of these algorithms, it can also be determined that, for some desired reaction rate, what should be the appropriate set of inputs. After applying all four algorithms over simulated data of reactions from Cantera, it is found that Genetic algorithms performs much better than all other algorithms in reaction 1, 2 and 4. In reaction 3, DE outperforms all the algorithms but GA has total best score.

## 7.2 Future Scope

In this thesis, simulation is done in Cantera using temperature and pressure as our tuning parameters for algorithms. Cantera is object-oriented tool in Python for simulation, so for future work, this can be integrate with some more parameters which affect the rate of reaction such as concentration, composition and presence of catalyst. We will be able to carry out more accurate simulations for chemical reactions.

# References

[1] Homer E Kissinger. Reaction kinetics in differential thermal analysis. *Analytical chemistry*, 29(11):1702–1706, 1957.

[2] Peter Atkins and Julio De Paula. *Physical chemistry for the life sciences*. Oxford University Press, USA, 2011.

[3] Mark Voorneveld. Characterization of pareto dominance. *Operations Research Letters*, 31(1):7–11, 2003.

[4] James H Espenson. *Chemical kinetics and reaction mechanisms*, volume 102. Citeseer, 1995.

[5] Octave Levenspiel and Cctave Levenspiel. *Chemical reaction engineering*, volume 2. Wiley New York etc., 1972.

[6] Homer E Kissinger. Variation of peak temperature with heating rate in differential thermal analysis. *Journal of research of the National Bureau of Standards*, 57(4):217–221, 1956.

[7] Dave Goodwin, Harry K Moffat, and Raymond L Speth. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. *Caltech, Pasadena, CA*, 2009.

[8] SR Logan. The origin and status of the arrhenius equation. *J. Chem. Educ*, 59(4):279, 1982.

[9] Finn Jensen. Activation energies and the arrhenius equation. *Quality and Reliability Engineering International*, 1(1):13–17, 1985.

[10] Siddhartha Kumar Khaitan and Mandhapati Raju. Dynamic simulation of air storage–based gas turbine plants. *International Journal of Energy Research*, 37(6):558–569, 2013.

[11] S Vakalisa, D Prandoa, F Patuzzia, and M Baratieria. Thermodynamic modeling of biomass downdraft gasifiers: Introduction to the" multi-box" concept. 2014.

[12] Liu Tao and Zhao Dongmei. Numeric simulation and analysis of h 2-o 2 premixed combustion based on openfoam. In *Robotics and Applications (ISRA), 2012 IEEE Symposium on*, pages 27–30. IEEE, 2012.

[13] Casimir Togbe, Philippe Dagaut, Fabien Halter, and Fabrice Foucher. 2-propanol oxidation in a pressurized jet-stirred reactor (jsr) and combustion bomb: experimental and detailed kinetic modeling study. *Energy & fuels*, 25(2):676–683, 2011.

[14] Aimo Törn, Montaz M Ali, and Sami Viitanen. Stochastic global optimization: Problem classes and solution techniques. *Journal of Global Optimization*, 14(4):437–

447, 1999.

[15] Yu M Ermoliev and RJ-B Wets. *Numerical techniques for stochastic optimization.* Springer-Verlag, 1988.

[16] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, 4(1):1–32, 1996.

[17] Anatoly Lisnianski and Gregory Levitin. *Multi-state system reliability: assessment, optimization and applications*, volume 6. World Scientific Publishing Co Inc, 2003.

[18] Yair Censor. Pareto optimality in multiobjective problems. *Applied Mathematics & Optimization*, 4(1):41–59, 1977.

[19] Xin-She Yang. Flower pollination algorithm for global optimization. In *International Conference on Unconventional Computing and Natural Computation*, pages 240–249. Springer, 2012.

[20] Gaige Wang, Lihong Guo, Hong Duan, Luo Liu, and Heqi Wang. A bat algorithm with mutation for ucav path planning. *The Scientific World Journal*, 2012, 2012.

[21] G Komarasamy and Amitabh Wahi. An optimized k-means clustering technique using bat algorithm. *European Journal of Scientific Research*, 84(2):26–273, 2012.

[22] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press, 1992.

[23] Xin-She Yang. *Nature-inspired metaheuristic algorithms.* Luniver press, 2010.

[24] Xin-She Yang. *Engineering optimization: an introduction with metaheuristic applications.* John Wiley & Sons, 2010.

[25] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[26] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[27] Rainer Storn. On the usage of differential evolution for function optimization. In *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*, pages 519–523. IEEE, 1996.

[28] Marco Dorigo, Gianni Di Caro, and Luca M Gambardella. Ant algorithms for discrete optimization. *Artificial life*, 5(2):137–172, 1999.

[29] James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 5, pages 4104–4108. IEEE, 1997.

[30] Peter Hänggi, Peter Talkner, and Michal Borkovec. Reaction-rate theory: fifty years after kramers. *Reviews of modern physics*, 62(2):251, 1990.

[31] Rajib Kumar Bhattacharjya. Introduction to genetic algorithms. *IIT Guwahati*, 12, 2012.

[32] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[33] Kelly Fleetwood. An introduction to differential evolution. In *Proceedings of Mathematics and Statistics of Complex Systems (MASCOS) One Day Symposium, 26th November, Brisbane, Australia*, 2004.

[34] Kenneth V Price. Differential evolution: a fast and simple numerical optimizer. In *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*, pages 524–527. IEEE, 1996.

[35] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[36] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

# Supplement Information

## Publications

1. Akash Shrivastava, Jagmeet Kaur, Abhishek Kapoor, Shatrughan Modi and Prashant Singh Rana, *"Optimization of Rate of Reaction using Nature Inspired Algorithms"*, National Conference on Breaking Barriers through Bioinformatics & Computational Biology. [Communicated]