OPTIMIZATION: A DISCUSSION FOR CHEMIST TO USE THE TOOL AS PROBLEM SOLVER

Abstract

A problem might be cast as one of optimization problem and then it can be solved by using a suitable optimization algorithm. The strategy is not very familiar to chemists, however is now becoming popular among chemists of different genre. In this chapter a discussion has been made on optimization that could be used by chemist. After a brief introduction a general outlook on optimization has been drawn. The subsequent section contains a review on different optimization algorithms. Both deterministic optimization algorithms have been discussed. Afterwards some scope of using optimization, especially stochastic optimization algorithms have been pointed out with the expectations that readers may find many more probable sectors for using these algorithms.

Keywords: Chemists, Optimization

Author

Srijeeta Talukder
Assistant professor
Adamas University
Kolkata, India

I. INTRODUCTION

Optimization is not a common word for chemists. However, we do use optimization in our research in some way knowingly or unknowingly. Thus, in this chapter a discussion would be portrayed on different optimization techniques and off course the some of application in chemistry. The aim is to make people convinced to use optimization in their research which would be very helpful in some cases and may be could not be ignored sometimes.

II. GENERAL DISCUSSION ON OPTIMIZATION

In this chapter I would offer an integrated view on optimization algorithms. The algorithm would be analysed in a general perspective so that one could easily program with respect to their requirement. After that a review on different optimization algorithms would be discussed. However, the review is not extensive, some of the important or well used optimization algorithms would be included.

Optimization refers to selecting the best element from a set of feasible alternatives. It essentially involves minimization or maximization of a function by methodically selecting inputs in accordance with a predetermined algorithm. There is a long history of developing different optimization algorithm. Initially, optimisation was used to address some geometrical problems. Fermat and Lagrange found calculus-based formulas for identifying optima,while iterative methods for moving towards an optimum was proposed by Newton andGauss. Historically, the first term for optimization was `linear programming' which was developed by George B. Dantzig. Although much of the theory had been introduced by Leonid Kantorovich in 1939 before Dantzig. The term programming is not related with anything concerning computer programming, rather it is used to refer to proposed training and logistics schedules of United State military. This was actually the problem studied by Dantzig. Later due to the volume of the problem dealt by optimization method, it is considered to be synonymous with computer programming. [1]

1. **Variables:** Depending upon the problem, the nature of variables that would be optimized (i,e the solution string) may differ. It may be a scalar or a vector. Another possible classification of variables depending upon the search space are integer or real numbers. For discrete search space the variable generally defined is an integer, whereas for continuous search space it would bereal. However according to the way an optimization problem is defined, the search space may be interchanged from real to integer or vice versa. In some cases, variables may be defined as binary numbers also. As an example, we may take the evaluation of most stable state of spin-glass system. Again, in the problem of Travelling-salesman the variables are in teger vectors. However, for most of the problems of chemists the optimizable variables are real numbers. Also, the problems discussed in this thesis mostly involve continuous search space of real numbers.

   Sometimes the choice of variables used may be dictated by the optimization method used. Such as the representation of variables in binary numbers while using genetical gorithms and a discrete search space is used for ant colony optimization. However, these two optimizers might be used in solving continuous real number search space also and there are many reports in the literature in this context.

2. **Objective Function:** Optimization problem must have an objective function [2, 3] to be optimized to get the solution. It is some sort of mathematical function of possible solution variable, generally known as cost function for minimization problem and fitness function for maximization job.

$$\text{Cost Function} \Rightarrow \min\{O(x)\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$
$$\text{Fitness Function} \Rightarrow \min\{F(x)\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

If the system move following a single criteria to find optima, it would be a single objective optimization and the optimal solution would be maxima or minima (global) with respect to the potential surface of the system. To optimize the system to a certain value V_l one may write the cost function as

$$O(\mathrm{x}) = [f(x) - V_L]^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$
$$\mathrm{F(x)} = e^{-O(x)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

However, it is not always possible to define an optimization problem in terms of single objective function. A problem may have more than one constraints. Objective function or cost function may explicitly contain multiple constraints for such kind of problem for searching the feasible solution. It is commonly known as *multi-objective* optimization problem. The corresponding objective function may be written as

$$O(\mathrm{x}) = f(x) + \sum_i \lambda_i g(i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Where e*x*ist the possible solution variable and $g(i)$ are there straints.

An optimization problem is to basically find out the optimal solution with respect to the objective. In multi-objective problem there are more than one objective. For two conflicting objectives, each correspond so a different optimal solution. When one would try to satisfy these two objectives simultaneously a set of optimal solutions appear which would be obtained by gain in one objective and loss in the other objective and viceversa. The objective function for the two conflicting objectives may be written as

$$\alpha f(x) + \beta g(x) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Where $\alpha$and $\beta$are the weight factor and the set of optimal solutions can be generatedby varying these two parameters. When $\alpha < \beta$the optimal solution has preference of the second over the first objective and viceversa. Varying α and β one can generate a pool of optimal solutions where n one of the solutions can be said to be better than the other. For an optimization problem with two completely conflicting objective, if these solutions are plotted with respect to the weight factor corresponding to the two objectives the curve generated is known as *parato-optimal front.*

But the objectives may not be conflicting always. In that case there should be an optimal solution which can satisfy all the objectives. Constraints are imposed as the form of functions of the variables, which are combined to define the complete objective function.

3. **Constraint:** Cases are there, where all the constraint to find the feasible solution, cannot be incorporated in the objective function for the optimization. Some limitations may be imposed by the nature of the problem or by the users o that certain objectives cannot be achieved. These constraints reduce the total search space to a *feasible search space*. The feasible search space is then actually sampled during optimization. Constraint can be mathematically expressed as the form of equality or in equality. Such as

$$\begin{matrix} g(x) = 0 \\ h(x) > min_h \end{matrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (7)$$

Another common type of constrain that can be imposed is the side constraints. This is one that simply bounds the range of values that a variable can take on.

4. **Algorithm of Search:** Finally for optimizing variables with respect to an objective function and ensuring all the constraints imposed, one needs an algorithm which would decide how the system moves to the optimal solution. There are a number of optimization algorithms available in the literature which have applications in various fields like economics, engineering, basic research of science etc. These algorithms vary with respect to the philosophy by which they are inspired, the nature of search process, the criteria for the acceptance of a move etc. In the following section we have discussed some of such optimization algorithms before coming to the application part.

III. CLASSIFICATION OF OPTIMIZATION ALGORITHMS

Optimization techniques may be categorized into two major classes with respect to the principle to deal with the problem, one is deterministic and another is by probabilistic approach. A move during optimization can be defined as

$$x_i = x_{i-1} + P_M \Delta x \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (8)$$

Both the schemes involve the iterative update of solution vector. In the above equation $_I$ is the solution vector at i^{th} iteration and P_m is the probability to change the vector from the previous iteration by Δx. But the form of P_m and Δx are different in the deterministic and probabilistic schemes. Randomness is incorporated into the choice of P_m and Δx in probabilistic algorithms.

Deterministic search process is usually very fast and quickly converge to the solution. But the limitation of the deterministic optimizer lies on its obvious search direction which basically causes local optimization. These searches are most often used if a clear relation between the characteristics of the possible solutions and their utility for a given problem exists. When the relation between the solution and the fitness or the cost function becomes complicated or the dimensionality of the search space is very high, deterministic search process will fail to find the global solution. Eventually for a system of even relatively small dimension, deterministic search would possibly involve exhaustive enumeration of the search space, which is not feasible. Also, in some practical problems such deterministic search direction is not available. All these reasons led to development of *Stochastic Optimization* techniques.

1. **Deterministic Search:** A deterministic search algorithm is an algorithm which, given a particular input, will always produce the same output. Thus, these algorithms are initial point dependent and generally involve enormous senumeration of the search space.

   A host of deterministic optimization algorithms exist[4],each with different applicability, efficiency, requirements, and robustness. One can make a classification among them that addresses unconstrained problem or more complicated constrained problem. Al though the problem with constraint can be cast as an unconstrained minimization problem even if the constraints are active. Another way to differentiate these algorithms is whether the algorithm employs only functional evolutions or gradient vector or even the Hessian matrix calculation with respect to the optimizable variables. Table 1 categorize some of the algorithms for deterministic search according to the mentioned criteria from which some have been briefly discussed in this chapter

Table 1: Classification of deterministic optimization methods [4]

	Unconstrained problems	Constrained problems
Only function evaluations	Golden section method	Simplex method
Function & gradient evaluations	Steepest descent method Conjugate gradient method Quasi-Newton method	Method of Lagrange multipliers
Function, gradient & hessian evaluations	Newton's method	sequential quadratic programming (sqp)

2. **Golden Section Method:** The golden section search [5] is a technique for finding the extremum (minimumor maximum) of a strictly unimodal function, i.e, a function with single extremum value. The process is carried out by narrowing the range of values of the variables in which the extremum is known to exist. From Figure 1, we can say that theminimum of the function `F' is in between x1 and x2 as the value of the function at $x3$ is less than both the values of the function at $x1$& $x2$. Now the Goldensearch method derives its name from the fact that the algorithm maintains the ratioof the distances between $x2 - x3$ and $x1 - x3$ same as the Golden ratio (1.61803...).

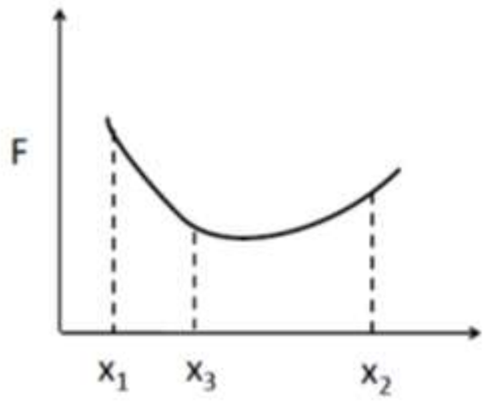

Figure 1: An unimodal one dimensional function

3. **Simplex Method:** Simplex method [6-9] was developed by George Dantzig in 1946. It provides us with a systematic way of examining the vertices of the feasible region to determine the optimal value of the objective function. It begins at an arbitrary corner of the solution set. At each iteration, the Simplex Method selects the variable that will produce the largest change towards the minimum (or maximum) solution. That variable replaces one of its compatriots that is most severely restricting it, thus moving the Simplex Method to a different corner of the solution set and closer to the final solution. In addition, the Simplex Method can determine if no solution actually exists. The algorithm is greedy since it selects the best choice at each iteration without needing information from previous or future iterations.

4. **Steepest Descent Method:** Steepest descent [10] is the simplest gradient based optimization technique. It is applicable to unconstrained problem with continuously differentiable search space. The move during the simulation is defined as

$$x_{n+1} = x_n - \gamma \nabla F(x_n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . (9)$$

Here $\nabla F(x_n)$is the gradient of the search space at $x_n$point and γ is a scalarquantity.γ would ideally be dependent on the iteration step and can be chosen in accord to some algorithm making the method more robust. The value of x at $(n+1)^{\text{th}}$ step depends on the gradient at n^{th} step. If the gradient becomes zero, the value of x will not be updated. Thus by updating$x$ iteratively using Eq (9) one can get the optimal solution when the gradient iszero. Although it is simple and easy to implement steepest descent method often suffer from convergence problem.

5. **Conjugate Gradient Method:** The efficiency of the steepest descent method can improve significantly by changing the search direction by extracting information about hessian from consecutive gradients[11, 12]

$$x_{n+1} = x_n - \gamma_n\big(\nabla F(x_n) - \beta \nabla F(x_{n-1})\big) \ldots\ldots\ldots\ldots . . \ldots\ldots\ldots\ldots\ldots\ldots\ldots . (10)$$

Where,

$$\beta = \frac{\|\nabla F(x_n)\|}{\|\nabla F(x_{n-1})\|} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

6. **Newton's Method:** Newton's method [13] involves the information of hessian along with function and gradient in iterative calculation. For pedagogical purpose, wecon sidera function with single variable. Applying Taylor series expansion

$$F(x_n + \Delta x) \approx F(x_n) + F^{'}(x_n)\Delta x + \tfrac{1}{2}F^{''}(x_n)\Delta x^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (12)$$

and attains its extremum when its derivative with respect to$x$ is equal to zero, i.e.when

$$\frac{F(x_n + \Delta x) - F(x_n)}{\Delta x} \approx F^{'}(x_n) + F^{''}(x_n)\Delta x = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (13)$$

Now $\Delta x = x_{n+1} - x_n$,
Then

$$x_{n+1} = x_n - \frac{F^{'}(x_n)}{F^{''}(x_n)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

It is worthwhile to notice that the Newton's method in optimization uses the second-derivative of the function because it is the zero-derivative equation that is solved by classical newton method. In multidimensional search space the derivative of the function is replaced by gradient and second-derivative by hessian.

The difficulty with this method of optimization is to evaluate the hessian matrix. Quasi-

Newton methods [14] attempt to overcome the problem, as this method does not involve the explicit calculation of hessian. These make use of the gradient vector of the previous step to get the nature of the value of hessian.

7. **Method of Lagrange Multipliers:** If we consider a search space $F(x)$ with equality constraint, say, $g(x) = 0$, it may bewritten in the form of Lagrange function [4]

$$L(x, \lambda) = F(x) + \lambda\, g(x) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (15)$$

Then one can follow the routine by setting derivative of the function equal to zero. For optimal solution

$$\nabla L(x, \lambda) = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (16)$$

More than one constraint-problem can be handled by adding Lagrange multipliers

$$L(x, \lambda) = F(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (17)$$

8. **Sequential Quadratic Programming:** Sequential quadratic programming (SQP) [4] is an iterative method for nonlinearoptimization.SQP methods are used on problems for which the objective function and the constraints both are continuously differentiable.

IV. STOCHASTIC SEARCH

The limitations of traditional deterministic optimizational techniques in finding global solution in a complicated search space drive the _eld of research to invent more and more efficient stochastic method-based optimization. Stochastic search algorithms are generally based on any physical or natural event or may follow the logic of a thermodynamical process. This includes all the method based onapproximate reasoning (fuzzy logic [15] and probabilistic model [16]), probabilistic search (evolutionary algorithm, Monte Carlo based algorithm [17, 18], swarm-based algorithm),functional approximation (neural network [19, 20]).

1. **Monte Carlo Based Algorithm:** Monte Carlo algorithm may be crudely defined as a statistical simulation by usinga sequence of random numbers. It is generally used in problems having some uncertainty in inputs and many coupled degrees of freedom. In this method a systemis considered to have a probability distribution function (PDF) and simulation may be performed by random sampling from the PDF. But all the system may nothave a well-defined PDF. Thus, Monte Carlo based optimization algorithms follow a model, that the system is sampled initially in a broad PDF and with iteration the parameters defining the PDF are evolved to reduce the width of PDF in order to locate the global optima. The random search, direct Monte Carlo sampling and the random hill climbing methods might be viewed as direct sampling methods where as Simulated Annealing, Quantum Annealing and Parallel Tempering fallunder the rubric of "random walks" as introduced by Metropolis [21].

2. **Random Search:** Random search [22, 23] may be the simplest one among all the stochastic algorithms. The algorithm is very easy to code, but has an appreciable effectivity in solving problems with reason able dimension, and also flexible to cast systems with continuous as well as discrete variables. Moreover, computational cost is manage able for this method. In the method the iterative modification of the variables go through a random event and the current vector is updated to minimize the objective function and the process goes on until the optimal solution is reached. According to the type of modifying the vectors Random Search method may be classified into two classes. In one class the vectors are randomly generated in each iteration, known as Blind random search, and another one is Localized random search, in which a local search is done around a guess solution to obtain a new vector.

3. **Random Mutation Hill Climbing (RMHC):**Random Mutation Hill Climbing [24,25] is similar to local random search but only involves mutation or modification of the current solution vector randomly, based on a user defined probability of mutation for each vector dimension.

4. **Simulated Annealing (SA):** Simulated Annealing [26,27] is a stochastic optimizations cheme which mimics the physical process of annealing to produce the best thermodynamic state of alloy. In thermodynamic annealing process, the molten mixture is prepared at a sufficiently high temperature (known as starting annealing temperature) and then gradually cooled to get the most stable state of alloy. SA follows exactly the same principle. The simulation starts at a high temperature, T_{at} (annealing temperature) and then the annealing temperature is slowly decreased with a specific rate, known as the annealing schedule. At high T_{at} more and more area of

the search space is sampled and gradually with the lowering of temperature the extent of area for sampling is decreased and finally becomes directed towards the optimum solution. During simulation, the parameter set obtain edine a chiteration is fedin to the calculation to get the objective function, popularly known as the cost function. The cost has to be minimized with simulation and for the optimum solution It must tend to zero.

During the course of optimization, if the surface is rugged, there is always a finite probability of being trapped in a local minimum. If one has to come out of these local attractive basins, temporarily one needs to accept moves in which the cost function might increase in magnitude, with the eventual goal being to reduce it to zero. This is implemented in SA by controlling the thermal fluctuation, which is induced by the annealing temperature T_{at}.The thermal fluctuation is used to cross the energy barrierseparating one minimum from the other.This is technically implemented by the so-called Metropolis Test. A quantity "Δ"is defined which is, $\Delta = \text{cost}_i - \text{cost}_{(i-1)}$. Here, $\text{cost}_i$is the value of the cost function for the present move and $\text{cost}_{(i-1)}$is the one for the previous move. If"Δ"is negative, the movies accepted straight away. If not, it is subjected to the Metro polis test. The probability of accepting a move in Metropolis test is

$$P_M = Exp\left[-\frac{\Delta}{kT_{at}}\right] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots. (18)$$

P_M can be between 0 and 1. A random number between 0 and 1 is invoked and if P_M isgreater than the called random number the move is accepted. At higher Tat, P_M will beclose to 1 and more moves will pass the Metropolis test. The physical meaning is thatat higher simulation temperature, due to strong thermal fluctuations, a greater length of search space is sampled and nearly all moves become accepted. As the simulation proceeds, T_{at} is gradually decreased by following some scheme, known as annealingschedule. At low Tat, lesser number of moves pass the Metropolis test and only those moves for which the cost function predominantly decreases are accepted and in the limit of $T_{at} \rightarrow 0$, the correct solution or the global minimum is found out.

5. **Parallel Tempering (PT):** In this optimization scheme, one starts the search with different temperatures simultaneously. The temperature of a particular zone remains constant during the simulation. The temperature zones are dispersed from higher to lower range. The simulation at high temperature would sample a large area of search space than the one at a lower temperature. As in SA the system makes many moves and the move with lower cost is accepted. If the value of the cost function of a move is higher than the previous one, the move is subjected to Metropolis Test. Maximum moves in the high temperature zones will be accepted, whereas the probability of passing the Metropolis Test gradually decreases with the decrease in temperature of the different zones. The unique feature of PT is that swapping of search locations among these temperature zones is allowed. The swapping probability [28] is

$$P_{swap} = \min[1, exp\{-\Delta\beta\Delta\text{cost}\}] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (19)$$

Where $\Delta\beta$ is the difference in the inverse of temperature of two zones and $\Delta\mathrm{cost}$is the difference in cost of the same two zones.

This feature allows for a more efficient sampling of the search space both in terms of computational time and the quality of the solution achieved.

6. **Tabu Search:** Tabu Search in its present form was given by Glover [29]. Generally, in other search techniques only the information about the best one, so far found, is stored, whereas in TS the course of finding the solutions are stored. Tabu Search extends hill climbing by the concept that it declares solution candidates which have already been visited as tabu. Hence, they must not be visited again and the optimization process is less likely to get stuck on a local optimum and this is the effectivity of the technique.

7. **Quantum Annealing:** Quantum Annealing [30, 31] employs the quantum fluctuation to anneal the system down to global minima. Unlike classical annealing, where the thermal fluctuation is used to cross barrier, tunnelling is the reason behind the fluctuation in Quantum An nealing. In the algorithm the cost function is represented by a classical Hamiltonian (H_0)and a suitable quantum annealing term is defined ($H(t)$). Then one may write

$$i\hbar\frac{\partial\psi}{\partial t} = [H_0 + H(t)]\psi \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (20)$$

The solution of this time-dependent Schrödinger Equation approximately describesthe tunnelling dynamics of the eigen states of H_0. Like thermal fluctuations in(classical) simulated annealing, the quantum (tunnelling) fluctuations owing to $H(t)$ helps the system to come out of the local minima. If $H(t) \to 0$ for $t \to \infty$,the system eventually settles in one of the eigenstates of H_0 ; hopefully the groundstate. The effectivity of this technique over simulated annealing (classical) lies onthe adiabaticity ofquantum evolution, i.e., it offers sufficiently slow annealing andthe possibility to get trapped in a local minimum may decrease.

8. **Evolutionary Algorithm:** The algorithm of stochastic optimization based on the evolution of species in nature probably acts the best of optimizers. The species evolve according to the Darwinian concept of \survival of fittest". Only those survive which are _t to interact with surrounding and reproduce. For evolution the physical processes occurring, are selection, mutation, competition and reproduction [32], and these operations incorporated in evolutionary optimization to evolve the trail solutions (initially chosen) to get the best solution or global solution.

   The three different kinds of evolutionary algorithms are [33-35]

   - Evolutionary Programming [36]
   - Evolutionary Strategies [37, 38]
   - Genetic Algorithm [39, 40]

9. **Genetic Algorithm:** Akin to genetics, GA [25, 39-46] uses analogues of selection, crossover, mutation operations to find the optimum solution. The simulation starts from a

pool of trial solutions (generated randomly) and not a single one as is the case with most search techniques. The individual solutions are popularly known as strings. How far are the individual strings from the optimal solution is described in terms of the fitness function. The calculation of fitness function are problem specific. Generally, GA follows the maximization of objective function or _t function.

The simulation is started with the solution pool containing strings of low fitness. These strings are iteratively subjected to the operations of selection, cross over and mutation, till a string is obtained with fitness value close to one (which is the desired solution). The operational details of the three operations are as follows:

- **Selection:** In this operation, some of the strings in the solution are selected for the next operation. The strings of relatively higher fitness have greater probability of being selected**.** There are various ways of doing the selection and we have used the “roulette wheel procedure" or the “elitism" procedure. Selection will discard strings of low fitness, create multiple copies of the ones with high fitness and there by increase the average fitness of the solution pool.

- Crossover: In conventional genetics, exchange of information occurs to produce fitter individuals and better genes. This phenomenon is called crossover. In GA crossover of data occurs among pairs of strings. There is a crossover probability, which is defined for a particular simulation process. This controls the number of strings which are picked up for crossover. Among the selected strings, pairs are arranged randomly and exchange of information takes place by partial swapping of the information contents in the strings. Hence this process introduces new variety and information in the solution pool, which was absent at the beginning.

- **Mutation:** Mutation is very rare event in conventional genetics and so is it in a GA optimization. In genetics, mutation means sudden change in genetic structure. We set the mutation operation in simulation as

$$S' = S + (-1)^{mk} \times r \times \Delta$$

Where r is a random number, mk is either 1 or 2 and Δ is maximum allowed change. Thus any element S in the solution pool has been chosen (randomly) and it becomes S' after mutation. Mutation is the source of introducing absolutely new information into the solution pool. Mutation is a beneficial operation, but a high mutation probability can offset the beneficial effects and so one must exercise some caution while using it.

Conventionally the solution pool in GA contains infinite number of strings. However in practice a GA simulation is carried out with only a finite number of strings to keep a balance between efficiency and efficacy.

V. DIFFERENT ARENA WHERE OPTIMIZATION IS USED BY CHEMISTS:

Application of optimization in various disciplines of chemical physics is un questionable. Usage of stochastic search algorithm or soft computing techniques (more general abbreviation) as a method of optimization expedites many area of research in chemical physics. Increasing demand for global optimization in this field motivates chemical physicists not only to apply these algorithms, but also to develop new soft computational algorithm or to make modification in existing algorithms according to perspective.

Literature is enriched with lots of applications of soft-computational algorithms in chemical physics. In this review we can allow us only for a very brief discussion about the applications of optimizations, rather stochastic optimization techniques in different avenues of chemical physics.

VI. GEOMETRY OF ATOMIC AND MOLECULAR CLUSTER

Stochastic optimizer had made easy access in obtaining proper geometrical shape of cluster. The clusters may be made of some kind of lattice points or atoms or even molecules. To optimize cluster structures one need to get the global minima on the potential energy surface as generally the structure of a cluster very close to the most minimum energy point of an appropriate potential energy surface. Literature is enriched with the example of getting cluster structure using different types of stochastic optimization algorithms, such that Basin Hopping Monte-Carlo, Genetic Algorithm,simulated annealing, Parallel Tempering. Now a days particle swarm optimization algorithms have also been practiced to evaluate structures of complicated systems. Staring from noble gas to metallic clusters, molecular clusters, ionic clusters have been optimized using stochastic search method [47-62]. The stochastic search algorithms can be clubbed with quantum chemistry package [63,64] which actually gives the field of structure optimization a leap.

VII. REACTION PATH

Optimization algorithm can be used to extract information about transition state and transformation path also. Only one need to design the objective function accordingly so that transition state or the whole path could be optimized. Here also the optimization variables are the co-ordinates of the atoms of the molecular system. Many reports are there where optimization algorithms have been used in different manner to get the transition state or the whole reaction path [65-67].

VIII. KINETIC PARAMETER ESTIMATION

Systems biology deals with the computational and mathematical modelling of complex biological systems. In doing so one has to successfully analyse the findings from high-throughput experiments. This leads to employ the inverse problem in system biology [68, 69]. The major use of inverse problem is in parameter estimation from experimental data sets. Optimization search algorithms can be used for this purpose. By taking the experimental finding as an objective one can optimize the kinetic parameter set [70-72].

IX. TO CONTROL QUANTUM PHENOMENA

Controlling quantum phenomena is one of the promising fields of research. Emerging improvement in laser (rather femtosecond laser) spectroscopy supplies the tool for doing quantum control. A properly designed laser pulse can actually manipulate a quantum phenomenon like selective control in photochemical reactions, tunnelling or barrier crossing dynamics. Now designing a proper laser field is a non-trivial job and stochastic optimizer can be applied in this occasion [73-77]. In such cases the optimization variables would be the pulse parameters, i.e intensity, frequency, pulse width etc and objective function could be designed according to the objective of the problems. Generally single-color pulse fails to achieve the objective, but polychromatic pulse could be designed by optimization algorithms which in most cases give good results.

X. EPILOGUE

There is vast scope of using optimization in chemistry. Some of them are only discussed. The algorithm could be used directly in designing experiments also. Pulse shaping could be merged with the experimental set-up to tune the results. Now there is a huge scope of using Artificial Intelligence (AI) in different aspects of chemistry which is one of the frontier topics of research in chemistry, especially computational chemistry.

REFERENCES

[1] L. T. Biegler Nonlinear Programming Concepts, Algorithms, and Applications to Chemical Processes, Mathematical Optimization Society and the Mathematical Optimization Society (2010).
[2] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiely (2001).
[3] T. Weise Global Optimization Algorithms Theory and Application (e-book), 2nd edition, (2009).
[4] T. Haukaas Deterministic Optimization Methods, www.inrisk.ubc.ca.
[5] J. Kiefer, Proceedings of the American Mathematical Society 4, 502 (1953).
[6] J. Kowalik and M. R. Osborne, Methods of Unconstrained Optimization Problems (Elsevier, New York, 1968).
[7] J. A. Nelder and R. A. Mead, Computer Journal, 7, 308313 (1965).
[8] D. M. Olsson, L. S. and Nelson, Technometrics 17, 4551 (1975).
[9] W. Spendley, G. R. Hext, and F. R. Himsworth, Technometrics 4, 441461 (1962).
[10] R. Fletcher, Practical Methods of Optimization, Wiley, Chichester (1981).
[11] G. Meurant, The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations, SIAM, Philadelphia, U. S. A. (2006).
[12] R. Pytlak, Conjugate Gradient Algorithms in Nonconvex Optimization; vol. 89, Pardalos, P. (Ed.), Springer-Verlag Berlin Heidelberg (2008).
[13] C. T. Kelley, Solving Nonlinear Equations with Newtons Method, vol. 1 of Fundamentals
[14] of Algorithms, Society for Industrial Mathematics, Philadelphia, PA (2003).
[15] L. M. Surhone, M. T Timpledon, and S. F. Marseken, (Eds.), Quasi-Newton Method: Maxima and Minima, Newtons Method in Optimization, Stationary Point, Hessian Matrix, Gradient, Positive-Definite Matrix, Betascript Publishing (2010).
[16] P. H_ajek, Metamathematics of Fuzzy Logic, (Trends in Logic, Springer, 2001)
[17] N. Limnios, D. C. Ionescu, Statistical and probabilistic models in reliability, (Statistics forIndustry and Technology, Birkh□auser Boston, 1998)
[18] G. S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, (Operations Research, Springer-Verlag, New York, 1996)
[19] C. P. Robert, Monte Carlo Statistical Methods, (2nd ed., Springer-Verlag,2004)
[20] S. Haykin, Neural Networks: A Comprehensive Foundation (Macmillan, New York, 1994)
[21] C. M. Bishop, Neural Networks for Pattern Recognition, (Oxford University Press, 1st ed., 1996)
[22] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, J. Chem. Phys. 21, 1087 (1953).
[23] D. C. Karnopp, Random Search Techniques for Optimization Problems, Automatica 1, 111-121 (1963).

[24] T. G. Kolda, R. M. Lewis, and V. Torczon, Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods, SIAM Review 45, 385-482 (2003)
[25] M. Mitchel, S. Forrest, S., Holland, J. H., When Will a Genetic Algorithm Outperform Hill Climbing, Cowen, J. D., Tesauro, G., Alspector, J. (Eds.), Advances in Neural Information Processing Systems, 6, Morgan Kaufmann, San Mateo, CA (1994).
[26] A. Mitchell, An Introduction to Genetic Algorithms, (1st ed., MIT Press, Cambridge, MA, 128-132, 1996).
[27] K. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi Science 220, 671 (1983).
[28] K. S. Kirkpatrick, J. Stat. Phys. 34, 975 (1984).
[29] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. 7, 3910 (2005).
[30] F. Glover, Computers and Operations Research 13, 533 (1986).
[31] T. Kadowaki, and H. Nishimori, Phys. Rev. E 58, 5355 (1998)
[32] J. Brook, D. Bitko, T. F. Rosenbaum, and G. Aeppli Science 284, 779 (1999)
[33] E. Mayr, The Growth of Biological Thought: Diversity, Evolution and Inheritance, (Belknap Press, Cambridge, MA, 1988)
[34] H. J. Bremermann, Optimization through evolution and recombination, (Self-OrganizingSystems, Yovits, M. C., et al., (Eds.), Washington, DC, Spartan, (1962)
[35] R. M. Friedberg, A learning machine: Part I, IBM J. 2, 213 (1958)
[36] R. M. Friedberg, A learning machine: Part II, IBM J. 3, 282287 (1959)
[37] L. J. Fogel, A. J. Owens, and M. J. Walsh, Artificial Intelligence through Simulated Evolution, (John Wiley, New York, 1966).
[38] I. Rechenberg, Evolutionsstrategie - OptimierungtechnischerSystemenachPrinzipien der biologischen Evolution., (Frommann-Holzboog, Stuttgart, 1973).
[39] H. P. Schwefel, Numerical optimization of computer models, (Wiley &Sons,Chichester, 1981).
[40] J. H. Holland, Adaptation in natural and artificial systems, (The University of Michigan Press, Ann Arbor, 1975).
[41] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, (Addison Wesley, Rading, MA, 1989).
[42] G. J. E. Rawlins, (Ed.), Foundations of Genetic Algorithms, (Morgan Kaufmann, 1991).
[43] L. D. Whitley, (Ed.), Foundations of Genetic Algorithms 2,(Morgan Kaufmann, 1993).
[44] L. Davis, (Ed.), Genetic Algorithms and Simulated Annealing, (Morgan Kaufmann Publishers, 1987).
[45] J. H. Holland and J. S. Reitman, Cognitive systems based on adaptive algorithms, (Pattern-Directed Inference Systems, D. A. Waterman and F. Hayes-Roth, (Eds.), Academic, New York, 1978).
[46] K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, (Ph.D. dissertation, Univ. of Michigan, Ann Arbor, 1975).
[47] K. A. De Jong, Are genetic algorithms function optimizers? 3 13, (Parallel Problem Solving from Nature 2, Elsevier, Amsterdam, The Netherlands, (1992).
[48] J. Bernasconi, J. Phys. 48, 559 (1987).
[49] J. P. K. Doye, Comput. Mater. Sci. 35 227 (2006).
[50] X. Shao, X. Liu, and W. Cai, J. Chem. Theor. Comput. 1 762 (2005).
[51] D.J. Wales and J.P.K. Doye, J. Phys. Chem. A 101 5111 (1997).
[52] W. J. Pullan, Comput. Phys. Commun 107 137 (1997); J. Chem. Inf. Comput.Sci. 37 1189 (1997).
[53] P. Chaudhury and S. Bhattacharyya, Chem. Phys. 241 313 (1999).
[54] E. Lee, D. Farelly, and K. B. Whaley, Phys. Rev. Lett. 83 3812 (1999).
[55] J. Zhao and R. H. Xie, J. of Comp. and Theo. NanoSc. 1 117 (2004).
[56] A. Stasikowski, M. Moneta, and T. W. Gwizdalla, Surface and Interface Analysis38 469 (2006).
[57] R. Poteau and G. M. Pastor, Euro. Phys. J. D 9 235 (1999).
[58] A. Rapallo, G. Rossi, R. Ferrando, A. Fortunelli, B. C. Curley, L. D. Lloyd,G. M. Tarbuck,and R. L. J. Ohnston, J. Chem.Phys. 122 19 (2005).
[59] S. K. Biring, R. Sharma, and P, Chaudhury, J. Math. Chem. 52 368 (2014).
[60] D. Wolf, O. V. Buyevskaya, and M. Baerns, App. Catalytic A 200 63 (2000).
[61] S. Ganguly Neogi and P. Chaudhury, J Comp. Chem. 35 51 (2014).
[62] D. Alfe, M. Alfredsson, J. Brodholt, M. J. Gillan, M. D. Towler, and R. J.Needs, Phys. Rev. B 72 014114 (2005).
[63] B. Hartke, App. of Evol. Comput. in Chem., (Springer-Verlag, Berlin, Heidelberg,11033-11053 (2004).
[64] P. Naskar, R. Roy, S. Talukder, P. Chaudhury, Mol. Phys, 116, 2172 (2018)
[65] P. Naskar, Mol. Phys. 117, 575 (2019)
[66] J. C. Grossman, W. A. Lester Jr, S. G. Louie, J. Am. Chem. Soc. 122 705(1999).

[67] P. Chaudhury, S. P. Bhattacharyya, Journal of Molecular Structure:THEOCHEM 429 175 (1998); Int. J. Quant. Chemi. 76 161 (2000).
[68] S. Talukder, S. Sen, R. Sharma, S. K. Banik andP. Chaudhury, Chem. Phys. 431-432 5 (2014).
[69] H. W. Engl, C. Flamm, P. Kügler, J. Lu, S. Müller, and P. Schuster, InverseProblems, 25, 123014 (2009).
[70] P. Kügler, Plos One 7 e43001 (2012).
[71] C. G. Moles, P. Mendes, and J. R. Banga, Genome Research 13 2467 (2003).
[72] A. G_abor and J. R. Banga, Computational Methods in Systems Biology,45-60 (2014).
[73] S. Da Ros, G. Colusso, T. A. Weschenfelder, L. de Marsillac Terra, F. deCastilhos, M. L. Corazza, and M. Schwaab, Applied Soft Computing 13 2205(2013).
[74] S. Sharma and H. Singh, Chem. Phys. 390 68 (2011).
[75] T. Klamroth and D. Krüner, J. Chem. Phys. 129 234701 (2008).
[76] S. Sen, S. Talukder, and P. Chaudhury, Indian J. Phys. 87 865 (2013).
[77] B. K. Shandilya, S. Sen, T. Sahoo, S. Talukder, P. Chaudhury, and S. Adhikari,J Chem. Phys.139 034310 (2013).
[78] S. Ghosh, S. Talukder, S. Sen, and P. Chaudhury, Chem. Phys. 425, 73(2013).