Review of the Application of Genetic Algorithm and Precision Points in Optimisation of the Four-bar Mechanism

Joseph K. Mwangi, Onesmus M. Muvengei, and Moses F. Oduori

Abstract—Four-bar linkages are the most popular mechanisms and several studies have been dedicated to improving them. One area that has attracted the attention of many researchers is kinematic synthesis. Kinematics synthesis is the reverse of kinematic analysis in that while the latter involves the determination of the motion in a mechanism, the former is concerned with the development of a mechanism to satisfy certain desired motion specifications, either displacement, velocity or acceleration or a combination of any of them. A specific task of kinematic synthesis is of interest: This is the dimensional synthesis which deals with the synthesis of linkage lengths. Over the years, two main techniques have been used for dimensional synthesis: Precision point synthesis and genetic algorithm optimisation. Precision point synthesis was the initial method in the optimisation of the four-bar mechanism designs. Recently, the use of evolutionary techniques such as genetic algorithm (GA) have made it easier to create efficient mechanisms courtesy of advancements in computer technology. The emergence of computational software such as MATLAB simplifies the use of GA. This paper reviews how precision points and genetic algorithms have been used to optimise the four-bar mechanism.

Keywords—dimensional synthesis,four-bar mechanism, genetic algorithm,optimisation, precision points

I. INTRODUCTION

THERE has been great research interest in the field of optimisation of the four-bar mechanism. Mechanism optimisation mostly involves a section of kinematic synthesis known as dimensional synthesis. Dimensional synthesis is the determination of mechanism elements such as the length of the members, angles and coordinates which are necessary for the creation of a mechanism with a desired motion. Researchers on dimensional synthesis have focused on two main areas, namely, the precision point synthesis and application of mathematical programming techniques to mechanism design [1],[2].

Precision point synthesis has excellent properties in motion geometry design optimisation. However, it has limitations in cases where the desired mechanism ought to satisfy certain constraints. In such cases, trial and error bases are used to repeatedly revise the design, a process which becomes very cumbersome. To solve such challenges comes the mathematical programming techniques. A combination of the two approaches is desirable for the best mechanism optimisation results [1]. One of the most reliable mathematical programming techniques is the genetic algorithm(GA).

Joseph K. Mwangi, Department of Mechanical Engineering, JKUAT (e-mail: joseph.mwangi@jkuat.ac.ke).

Onesmus M. Muvengei, Department of Mechanical Engineering, JKUAT (e-mail: mmuvengei@eng.jkuat.ac.ke).

Moses F. Oduori, Department of Mechanical and Manufacturing Engineering , UoN (e-mail: foduori@uonbi.ac.ke).

GA is part of an applied research process known as Evolutionary Computation used for evaluating optimum solutions [3]. As the name suggests, the process is analogous to Darwin's theory of natural selection where only the fittest survives. GA is a non-deterministic process that uses stochastic information to obtain global optimum values. Holland,(1962), [4] was the first to use the GA technique, though its use as an optimisation tool began in the late 1980s [2].

GA possesses exceptionally excellence in handling ill-behaved, discontinuous and non-differentiable algorithm functions. In its operations, unlike the biological genetic systems, the GA encompasses some randomness which guarantees better solutions rather than unfavourable ones. The most important advantages of the GA are its simplicity in implementation, fast convergence times and limited knowledge of solution space required to operate it [5].

II. PRECISION POINTS

Dimensional synthesis has two different but related segments called approximate and exact syntheses. Exact synthesis is only possible for some limited cases of "nice" functions [6]. In most synthesis procedures, the aim is to only approximate the desired solutions. The approximate solutions only match the actual solutions at selected points. These selected points are known as the precision or accuracy points [7].

The target link for the points is mainly the output link and in most cases is the coupler. Precision points are used with the assumption that the design will deviate slightly from the desired function in between the precision points and the deviation is within the acceptable region. The deviation is known as the structural error and is usually between 3-4% [8]. The aim of every mechanism design is to minimise the structural error. However, the error is dependent on the number and spacing of the precision points. The number of points chosen should be such that the structural error generated between the points is minimal. The synthesis of these points is controlled by the number of independent design variables that describe the mechanism [2].

The maximum number of accuracy points required is dependent on the type of dimensional synthesis. There are three forms of dimensional synthesis, namely function, path, motion generation[8],[9]. Fig.1 aids in understanding the equations and maximum precision points required in each of these synthesis processes.

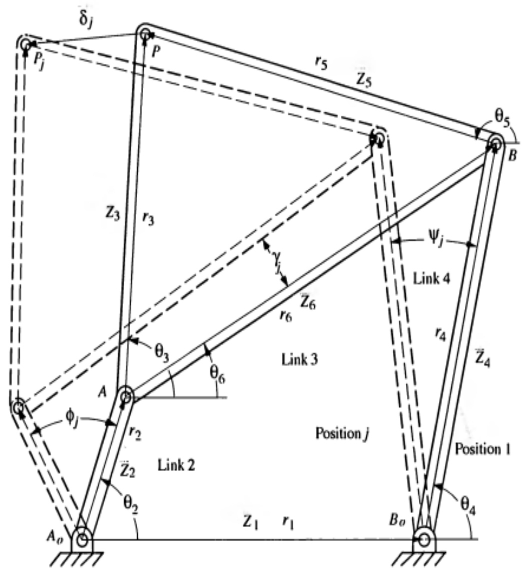


Fig. 1. Vector loop for a four-bar mechanism with a coupler point for two different positions, 1 and j [9]

The vector loop closure equation for position 1 is as shown in (1).

$$\vec{Z_2} + \vec{Z_3} - \vec{Z_5} - \vec{Z_4} - \vec{Z_1} = 0 \tag{1}$$

In position j, the loop is represented by (2).

$$\vec{Z_2}e^{i\phi_j} + \vec{Z_3}e^{i\gamma_j} - \vec{Z_5}e^{i\gamma_j} - \vec{Z_4}e^{i\psi_j} - \vec{Z_1} = 0 \tag{2}$$

where

ϕ_j, γ_j, ψ_j are the angular changes associated with links 2, 3, and 4 respectively.

First, function generation is the correlation of the sliding or rotary motion of the input and output links. For instance, to generate a function $y = f(x)$, the value of x could be related to the crank angle while the value of y could be the output angle at the precision points. From Fig. 1, only one vector is required to represent link 3 during function generation. Therefore, the vector equation for position 1 and j are as shown in (3) and (4) respectively.

$$\vec{Z_2} + \vec{Z_6} - \vec{Z_4} - \vec{Z_1} = 0 \tag{3}$$

$$\vec{Z_2}e^{i\phi_j} + \vec{Z_6}e^{i\gamma_j} - \vec{Z_4}e^{i\psi_j} - \vec{Z_1} = 0 \tag{4}$$

The overall equation for function generation is obtained by subtracting (3) from (4) as shown in (5).

$$\vec{Z_2}e^{i\phi_j - 1} + \vec{Z_6}e^{i\gamma_j - 1} - \vec{Z_4}e^{i\psi_j - 1} = 0 \tag{5}$$

By virtue of being a function generation process, the variables ϕ_j and ψ_j are prescribed. Therefore, the remaining unknowns are the direction and magnitude of vectors $\vec{Z_2}, \vec{Z_6}$, and $\vec{Z_4}$ as well as the angle γ_j. The number of unknowns for n number of precision points can, therefore, be written as $6 + (n - 1)$. If a vector loop equation was to be written for every precision point from point $j = 2$ to $j = n$, the number of scalar equations would be $2(n-1)$ since each vector equation gives rise to two scalar equations. If it happens that the number of unknown variables exceeds the number of scalar equations, there is need to assign values to some variables for deterministic evaluation of the function generation process. Such variables are called free variables or free choices. The number of free choices for function generation is determined from (6).

$$6 + (n - 1) - 2(n - 1) = 7 - n \tag{6}$$

Consequently, the maximum value of n or the maximum number of precision points for function generation is **seven** [9].

Second, path generation involves specifying the path traversed by a point on the output link. The equation for path generation can be obtained by subtracting (1) from (4) while taking note of the definition of the vector for link 3 which is given by (7) and thus obtaining (8).

$$\vec{Z_6} = \vec{Z_3} - \vec{Z_5} \tag{7}$$

$$\vec{Z_2}e^{i\phi_j - 1} + \vec{Z_3}e^{i\gamma_j - 1} - \vec{Z_5}e^{i\gamma_j - 1} - \vec{Z_4}e^{i\Psi_j - 1} = 0 \tag{8}$$

Equation (8) can be rearranged into vector pairs called dyads, as shown in (9) and (10).

$$\vec{Z_2}e^{i\phi_j - 1} + \vec{Z_3}e^{i\gamma_j - 1} = \vec{\delta_j} \tag{9}$$

$$\vec{Z_5}e^{i\gamma_j - 1} + \vec{Z_4}e^{i\Psi_j - 1} = \vec{\delta_j} \tag{10}$$

By virtue of being a path generation process, the displacement vector for point P, $\vec{\delta_j}$, is prescribed. The remaining unknowns are, $\vec{Z_2}, \vec{Z_3}, \vec{Z_4}, \vec{Z_5}, \phi_j, \gamma_j$, and ψ_j. Therefore, the total number of unknown variables for n precision points is $8 + 3(n - 1)$. Meanwhile, there are two vector equations each yielding two scalar equations for every value of n, thus the total number of scalar equations becomes $4(n - 1)$. Consequently, the number of free variables for path generation without prescribed timing is determined as in (11).

$$8 + 3(n - 1) - 4(n - 1) = 9 - n \tag{11}$$

Therefore, the maximum number of precision points required for path generation without prescribed timing is **nine** [9].

However, if there is a correlation between the position of point P and the input link position, the process is called path generation with prescribed timing. In this case, in addition to $\vec{\delta_j}$, either ϕ_j, or Ψ_j are prescribed. Therefore, there is one less unknown and the total number of unknown variables reduces to $8 + 2(n - 1)$. Since the number of scalar equations remain the same, the number of free choices available is given by (12).

$$8 + 2(n - 1) - 4(n - 1) = 10 - 2n \tag{12}$$

Consequently, the maximum number of precision points required for this case is **five**. Since the number of independent parameters has been reduced by prescribed timing, the synthesis problem requires less precision points and a maximum of five points is enough [9].

Third, motion generation involves guiding a rigid body to move in a specific motion. Usually, the body is attached or lies on the coupler of the mechanism. In this case, the free variables from (4) are: $\vec{Z}_2, \vec{Z}_3, \vec{Z}_4, \vec{Z}_5, \phi_j$, and ψ_j. Both γ_j and δ_j are prescribed. This is, therefore, similar to the path generation with prescribed timing where the total number of unknown variables for n precision points is $8 + 2(n - 1)$ whereas the total number of scalar equations is $4(n - 1)$. Similarly, (12) applies for the number of free variables and the maximum number of precision points obtained thereafter applies too, i.e. **five**. Despite their similarities in solution, motion generation differs from path generation in that while the former can be described as the motion of a line in a plane, the latter is the motion of a point in a plane [9].

The range of selection for the number of precision points is between two and the maximum number as defined by different forms of dimensional synthesis. If the maximum number of precision points is not used, free variables must be chosen to ensure deterministic solution to the problem. On the other hand, using more precision points than the maximum required leads to non-linearity, difficulties in finding solutions as well as non-feasible solutions, which for instance, contains complex numbers [2].

Typically, precision points at the extreme range of selection are not desirable as they result in very large errors at the middle of the range while having no errors at the extremes. The essence of precision point synthesis is not to match the desired output but to minimize the overall structural error. Nevertheless, some mechanisms demand for the precision of matching the desired output given by the extreme range of precision points. This is the case where a mechanism is to be used to open and close values as controlled by input link parameters. However, for other mechanisms, the best number of precision points is that which gives equalised structural error in the spaces between the precision points [10].

Apart from the number, spacing is also an important part in precision points synthesis. Several techniques have been proposed for determining the spacing of accuracy points. However, the most prominent and efficient technique is the Chebyshev's Spacing method [10]. According to the method, for n number of accuracy points in the range of $x_0 < x < x_f$, the most appropriate spacing for the precision points x_j is given by (13).

$$x_j = \frac{1}{2}(x_0 + x_f) - \frac{1}{2}(x_0 + x_f)cos\left(\frac{\pi(2j-1)}{2n}\right) \qquad (13)$$

The Chebyshev's spacing method, though efficient and convenient, does not result in optimum point spacing. Due to the assumptions made in deriving (13), it is not necessarily true that the method would result in equal minima and maxima of the structural error. The way to achieve optimum design is through trial and error, starting with the Chebyshev's spacing method and plot the resulting errors against the design variable. At points where maximum errors occur, the spacing between points is reduced and the process repeated until the maxima and minimal of structural error are equalized in the entire range of accuracy points [6].

A. Application of Precision point to Four-bar Mechanism Optimisation

Several authors have done research on precision points synthesis. Freudenstein [11], acknowledged as the father of modern kinematics, presented the synthesis of a four-bar linkage using two types of approximation methods. The first method was based on coinciding the ideal and actual function at several precision points. Approximation equations were developed for three different cases namely, three point, four point, and five point case. The author considered the five point case as the limiting order of approximate synthesis for complex functions using the algebraic method. However, Freudenstein insinuated on the possibility of higher order approximations for simple functions. In the second method of approximation, the author made the ideal and actual functions to coincide at only one precision point but also found out that several finite number of derivatives could be made to coincide at the same point up to the desired order. The limitation of this study is that the methodology used requires deep knowledge in algebraic mathematics. In addition, the algebraic method used has a limit of the number of precision points that can be used for synthesis.

McLarnan [12] sought to improve on the idea of Freudenstein. To do this, the author would minimise the structural error by using half as many precision points as used by Freudenstein thus saving on computation time and space. For a mechanism which allowed the synthesis of n points, McLarnan used fewer points by extending the idea of derivative approximation synthesis to not only one point but also to all other interior precision points. Despite the obvious improvements in utilising precision points, the methodology involved trial and error to minimise the structural error making the procedure cumbersome and time consuming. In addition, the analytical method used cannot allow for some of the synthesis tasks that demand the inclusion of many precision points.

Maclaine [13] used a numerical method to perform the Chebyshev optimum linkage design. The research involved an initial approximation with five precision points followed by the application of Newton's method to perform iterations that generated a crank-rocker linkage for solar declination. Th resultant structural error was highly minimised. The gap identified in this study is that it demands the use of an extra method to generate the initial approximation (of the structural error and derivatives) required for the Newton's method. This extra method could either be graphical(which is prone to human errors) or algebraic (which requires extensive algebraic knowledge). In addition, if these two methods did not give equal minimum structural error, the process had to be repeated, which made the exercise cumbersome and time-intensive.

Rose and Sandor [14] used a rather unconventional method for acquiring optimal dimensions of a four-bar mechanism via function generation by departing from the usual manual iteration method. Normally, the number of precision points is chosen by the designer while the spacing is obtained using the Chebyshev's theorem. After analysing the structural error, the points are then respaced through Freudenstein's respacing formula to minimise the errors with the process being repeated until the errors between the accuracy points are equal. However, this analytical study suggested that the synthesis could be done in one step by forcing the errors between the points to be equal and minimal by applying additional constraints. The minimised errors were obtained using the Newton-Raphson method implemented using a computer programming software. After an initial approximation, the method linearized the nonlinear equations using Taylor expansion method and solved for the error. The error was then fed to the initial approximation to get the second approximation. The computational process proceeded until the an equalised extrema was achieved. Nevertheless, this method is not foolproof as an equalised extrema does not necessarily mean minima of error.

Kinzel et al. [15] proposed the use of Geometric Constraint Programming (GCP) in kinematic synthesis for finitely separated precision points. The GCP method used the sketching window of commercial Computer Aided Design (CAD) software to parametrically develop kinematic diagrams. Numerical solvers were also integrated into the software to solve the synthesis equations thus combining the graphical and analytical techniques for better synthesis results and experience. To show the robustness and versatility of the GCP method, the authors used it to demonstrate motion generation for five precision points, path generation for nine precision points and function generation for four precision points. However, GCP has the limitation that only one solution is obtained; the solution nearest to the one sketched by the designer. This could be a local solution and thus does little in searching for a global solution [16].

Modern precision point synthesis problems are making use of advanced computational resources available. Martinez-Alfaro [17] studied the problem of four mechanism synthesis using Simulated Annealing method. This method enabled the author to transform the synthesis problem into an optimisation one and in so doing permitting the use of more precision points than the classical methods (analytical and graphical). In addition, the method also allowed the inclusion of more constraints for the synthesis of a more practical mechanism. The results obtained showed a good control of the trajectory of the mechanism. It is therefore a continuous path generation method instead of a point-to-point path generation process.

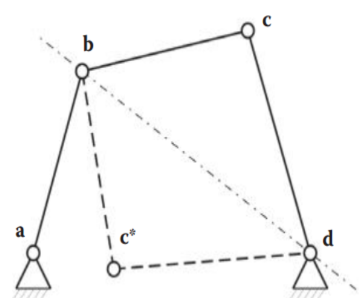
Mehar et al. [18] optimised the dimensions of a four-bar mechanism using five precision points by employing the Freudenstein's equation shown in (14).

$$K_1 cos(\varphi_j) - K_2 cos(\Psi_j) + K_3 = cos(\varphi_j - \Psi_j) \qquad (14)$$

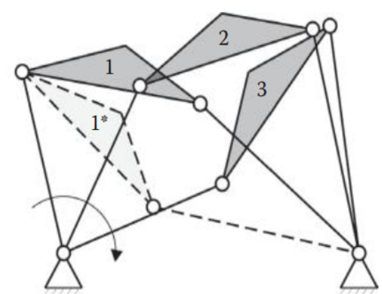
where $K_1 = \frac{a}{d}$; $K_2 = \frac{a}{b}$; $K_3 = \frac{a^2+b^2-c^2+d^2}{2bd}$, φ_j = input angle at point j; Ψ_j = output angle at point j; j = number of precision points; a, b, c, d = frame, crank, coupler, and rocker link lengths respectively. The Least Square Method was used to minimise the structural error with the results being compared to those obtained using four accuracy points. The authors results were more accurate compared to similar results from literature using four precision points showing that the Least Square Method proved is an effective method of minimising structural errors.

B. Limitations of the Application of Precision Points in Four-bar Optimisation

One of the major problems encountered with the application of precision points is branching. Branching occurs when it is physically impossible to have the coupler pass though all the intended precision points without having to be disassemble and reassemble the mechanism. This could happen even though synthesis results show that the coupler is capable of satisfying all the precision points[19],[20]. This occurrence is due to having more than one possible values of the position of a coupler corresponding to a given crank position. The number of mechanism branches is equal to the number of possibilities. In the case of all types of Grashof linkages, there are always two disconnected branches. Fig.2a shows two possible configurations of a four-bar mechanism. From Fig. 2a, a-b-c-d achieves the positions 1-2-3 in Fig. 2b while a-b-c*-d achieves position 1*. Since the mechanism cannot achieve all these positions continuously without having to be disassembled and reassembled, it is said to have a branch defect.



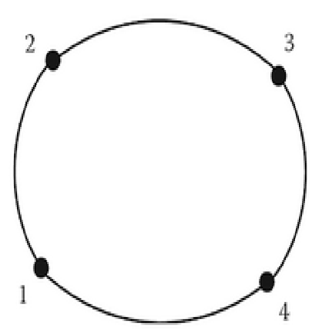
(a) two possible configurations of a four bar mechanism



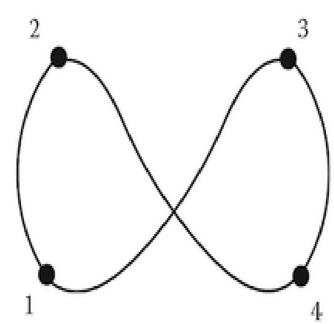
(b) branch defect

Fig. 2. Branching in four-bar mechanism[20]

The second problem in precision points is the order defect. During synthesis, it is normally desirable that the mechanism passes through the precision points in the prescribed order. Although this could happen, the physical outcome of the mechanism might not pass through the points in the specified order [21]. This is illustrated in Fig. 3a and Fig. 3b.



(a) point order 1-2-3-4-1



(b) point order 1-2-4-3-1

Fig. 3. Different path for the same precision points[20]

In graphical synthesis, the branching and order defects can be eliminated using the methods of Filemon's construction [22] and Waldron's construction [23]. However, in analytical synthesis, additional constraint equations as well as prescribed timing help in achieving defect-free mechanisms. Gupta [24] also introduced another method of separating the mechanism branches by partially differentiating the generator function with respect to the output(ϕ).

III. Genetic Algorithms

Genetic Algorithm, or simply GA, as defined by Holland [25], is a search technique that involves proceeding from an initial population of chromosomes (design variables) to a new population via the natural selection and other genetic operators such as crossover, mutation, and inversion. In the recent past, GA has come in handy to solve optimisation problems in the field of engineering giving rise to fresh research ideas as well as applications. This is because most computational problems in this field involve a search through vast amount of possible solutions. In this case, the GA technique offers great versatility by providing a parallel search for solutions, which makes it easier and faster to arrive at the most efficient solution. In addition, GA offers an intelligent approach to choosing the next population to consider for subsequent iterations, which is often desirable [26]. Michalewicz [27] summarised the five fundamental components of GA as follows:

1) A genetic model of the potential solutions to the problem.
2) Generation of an initial population
3) A fitness function to rate the solutions
4) Genetic operators(e.g. mutation operator and crossover operator) to change the offspring composition during reproduction.
5) Values for the GA parameters such as population size, probability of mutation, and probability of crossover.

A. *How GA works*

It is important to understand how the GA works. The flow chart in Fig. 4 is a summary of the stages of implementation of the GA.

The GA works to optimise a given function $f(x)$ which must be defined beforehand [27]. This is called the objective function. The next stage is the encoding stage where the design variables are represented genetically using an encoding system. For instance, if a binary bit encoding system is used, a single design variable is represented by a gene g_i made up of l_i bits [26]. The number of bits representing the design variable is dependent upon the precision required. A collection of the genes then form a chromosome.

Afterwards, it is necessary to establish the initial population. An initial population is composed of a number of chromosomes that form the first search space. Typically, this is done randomly to avoid biased solutions that tend to fall around particular areas of the search space.

Next stage is the evaluation stage where each chromosome is evaluated according to the fitness function and the current population. The chromosome with the best fitness function value is then stored by the logarithm and becomes a candidate for selecting into the future population.

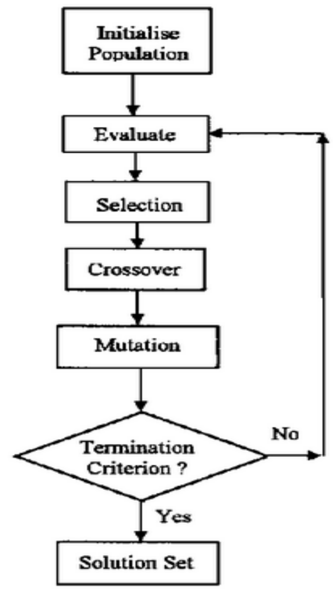


Fig. 4. Stages of the GA process [28]

Taking a step further is a selection process whereby, the current population saved after evaluation becomes the future population. In most cases, the selection process is done through a roulette wheel. The selected chromosomes then become parents for the next generation [28].

Next is the crossover genetic operation in which parts of the parent chromosomes on the different positions of the chromosomes are interchanged to form a different chromosome for the offspring. Crossover can either be single-point, double-point or multi-point. In single-point, there is one crossover point for the parents's genes whereas in two-point crossover, there are two crossover points and so on [29]. It should be noted that crossover occurs for the probability, Crossover Probability, CP $\epsilon[0, 1]$ [5].

There is also the mutation genetic operation which is achieved by random creation of a potential value to replace the old value in the chromosome. It occurs when the operator randomly selects a value in the region of real values $(x_i, x_i \pm range)$ which is then subtracted or added to x_i depending on the direction of mutation. Like crossover, mutation has a probability, mutation probability, (MP) between 0 and 1 but is always less than CP [5]. Both crossover and mutation operations are the two main stages in genetic algorithm.

GA operations can be summed in two phrases: search for the best solution and exploration of the solution space. The genetic operators conduct blind searches and should be directed by the selection operators into feasible areas of the search space [30]. Consequently, all GA elements should be carefully studied and additional heuristics added to improve GA performance.

B. Application of Genetic Algorithm to Four-bar Mechanism Optimisation

Kunjur and Krishnamurty [28] developed a genetic algorithm approach into the synthesis of a crank-rocker mechanism. The synthesis in question was the coupler path generation with prescribed timing. Real number encoding was used to represent the population values. The objective function was an expression of minimization of the structural error which was calculated as a sum of the square of errors at each target point. Constraints used were the Grashof's condition for a crank-rocker and link lengths. Two cases were considered, namely five target point case and an eighteen target point case. From the results, the author's concluded that real number coding achieved better global optima at even a faster rate compared to binary representation. In addition, the eighteen point case yielded more accurate results but required more computational costs in terms of both processor and time. The results were validated against those obtained from two gradient based methods. It emerged that the results were inferior to the exact gradient method results but superior to those of the central difference gradient method. However, the author pointed out that the exact gradient method had its own disadvantages which affect its application in complex mechanism optimisation problems. Nevertheless, this study had some limitations. First, crank angles were considered up to $90°$ for the five point case and up to $140°$ for the eighteen point case. Consequently, the entire motion path generation is not complete since one revolution of the crank must be $360°$. Second, the target points were created at the user's discretion and were not optimally synthesised as required by the Chebyshev's spacing method or any other optimal space generation technique.

Cabrera et al. [5] also studied genetic algorithm as a solution method for optimal synthesis of planar mechanisms. The method was comparable to that of Kunjur and Krishnamurty [28] albeit with modifications. First, more constraints were included such as the input angle variations and sequence of input angles. Furthermore, the constraints were incorporated into the objective function as penalty functions. Population size and probabilities of the genetic operators were also among the modifications. A disturbing vector was also added to guide the selection operator. Path generation with and without prescribed timing was considered for aligned and misaligned target points. From the study, the authors concluded that GA required more iterations than gradient based methods though there is the benefit of simple function evaluation. The inclusion of additional constraints in this study bore more accurate results in comparison with [28]. However, the methodology in this study could be improved by employing a better technique of creating the precision points instead of just the aligned and misaligned criteria. In addition, some cases considered do not account for the whole range of the input angle.

Dulger et al. [31] presented optimisation of four-bar mechanism using GA in Matlab software(Optimisation Toolbox) and compared the results to those obtained using $fmincon$ in the same software. The type of synthesis was path generation without timing and it utilised six target points arranged in a straight vertical line. After carrying out kinematic analysis of the mechanism, the objective function established was similar to that of Carbrera et al. [5]. According to the results, the use Optimisation Toolbox enhanced faster convergence of the algorithm. For every design variable, the $fmincon$ command yielded only one result while the GA yielded a result for every precision point used. This meant that GA was superior and provided enough data for error analysis. In addition, the Toolbox was easy to use and did not require vast knowledge to perform the optimisation. However, very few mechanisms, if any are known to trace a vertical line of coupler motion and therefore, the results of this study can barely be compared to real-life mechanism application. Furthermore, the study lacked a proper technique to select precision points(number and position) for practical application of the four-bar mechanism synthesis results.

Laribi et al. [32] used a different technique for path generation in a four-bar mechanism. The genetic algorithm approach was augmented with the fuzzy logic method controller whose work was to monitor the variable evolution in the first run and modify the initially set boundaries in readiness for the second run. This adjustment yielded better results in subsequent runs. The flowchart of the Genetic Algorithm-Fuzzy Logic(GA-FL) optimisation was as shown in Fig. 5. The symbols in the flowchart were defined as follows: G = generation; $G_(max)$= maximum number of generations; E_s = structural error; ϵ = tolerance.

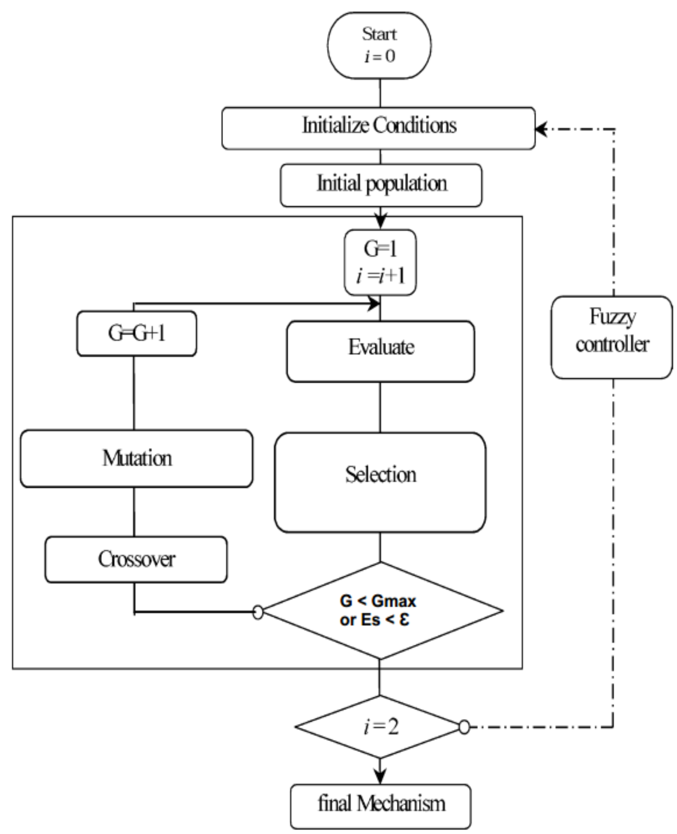


Fig. 5. Optimisation using a GA-FL [32]

Because of the controller, the GA was applied twice in the optimisation cycle. However, the number of function evaluations were comparable to pure GA methods since convergence was achieved faster. Using this approach, it was possible to reduce the population size required to about half the number in literature without affecting the accuracy of the results. The authors suggested the GA-FL approach was suitable for large domains where GA gave less accuracy compared to deterministic methods. Nevertheless, this method has its shortcomings especially in practical application of the four-bar mechanism whereby manipulating the domain of the design variables could result in a mechanism which does not fit in its assigned location.

Starosta [33] applied GA to synthesis a four-bar linkage as a closed curve generator with the curve being represented by normalised Fourier coefficients (NFCs). By normalising the curve, the number of curve parameters being investigated were minimised resulting in faster convergence of the GA.The objective function was the normal distance between the NFCs and the generated curve. After the synthesis, geometric transformation procedures namely rotation, scaling and reflection were done to determine the configuration of the four-bar linkage. Though the methodology in this paper achieved accurate results, it required the additional procedure of performing transformations after the GA work is complete.

Lin [34] studied path synthesis of the four-bar linkage using a hybrid methodology consisting of GA and Differential Evolution (DE). Though the procedure was majorly similar to that of GA, the aspect of DE was introduced in the crossover stage whereby a differential vector pertubation replaced the normal crossover operation. Consequently, the differential vector became the main perturbation while the mutation operation served as the minor perturbation. As in most other studies, the objective function was the sum of the square of the Euclidean distances between the desired and actual points. This study also established the stopping criteria as the mean error in Euclidean distance. Nevertheless, the author admitted that the proposed hybrid method could not be judged to be better than other evolutionary algorithms and proposed for proper combination amongst evolutionary algorithms for better optimisation potential.

Nariman-Zadeh et al. [35] presented a multi-objective GA application for Pareto optimum four-bar linkage synthesis. The multi-objective aspects came from the attempt to simultaneously minimize two objective functions, namely the tracking error (TE) and the deviation of the transmission angle from 90° (TA). The authors noticed that most synthesis problems were single objective problems based on the tracking errors between actual and desired target points. This was opposed to the real-life optimisation problem which the authors described as multi-objective thus the addition of the TA objective in the study. Pareto fronts showed the existence of two four-bar mechanisms which were a trade-off between the two conflicting objectives under consideration. Nevertheless, even with the benefits of a compromised solution, the problem of definition of transmission angle could render the algorithm inapplicable in some types of four-bar mechanisms whose efficiency is not suitably measured using the transmission angle. This includes such mechanisms as the jaw crusher whose output link is the coupler and not the rocker. Therefore, another objective should replace the TA objective if such mechanisms are to be optimised using the proposed method.

In the work of Acharyya and Mandal [36], the effectiveness of GA in four-bar linkage synthesis was compared to that of two other evolutionary algorithms, namely Particle Swarm Optimisation (PSO) and DE. To do this, the study used the same data and objective as Cabrera et al. [5] but included an additional refinement for selecting the initial population. When the results for the different cases considered were analysed, the DE technique emerged the most accurate followed by the PSO technique. GA was the least accurate with the additional disadvantage of longest convergence time. However, the authors did not clarify why the GA results obtained in the study compared poorly to those obtained by Cabrera et al. [5]. This could be proof that there were some inaccuracies in the preparation of the GA technique in this study.

Chaudhary and Chaudhary [37] utilised GA for two types of optimisation: dynamic balancing and shape optimisation. In dynamic optimisation, the problem involved reducing both shaking forces and shaking moments by applying GA and converting the links to equimomental point-mass systems. Using GA in Matlab, a reduction of 68% and 50% was achieved for shaking moment and shaking force respectively. After dynamic balancing, shape optimisation was done. The link shapes were modelled as cubic B-spline curves. The boundary of the curves were then optimised by considering minimum error in the inertia of the balanced links. Shape optimisation was then done using GA in Matlab and optimum link shapes obtained. However, the results obtained from the study could be erroneous since the problems were not solved to full dimensions as some assumptions were made to truncate the problem size. In addition, the weighted factors assigned to shaking forces and shaking moments were arbitrary hence the probability of error propagation in the optimisation process.

Shinde at al. [38] presented a dimensional synthesis study using GA for a six-bar mechanism which comprised of an input link, fixed link, triangular coupler and output link. The study was an improvement of precision point mechanism synthesis via the incorporation of GA for optimisation. The authors considered two cases of path generation with prescribed timing whereby one case involved an open curve generation with five points while the other involved a closed curve generation with eighteen points. In addition, the study dealt with straight line path generation without prescribed timing with six precision points. The objective function was the structural error calculated as the square of the Euclidean distances between the actual and desired points. After formulation, the objective function was coded into MATLAB and used to solve the three cases. From the resultant plots obtained, the study was successful in attaining paths close enough to those projected. Nevertheless, the task was void of additional constraints that would help to prevent the branching defect. In addition, the errors in the study, though small, could have been further reduced by initially determining the optimum number of precision points required for the different cases.

C. Advantages of the use Genetic Algorithm over Conventional Optimisation techniques in Optimisation Four-bar

Unlike conventional methods which deal with a single sample, the GA searches amongst a population of samples. This reduces the probability of locating false peaks which correspond to false solutions [39]. Therefore, four-bar synthesis results obtained through GA are like to be more accurate.

Moreover, GA uses information about the objective function itself rather than derivative or auxiliary information. Most sequential search methods, for instance, the gradient technique, demands function derivatives in order to carry out search procedures. The fact that GA requires little knowledge of problem structure and parameters is desirable and makes the algorithm more flexible to deal with non-linear, non-differentiable and noisy functions [39]. Some of the constraints equations and variable relationship in four-bar mechanisms are non-linear thus making GA a good candidate technique for synthesis.

Unlike traditional methods which are mostly deterministic, GA possesses stochastic operators that use probabilistic transition rules in the search operation. The organised random search is an important aspect that is aimed at guiding the search into regions of the solution space with better solutions. This makes GA more robust and effective [40].

In addition, while conventional methods act directly on the values of the population, the GA uses a coding system of bit strings. This enables GA to easily handle a variety of design variables [41].

Finally, GA applies multiple operators to manipulate the entire population at each generation thus making the search global and multi-directional. The search space is therefore well exploited and the possibility of being trapped in local optima is low [41]. Moreover, the fact that GA preserves a population of potential solutions reduces the risk of local convergence and promoting the evaluation of global optimum which is good for multi-modal optimisation problems [42].

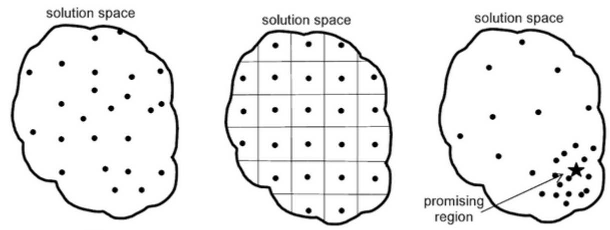


Fig. 6. Population Initialisation approaches [41]

D. Limitations of GA

Problems associated with GA could be due to the random sampling in population initialisation. When the population size is small, randomisation does not result in satisfactory sampling of the search space as shown in Fig.6a. Preferably, the solution space could be subdivided into several zones and each of the zones thereafter represented in the initial population so as to ensure uniform distribution of the population as shown in Fig 6b. Another approach is to use prior knowledge to bias the solution space towards a region rich in potential optimum solutions as shown in Fig 6c. This may be helpful in getting to the global optima faster as well as getting high quality results without having to increase the population size [27],[41].

Another limitation of GA is infeasibility and illegality. It should be understood that GA search operations involve alternating between the coding space and the solution space. The coding space is the genotype space while the solution space is the phenotype space. Usually, the GA operators function in the genotype space while the evaluation and selection occur in the phenotype space [30]. The transformation from the genotype to the phenotype is an important feature in GA. Sometimes, the transformation could yield results which are infeasible to the optimisation problem especially in cases of constrained problems and multi-modal optimisation [30]. Infeasible solutions are those that are within the solution space but are outside the expected feasible region. In other cases, illegal solutions could result from the GA. Illegal solutions are those that when decoded do not even fall within the solution space. This includes results consisting of complex numbers. Practically, the lengths of the four-bar links cannot be complex numbers meaning that such results would be useless to a designer. These two scenarios are illustrated in Fig. 7.

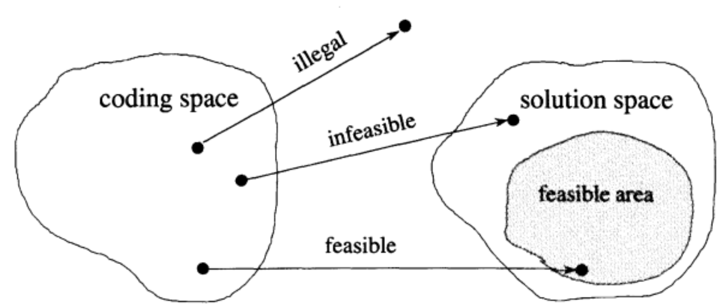


Fig. 7. Infeasible and illegal solutions in GA [30]

Penalty functions are employed to solve the problem of infeasibility. On the other hand, repair techniques are used to convert illegal solutions to legal ones [30].

GA also suffers from wrong choices of genetic operator parameters with the consequence of non-convergence and production of useless results. It is important to select an appropriate population size, rate of mutation and crossover as well as formulation of the fitness function [43].

IV. Conclusion

It is clear that over the years, both GA and precision points have overseen the synthesis four-bar for different practical application and research. A good combination of the legendary precision point theory and evolutionary theory has shown its ability to produce very accurate results. Nevertheless, there is room for improvement into the combination to achieve even better accuracy especially with the high precessor capacities of today's computational resources. Whatever was previously viewed as impossible using the graphical and analytical methods can now be done with outmost precision. GA continues to be an important tool in modern mechanism optimisation. Since its discovery, researchers have found it handy in dimensional synthesis of different forms of the four-bar mechanism. However, its usage demands selection of certain prerequisites which cannot be ignored if the accuracy of the results obtained is to be guaranteed. Meanwhile, optimal selection of precision points is crucial in all instances. Necessary modifications to the objective functions should be considered to account for different constraints that govern the motion of the mechanisms. Furthermore, the choice of genetic operator values cannot be underrated since it also has a great effect on the accuracy of the results. Moving forward, it is high time that the combination of GA and precision points is implemented to improve real-life mechanism applications to ensure best approximations to desired output. The benefits of both techniques should be exhaustively exploited while remedies should be sought for the shortcomings that befall them. With that, the future of four-bar mechanism optimisation will be brighter than it has ever been.

From the literature reviewed, the following gaps were identified:

1) Most researchers use arbitrary number and spacing of precision points. The results obtained, thus, cannot be compared to practical mechanism designs.
2) Very few researches have been done on multi-objective four-bar mechanism optimisation.
3) Very little information is available on post-optimisation analysis of the mechanisms to evaluate their viability in practical usage.
4) Most researches only consider a section of the whole range of crank angles thus only generating a section of the coupler motion
5) The limitations of GA have not been well tackled with a view to finding remedy.
6) Most of the research is has not been implemented into practical mechanisms to evaluate the effectiveness in real-life application.

V. Abbreviations and Acronyms

CAD	Computer Aided Design
CP	Crossover Probability
DE	Differential Evolution
GA	Genetic Algorithm
GA-FL	Genetic Algorithm-Fuzzy Logic
GCP	Geometric Constraint Programming
MP	Mutation Probability
NFC	Normalised Fourier Coefficients
PSO	Particle Swarm Optimisation
TA	Transmission Angle
TE	Tracking Error

References

[1] R. L. Fox and K. C. Gupta, "Optimization Technology as Applied to Mechanism Design," *Journal of Engineering for Indsutry*, no. 72, pp. 657–663, 1973.
[2] R. Bulatovic and S. Djordjevic, "Optimal synthesis of a four-bar linkage by method of controlled deviation," *Theoretical and Applied Mechanics*, vol. 31, no. 3-4, pp. 265–280, 2004.
[3] P. Venkataraman, *Applied Optimization with MATLAB Programming*. New York: John Wiley & Sons, 1st ed., 2002.
[4] J. H. Holland, "Outline for a Logical Theory of Adaptive Systems," *Journal of the ACM*, vol. 9, no. 3, pp. 297–314, 1962.
[5] J. Cabrera, A. Simon, and M. Prado, "Optimal synthesis of mechanisms with genetic algorithms," *Mechanism and Machine Theory*, vol. 37, no. 10, pp. 1165–1177, 2002.
[6] R. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*. New York: McGraw-Hill, 1st ed., 1964.
[7] G. P. Roston and R. H. Sturges, "Genetic algorithm synthesis of four-bar mechanisms," *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, vol. 10, no. 05, pp. 371–407, 1996.
[8] S. S. Rattan, *Theory of Machines*. New Delhi: Tata McGraw Hill Education Private Limited, 3rd ed., 2009.
[9] L. L. Howell, *Compliant Mechanisms*. New York: John Wiley & Sons, 2001.
[10] J. S. Rao and R. V. Dukkipati, *Mechanism and Machine Theory [Textbook]*. New Delhi: New Age International Publishers, 2nd ed., 2006.
[11] F. Freudenstein, "Approximate Synthesis of Four-Bar Linkages *[1 , 2]," *Transactions of the ASME*, vol. 77, no. Aug, pp. 853–861, 1955.
[12] C. W. McLarnan, "On linkage synthesis with minimum error," *Journal of Mechanisms*, vol. 3, no. 2, pp. 101–105, 1968.
[13] I. L. Maclaine-cross, "A numerical method of Chebyshev optimum linkage design," *Journal of Mechanisms*, vol. 4, no. 1, pp. 31–41, 1969.
[14] R. S. Rose and G. N. Sandor, "Direct Analytic Synthesis of Four-Bar Function Generators With Optimal," *Journal of Engineering for Indsutry*, vol. 8, no. 72, 1973.
[15] E. C. Kinzel, J. P. Schmiedeler, and G. R. Pennock, "Kinematic Synthesis for Finitely Separated Positions Using Geometric Constraint Programming," *Journal of Mechanical Design*, vol. 128, no. 5, p. 1070, 2006.
[16] Y. Tong, D. H. Myszka, and A. P. Murray, *Four-Bar Linkage Synthesis for a Combination of Motion and Path-Point Generation*. Masters thesis, University of Dayton, 2013.
[17] H. Martínez-alfaro, "Four-bar Mechanism Synthesis for n Desired Path Points Using Simulated Annealing," in *Advances in Metaheuristics for Hard Optimization*, pp. 23–37, Berlin: Springer-Verlag Berlin Heidelberg, 1st ed., 2008.
[18] K. Mehar, S. Singh, and R. Mehar, "Optimal synthesis of four-bar mechanism for function generation with five accuracy points," *Inverse Problems in Science and Engineering*, vol. 23, no. 7, pp. 1222–1236, 2015.
[19] S. Krishnamurty, *Multiple Objective Optimization Techniques in the Synthesis and Design of Mechanisms*. Madison: University of Wisconsin, 1st ed., 1989.
[20] K. Rusell, Q. Shen, and R. S. Sodhi, *Mechanism Design:Visual and Programmable Approaches*. Boca Raton: CRC Press, 1st ed., 2010.
[21] C.-T. Ho, *Precision Position Synthesis for General Planar Mechanisms by Predictor Corrector Continuation Method*. Minnesota: University of Minnesota, 1st ed., 2001.
[22] E. Filemon, "Useful ranges of centerpoint curves for design of crank-and-rocker linkages," *Mechanism and Machine Theory*, vol. 7, no. 1, pp. 47–53, 1972.
[23] K. J. Waldron, "Elimination of the Branch Problem in Graphical Burmester Mechanism Synthesis for Four Finitely Separated Position," *ASME Journal of Engineering for Industry*, vol. 98, no. 1, pp. 176–182, 1976.
[24] K. C. Gupta and S. O. Tinubu, "Synthesis of Bimodal Function Generating Mechanisms Without Branch Defect," *Transaction of the ASME, Journal of Mechanisms, Transmissions and Automation in Design*, vol. 105, no. 82, pp. 641–647, 1983.
[25] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Massachusetts: MIT Press, 2nd ed., 1992.
[26] M. Mitchell, *An introduction to genetic algorithms*. Massachusetts: The MIT Press, 1st ed., 1996.
[27] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Charlotte: Springer Science & Business Media, 3rd ed., 1996.

[28] A. Kunjur and S. Krishnamurty, "Genetic Algorithms in Mechanical Synthesis," *Journal of Applied Mechanisms and Robotics*, vol. 4, no. 2, pp. 18–24, 1997.

[29] V. M. Kureichik, S. P. Malioukov, V. V. Kureichik, A. S. Malioukov, V. I. Dempsey, and M. O. Neill, *Genetic Algorithms for Applied CAD Problems*. Berlin: Springer-Verlag Berlin Heidelberg, 1st ed., 2009.

[30] M. Gen and R. Cheng, *Genetic Algorithms & Engineering Optimization*. Ashikaga: John Wiley & Sons, 1st ed., 2000.

[31] L. Dulger, H. Erdogan, and M. Kutuk, "Matlab's GA and Optimization Toolbox: A Fourbar Mechanism Application," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 2, no. 1, pp. 10–15, 2014.

[32] M. A. Laribi, A. Mlika, L. Romdhane, and S. Zeghloul, "A combined genetic algorithm-fuzzy logic method (GA-FL) in mechanisms synthesis," *Mechanism and Machine Theory*, vol. 39, no. 7, pp. 717–735, 2004.

[33] R. Starosta, "APPLICATION OF GENETIC ALGORITHM AND FOURIER COEFFICIENTS (GA-FC) IN MECHANISM SYNTHESIS Roman Starosta," *Theoretical And Applied Mechanics*, vol. 46, no. 2, pp. 395–411, 2008.

[34] W. Y. Lin, "A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage," *Mechanism and Machine Theory*, vol. 45, no. 8, pp. 1096–1107, 2010.

[35] N. Nariman-Zadeh, M. Felezi, A. Jamali, and M. Ganji, "Pareto optimal synthesis of four-bar mechanisms for path generation," *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 180–191, 2009.

[36] S. K. Acharyya and M. Mandal, "Performance of EAs for four-bar linkage synthesis," *Mechanism and Machine Theory*, vol. 44, no. 9, pp. 1784–1794, 2009.

[37] K. Chaudhary and H. Chaudhary, "Shape Optimization of Dynamically Balanced Planar Four-bar Mechanism," *Procedia Computer Science*, vol. 57, no. 2015, pp. 519–526, 2015.

[38] A. S. Shinde, S. A. Kulkarni, and S. S. Shete, "Dimensional Synthesis of Four Bar Mechanism Using Genetic Algorithm," *International Journal of Current Engineering and Technology*, vol. 7, no. 4, pp. 1572–1580, 2017.

[39] S. Fidanova, *Recent Advances in Computational Optimization*. Cham: Springer International Publishing Switzerland, 2013.

[40] F. Erdogdu, *Optimization in Food Engineering*. New York: Taylor & Francis Group, 2008.

[41] P. Rangaiah, *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*. London: World Scientific Publishing Co. Pte. Ltd, 1st ed., 2010.

[42] D. Reid, "Genetic algorithms in constrained optimization," *Mathematical and Computer Modelling*, vol. 23, no. 5, pp. 87–111, 1996.

[43] X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*. Hoboken: John Wiley & Sons, 1st ed., 2010.