Methods And Solvers Used For Solving Mixed Integer Linear Programming And Mixed Nonlinear Programming Problems: A Review

Pujari. Harish Kumar, Dr. R. Mageshvaran

Abstract: This paper presents a complete review of the significance of deterministic mixed-integer linear program (MILP) and mixed-integer nonlinear program (MINLP) solution methods for problems involving linear, nonlinear, convex and nonconvex functions. The mathematical description of methodologies, algorithms, software, and solvers to solve this problem are presented. Mixed-integer program (MIP) problem is one where some of the decision variables are constrained to be integer values. MILP involves problems in which only some variables are constrained to be integers, while other variables can be non-integers. It deals with linear objective function subjected to linear constraints. Minimisation and maximisation of a nonlinear objective function subject to nonlinear constraints with continuous and discrete variables are performed by MINLP solution methods. In this paper, the recent MILP and MINLP solution methods, algorithms along with solvers and software, introduced by various researchers has been elaborated profoundly.

Index Terms: Mixed-integer linear programming, Mixed-integer nonlinear programming, optimization solvers, Convex and Nonconvex, Combinatorial optimization, Global optimization.

◆

1 INTRODUCTION

In general, all the optimization problems that are available in the current scenario, necessitates the basic need for modelling the continuous variables and discrete variables using the Mixed Integer Linear Program (MILP) and Mixed Integer Non-Linear Program (MINLP) methods. Both the objective functions and constraints in MILP problems should be considered as linear, whereas, in the case of MINLP, they should be considered as a non-linear variable. Furthermore, the optimization problems related to MINLP are classified mainly into two groups. The first one is a convex MINLP which deals with the minimization of convex objective function with a feasible convex region. The second group is a non-convex MINLP which indeed deals with the non-convex objective function. At the same time, it also deals with the values that have convex nature, but unidentified in the feasible area. Refs. [1-4] describes MILP techniques elaborately and utilised the same for solving their optimization problems. To solve the MILP problems, different advanced methods had been devised experimentally in [5]. However, many researchers have developed a few commercial and non-commercial solvers to solve both the MILP and MINLP problems. CPLEX [6], Gurobi [7], and XPRESS-MP [8] are the some of the solvers that are developed and utilised in the literature. In [9] John K.Karlof have discussed the integer programming theory for solving both MILP and MINLP problems. The methods used to solve MINLP problems has significant advantages when compared to the methods that are used to solve MILP and Nonlinear programming (NLP) problems. In [10], Belotti et al. have provided a comprehensive review of MINLP problems and also about the methods to solve them.

- P. Harish Kumar, Research Scholar, School of Electrical Engineering, Vellore Institute of Technology, Vellore, Tamilnadu, India, E-mail: pujari.harishkumar2017@vitstudent.ac.in.
- Dr.R. Mageshvaran, Associate Professor, School of Electrical Engineering, Vellore Institute of Technology, Vellore, Tamilnadu, India, E-mail: rmageshvaran@vit.ac.in. (Corresponding author

Here, belotti also describes about the tree search method which is one of the widely used method to solve MINLP problems. The tree search method has been classified into single tree and multi-tree method. These two classes of methods solve a problem involving function like the convex type. The classical single tree method uses nonlinear branch and cut method, cutting plane method, and branch and bound method for solving MINLP problems. The conventional multi-tree approach comprises of both outer approximation and benders decomposition methods, in order to obtain global optimal solution for the considered problem. By combining the above approaches, a new hybrid method has been proposed in the literature, for solving convex MINLP problems more effectively, in terms of computation time and the quality of optimal solutions. In [11], Bonami et al. have provided a summary of various convex MINLP algorithms and software used to solve multiple MINLP problems. It is too challenging to solve nonconvex MINLP problems because it consists of nonconvex functions and nonlinear constraints in the objective function. Methods like Spatial branch and bound Piecewise linear approximation, and Generic Relaxation Strategies are used to solve nonconvex MINLP problems. In [12], Tawarmalani and Sahinidis N.V have explained about global optimization theory for solving MINLP problems using different algorithms and solvers.

2 MIXED INTEGER LINEAR PROGRAM (MILP)

An integer programming (IP) problem denoted as a mathematical optimization process and a viability program in which few or all of the variables are limited to be integers. In [13], if decision variables are not discrete, then the problem is considered as a MIP problem.

2.1 Linear Programming (LP)

Linear Programming (LP) technique has applied for optimizing a linear objective function, which is subject to both the linear equality and inequality constraints.
The canonical representation of a linear integer program is stated as in [14]:
Maximize $c^T x$, Subject to $Ax \leq b, x \geq 0,$ and $x \in \mathbb{Z}^n$ (1)

- The standard representation for an Integer linear programming (ILP) has stated as

Maximize $c^T x$, Subject to $Ax + s = b, s \geq 0, x \geq 0$, and $x \in \mathbb{Z}^n$
(2)

Whereas x is the vector of variables to be computed. c and b are the known coefficient vectors. A represents a known constant matrix in which all the entries are integers.

2.2 Mathematical representation of MILP

A Mixed Integer Programming (MIP) problem performs both minimize and maximize of problems with a linear objective function which is subject to linear constraints. But MILP involves problems with variables that are constrained to be only integers and allow all to be non-integers. The general representation of MILP is express as

Min { $c^T x \quad Ax \geq b, x \geq 0, x \in \mathbb{Z}$ }. (3)

For example, if variables are said to be rational, then the set is said to be null, which represents an LP problem solved by using the polynomial-time method. If variables are considered to be integers, then it denotes a MIP problem.

2.3 Algorithms used for solving ILP problems

Mainly, the underlying algorithms implemented for solving the ILP problems are listed as follows:

- Branch and Bound algorithm.
- Branch and Cut algorithm.
- Branch and Price algorithm.

2.3.1 Branch and Bound Algorithm

This methodology is used to find the solution for ILP problems. And the subproblems are formed according to the limiting range of integer variables in the program. Only two values are assigned to binary variables, one variable is 0, and the other is 1. The lower bound value represents as L and an upper bound value as U, which are considered to be variable that is separated into two problems having a range from ‘L to m’ and ‘m+1 to U’. In general, the LP relaxation method is being used to obtain lower bound values for a given problem, to satisfy both the objective function and limitations of constraints for an optimal solution. If the obtained optimal solution is integral, then it is represented as a solution to a subproblem. The search for the subproblem whose lower bound is higher can be terminated using these values [15].

2.3.2 Branch and Cut Algorithm

In this algorithm, the LP relaxation process is being utilized to estimate the lower bound values. The optimal solution for the ILP problem denotes the values in the feasible region, which satisfies the constraints. The step by step process of the branch and cut algorithm is:

Step 1: If obtained optimal solution by LP relaxation process is not integral, then the algorithm search for the constraint, which is violated by this optimal solution and not violated by optimal integer solutions. The constraint is known as cutting plane.

Step 2: If the old obtained optimal solution by the LP relaxation method is not valid, then the cutting plane constraint values are being added to the LP problem. The new optimal solution will be dissimilar, and this provides a better lower bound value.

Step 3: The cutting plane algorithm runs continuously until an optimal integral solution obtained; else, it is difficult to get the cutting plane for the problem. In [15], another technique known as the traditional branch method is used to obtain lower bound value and optimal solution.

2.3.3 Branch and Price Algorithm

A hybrid algorithm developed with the combination of methods: branch and bound method along with the column generation known as Branch and price algorithm. This algorithm has utilized for solving the ILP problems with many variables associated with the problem. For obtaining the optimal solution, this algorithm uses the current set of variables that has preserved, and the column generation is done according to the requirement of linear programming problem. In [15], the column generation procedure has used for a given specific problem, and this can be again used to interrelate with branching decisions to obtain the lower bound value.

2.4 Basic components of MILP solvers

The main essential parts of MILP solvers are Pre-solving, Cutting plane generation, Sophisticated branching strategies, Primal Heuristics in MILP, Variable selection, and Parallel implementation.

2.4.1 Pre-solving

In the pre-solving phase of MILP, solvers make an effort to identify individual variations in the input, which helps in achieving the improved execution of the solution strategy, without changing the optimal solution for the problem. Pre-solving denotes to a group of problem reductions that have used before the start of the branch and bound procedure, and this collection makes to construct its formulation and reduce the size of the problem. The MILP models are converted into problem formulation, effectively by removing excess data (like variables and constraints) and by reinforcing variable integrity. One main advantage of this pre-solving method is to provide a smaller LP relaxation process that helps in solving the problems quickly and for improving the approximating of the convex integer solution. Finally, more sophisticated pre-solve mechanisms are available to find the necessary inferences and substructures for the computation of branching determination and the cutting plane generation. Martin in [16] and Savenlsbergh in [17] has clearly explained pre-solving methods to solve MILP problems.

2.4.2 Cutting plane algorithm

Ralph E.Gomory developed this algorithm in 1950 and used for solving both ILP and MIP problems. The basic idea of the algorithm is to cut off some part of the feasible regions in the solution obtained by the LP relaxation process, in such way optimal integer solution becomes an extreme point and can be found by using the simplex method [18]. The cutting plane algorithms include different groups such as Chvatal Gomory cuts, mixed-integer rounding cuts, lift-and-project cuts, Gomory mixed-integer cuts, and split cuts [19].

The basic form of formulating the cutting plane algorithm is as follow:

Step 1: If the solution of the LP problem is optimal, and if the feasible region of the solution doesn’t contain a line, then easily an extreme point is obtained, which is optimal.

Step 2: If an obtained optimum is not an integer solution, a linear inequality exists which separates the optimum from the convex hull of the exact feasible set.

Step 3: Finding such inequality is called a separation problem and is considered as cutting line in the feasible region.

Step 4: Then, the obtained current non-integer solution is no longer feasible for the relaxation process, and the above steps continuously repeated until an optimal integer solution is obtained.

2.4.3 Sophisticated Branching Strategies

At any step of processing, the branching mechanism requires two independent and significant decisions, one is node selection, and the second is variable selection. The branching strategies involves choosing a variable Y_i , with a fractional value Y_i^1 , in the optimal solution to the current LP relaxation process. Then constraints for the variable are formulated as [$Y_i \leq Y_i^1$] and [$Y_i \geq Y_i^1$].

- Node Selection: Doran and Michie proposed the node selection process in 1966 based on the evaluation function. The different procedures used by Rafael Pastor, Albert Corominas to the node selection are as follow [20]:
  - British Museum Procedure.
  - Random Search [Rich and knight in 1991].
  - Depth First Search (Liao in 1994) or Single Branch Search (Ibaraki in 1976).
  - Breadth-First Search (Dijkstra and Moore in 1959).
  - Best Bound Search (Ibaraki 1976/1987).
  - Best Few Search (Agusti and Lopez in 1987).
  - Parallel Depth First Search (Imai in 1982).
  - Jump Backtracking (Fisher and Hochbaum,1980).
  - Locally Best Bound Rule (Liao in 1994).
  - Weighted -A* algorithm (Korf in 1993).
  - Depth m-Search (Ibaraki in 1978).
- Variable Selection: This selection problem decides how to perform the partition of the current node consequently, which variable to branch on it, such that to create two children. The variable selection mainly done based on branching rules which have presented in [21] are listed as the most fractional variable, active branching variable, pseudo cost branching variable, and branching priorities. Basically, in the variable selection problem, a traditional choice is to branch on a variable with the fractional part closest to 0.5. In [22], Pseudocost branching was another sophisticated technique used for the variable selection problem implemented by Benichou. This technique keeps track of success variables which are already branched on.

Another method developed by Linderoth in [23], for the variable selection problem based on the strong branching technique. In this technique at any node, each fractional variable has to branch on for checking progress, before actual branching is done.

2.4.3 Primal Heuristics in MILP

This method provides solutions and improvement for finding the Integer Feasible Solutions (IFS) in the search tree and the upper bound value for a minimization problem. There are two heuristic methodologies implemented in MILP solver, which are rounding and iterative rounding heuristic. Heuristic methods construct the solution for the IFS by using partial solutions. The transformation of the solution obtained from the LP relaxation method into IFS has achieved by applying the heuristics methods at the internal nodes and root node of the search tree in MILP solver.

2.4.5 Parallel Implementation

When the process of search tree node independent, parallelization is a must for the search tree. The branch and bound algorithm are used widely for these purposes [24]. The different categories of parallel integer programming investigation are sorted depending on the type of parallel computing architecture that existed. The Message transferring to communicate solution of the algorithm depends on distributed memory architecture. Shared memory computers move the data between the Central Processing Unit (CPU) by reading from, and writing to a common memory pool. In the parallel branch and bound method, the order in which node calculations performed may cause an impact on the execution of the program and sometimes lead to abnormal behaviour [25].

2.5 MILP Optimization software packages

A complete review of the features of both commercial and non-commercial MILP software packages has presented in [26]. The history of this MILP software package development and specifications like SCICONIC, UMPIRE, and MPSX/370, etc., are evaluated by Forrest and Tomlin in [27]. With the help of these package set up, several successful methods have formulated for handling the problems related with the branching and node selection, out of which the significant techniques are still in use for solving MILP. In [19] & [20] some fundamental changes as having been advanced in MILP techniques to solve many major problems in the engineering field. The advanced methods used like the pre-processing method and structure-specific cutting plane method are used even today for solving many of the engineering problems. Some authors in [28,29] represented the inequalities which are also utilised efficiently as a branch and cut method, that can be accompanied in the cutting plane procedure. In [30] detailed history of MILP software over a past period is explained clearly by R.Bixby.

2.6 User Interface

The user interface is an essential feature for designing software for solving MILP problems. As the extent of purposes for MILP programming software is generally enormous, as a result of this user interface types number is correspondingly becoming large.

2.7 Commercial software packages

The following commercial software packages used to solve MILP problems are available for academic users at free (no cost) and limited cost licensing options. Han Mittelmann has executed independently many MILP software to get optimal solutions and had published results for the past many years. Table .1 provides information about the MILP Optimization Software Packages.

TABLE 1

MILP OPTIMIZATION SOFTWARE PACKAGES

Types of MILP Optimization Software Packages.	*Software name*	*Founders*	*Algorithms utilized*	*Parameters included*	*Features*	*Specifications*	*Interfaces, modelling languages*
Commercial Software Packages	*CPLEX [31] (IBM ILOG CPLEX Optimization Studio)*	*Bixby the founder of CPLEX, retained and provided by IBM.*	*Branch and cut algorithm and Dynamic search algorithm.*	*Mipemphasis meta parameter*	*Capable of calculating multiple optimal solutions and the solutions have stored in a solution pool.*	*Version: 12.8.0, Website: http:// www.ibm.com/ analytics/cplex-optimizer. License: proprietary.*	*C, C++, Java, .NET, MATLAB, Python, Microsoft Excel.*
	GUROBI	*Zonghao Gu, Edward Rothberg, and Robert Bixby.*	*Include cutting planes algorithm, heuristics and search techniques.*	*MIP-Focus meta parameter*	*New MILP solver that is designed with modern multicore processing technology to obtain an optimal solution.*	*Version: 3.0, Website: www.gurobi.com. License: proprietary.*	*Object-oriented interfaces for C++, Java, .NET, and Python.*
	LINDO [32] (Linear, Interactive, and Discrete Optimizer)	*LINDO SYSTEMS INC.*	*It offers different forms of cutting planes algorithms and different node selection rules.*	*LINDO also comprises a mipemphasismeta parameter that has used for adjusting algorithm parameters.*	*Significantly Faster on Large Quadratic Models. Improved Handling of Models with Discontinuous Functions.*	*Version:10.0, Website: www.lindo.com. License: proprietary.*	*C, Visual Basic, MATLAB, Microsoft Excel.*
	MOSEK	*Mosek ApS, a Danish company.*	*Branch and bound, branch and cut, and state-of-the-art interior-point optimizer algorithm.*	*Parameters include optimizer choice for solving linear problems, turning pre-solve parameter value and feasibility of tolerances value.*	*MOSEK interior-point optimizer can reliably detect a primal and dual infeasible status of solutions.*	*Version:9 beta, Website: www.mosek.com. License: proprietary.*	*C, C++, Java and Python languages. Mosek is accessible for use by customers through a GAMS interface on the NEOS Server.*
Non-Commercial Software Packages [33]	*BLIS(BiCePS Linear Integer Solver),*	*Open source solver developed by the COIN-OR project to solve MILP problems. (COR@L Lab).*	*COIN-OR linear programming solver, cutting planes algorithm and parallel tree search algorithms are used to solve MILP problems, available from the Coin Cut Generation Library (CGL).*	*Parameters to solve problems are taken from the COIN-OR linear programming solver and CGL.*	*It controls the suitable methods to progress a state-of-the-art parallel algorithm for a particular problem set.*	*Version:0.91, Website: https://projects.coin.or.org/ CHiPPS. License: Common Public License.*	*C++ library is similar to SYMPHONY.*
	CBC (COIN-OR Branch and Cut Solver)	*John Forrest, an open-source solver, developed by COIN-OR (Computational Infrastructure*	*COIN-OR linear programming solver, cutting planes algorithm and parallel*	*Parameters taken to solve problems are taken from the COIN-OR linear programming solver and CGL.*	*CBC can be parallelized using shared-memory parallelism to solve the problem for*	*Version:2.5, Website: https://projects.coin.or.org/CBC. License: Common Public License.*	*C++*

		for Operations Research.) project to solve MILP problems. (COR@L Lab).	*tree search algorithms are used to solve MILP problems, available from CGL.*		*finding an optimal solution.*		
	GLPK [34] (GNU Linear Programming Kit)	*Andrew O. Makhorin, GNU Project.*	*GLPK utilizes the simplex method and the primal-dual interior-point method to solve non-integer problems. The branch and bound algorithm, including Gomory mixed-integer, cuts to solve integer problems.*	*A set of subroutines with a callable library and black box solver has used.*	*Problems have modelled in the GNU language, MathProg shares the syntax with AMPL, and is solved with standalone solver GLPSOL.*	*Version: 4.44, Website: http:// www.gnu.org/ software/glpk/, License: GNU General Public License (GPL).*	*GNU, MathProg, C.*
	MINTO (Mixed Integer Optimizer)	*Savelsbergh and Nemhauser, 1993.*	*Branch and bound algorithm with LP relaxations method.*	*Global and local constraints are valid at any node and constraints are generated.*	*It offers constraint classifications, pre-processing, primal heuristics and constraints generation automatically. MINTO is accessible for users through an AMPL language interface, on the NEOS Server.*	*Version:3.1, Website: http://coral.ie.lehigh.edu/minto/, License: Provided by the library only.*	*C, MINTO 1.4 is accessible on SUN SPARC station with CPLEX.*
	SCIP [35] (Solving Constraint Integer Programs)	*Konrad Zuse-Zentrum fur Information stechnik Berlin (ZIB).*	*Branch and cut algorithm, and Branch and price algorithm used to solve problems.*	*It supports about 20 constraint types for MILP, MINLP, and mixed integer all quadratic programming*	*The strategy of SCIP has constructed on the concept of constraints.*	*Version: 1.2, Website: http://scip.zib.de/, ZIB Academic License.*	*C, C++, GAMS, MATLAB, JAVA PYTHON.*
	SYMPHONY [36]	*Developed by COIN-OR.*	*Sequential and parallel types of the branch, cut, and price algorithms are used to solve MILPs problems.*	*The user of the library can change the algorithm using custom data files.*	*Ability to Solve bi-objective MILPs and perform basic sensitivity studies.*	*Version:5.6,5.6.17, Website: https://projects.coin-or.org/SYMPHONY/, www.coin-or.org, License: Common Public License (CPL), open-source solver.*	*C, AMPL, GMPL, GAMS (General Algebraic Modelling System).*

3 MINLP INTRODUCTION AND MATHEMATICAL REPRESENTATION OF MINLP

MINLP optimization problems contain both the discrete, continuous variables and nonlinear functions in the objective function and constraints. MINLP includes non-linear programming (NLP) and MILP as subproblems while modelling a system for obtaining an optimal solution. The general

formula of an MINLP model representation has expressed as follow:
$Min\ z = f(x,y), g(x,y) \leq 0, x \in X, y \in Y\ (MINLP)$ (4)
Where $f(\cdot)\ and\ g(\cdot)$ are differentiable functions, x is a continuous variable and y are discrete variables.

MINLP has expressed as:
$Max\ f(\cdot) = -\min(-f(\cdot))$
(5)

Generally, the integer variables are limited to 0, and 1 values and constraint function are naturally linear [37]. There are two nonlinear programming subproblems obtained from MINLP, used in many solving methods.

- Continuous relaxation of MINLP(r-MINLP): It is stated as

$Min\ z = f(x,y), g(x,y) \leq 0, x \in X, y \in Y\_(R)\ (r - MINLP)$ (6)

Where $Y\_(R)$ denotes the continuous relaxation of Y, integer variables y has considered as continuous.

- Fixed MINLP $(fx - MINLP)$: The following expression is used when nonlinear programming subproblems having fixed $y\hat{}p$,

$Min\ z = f\ (x, y\hat{}\ p)\ , g(x, y\hat{}p)\ \leq 0, r \in\ R\hat{}n\ \ (fx - MINLP)$ (7)
when the solution is feasible, the upper bound and lower bound for MINLP are obtained by $(fx - MINLP)$ and $(r - MINLP)$ respectively.

3.1 Classifications of MINLP
MINLP generally classified into two types, such as convex MINLP and nonconvex MINLP.

3.1.1 Convex MINLP
Convex MINLP includes a minimization problem with the objective function around a feasible convex region. The different methods used for solving Convex MINLP has described below.

- Nonlinear Branch and Bound Method.
- Outer Approximation Method.
- Generalized Benders Decomposition Method and
- Extended Cutting Plane Method.
- Nonlinear Branch and Bound Method: It might have considered as the advanced method to the linear branch and bound method. This method uses an extensive tree search process for obtaining an optimal solution over an integer variable [38-40].

Step 1: Initially, it solves the continuous relaxation method on MINLP problems.
Step 2: If the solution assigns integer values to all integer variables using the linear relaxation method, then it is said to be an optimal solution, and then the algorithm stops working.
Step 3: And if integer variable values have not assigned at the current node, then it is chosen as $[V\_i = (V\_i\ \hat{}(0))]$.
Step 4: Next branching method is applied to these integer variables, forms a new NLP problem having bound value represented as $[V\_i\ \leq (V\_i\ \hat{}(0))]\ and\ [V\_i\ \geq (V\_i\ \hat{}(0))]$.
Step 5: The above steps have repeated until an optimal integer feasible solution is obtained.

- Outer Approximation: This method employs two problems such as Master-MIP (M-MIP) and Fixed-MINLP $(fx - MINLP)$ [41,42]. The M-MIP have used for finding a lower bound value $(w\ \hat{}\ lower\ value)$ and also used to find out integer solution for the given estimated problem $(v\ \hat{}\ q)$. These lower bound problems are known as a master problem. For solving the subproblem solution, first the binary variables are fixed, and the next $fx - MINLP$ has computed. If the solution obtained for $fx - MINLP$ is a feasible one, then it gives an upper bound value $(w\ \hat{}\ upper\ value)$.If the obtained solution is not feasible, and then the feasible-MINLP method is used to solve the subproblem to get upper bound value. This process has performed until tolerance exists between lower and upper bound values.
- Generalized Benders Decomposition (GBD): GBD is almost similar to the outer approximation process; a slight difference exists in the linear master problem formulation [43]. The master problem formulation considers discrete variables and inequality variables.
- Extended Cutting Plane (ECP): The ECP method also follows the same concept as an outer approximation, but this method does not solve the nonlinear programming subproblems [44].In this process solution obtained from a master MILP, $(u\ \hat{}\ q), (v\ \hat{}\ q)$ contains all the constraints that are linearized, and then linearized constraints are violated have been added to the master problem. When violation lies is almost of the specified tolerance value, convergence has achieved. After each iteration algorithm gives a lower bound value. The main disadvantages of this algorithm are convergence is slow, and it doesn't provide an upper bound value, feasible solution until it achieves required convergences value [45].

3.2 Convex MINLP optimization solvers
Many convex MINLP solvers are used to optimization problem solving in [46,47]. Some essential convex MINLP solvers have discussed below.

- Basic open source nonlinear mixed integer programming (BONMIN) [48]: This is a free solver to the users for solving MINLP problems, using methods like outer approximation, branch and bound method, extended cutting plane, NLP branch and bound method.
- Discrete and Continuous Optimizer (DICOPT) [49]: This is a commercial solver to the users for solving the MINLP problems by using an outer approximation algorithm that has interfaced with GAMS.
- Mixed Integer Linear and Nonlinear Optimization [50]: This is an open-source solver used to solve nonlinear programming (NLP) problems by using nonlinear branch and bound, and outer approximation algorithms in MATLAB software.
- Mixed-Integer Nonlinear Programming Branch and Bound (MINLPBB) [51]: MINLPBB is a commercial solver to users for solving the MINLP problems by using generalized bender decomposition and outer approximation algorithms.
- Mixed-Integer Nonlinear Optimization Toolkit Algorithms Under Estimators and Relaxation (MINOTAUR): This is also an open-source toolkit used by users for solving convex MINLP problems by using nonlinear branch and bound algorithm interfaced with C++ [52].
- Alpha Extended cutting plane(α-ECP) [53]: This is a commercial based solver used for solving MINLPs by using the extended cutting plane method.
- Simple Branch and Bound (SBB): This solver has executed in GAMS, which allows the users to select an

NLP solver for solving MINLP problems by using the NLP branch and bound algorithm.
- Branch and Reduce Optimization Navigator (commercial solver) [54].
- Algorithms for Continuous Global Optimization of nonlinear equations [55].
- Convex Over and Under Envelopes for Nonlinear Estimation (open sources solver) [56].
- Solving constraint integer programs (non-commercial solver) [57].

3.3 Non-Convex MINLP
The most complex and dynamic fields in optimization are non-convex MINLP. For the exact modelling of the real-time problems, the nonconvex constraints are used for obtaining the optimal solution is necessary [58,59]. The spatial branch and bound algorithm are the deterministic methods used for solving the non-convex MINLP problems. The main concept related to the nonconvex solving process involves relaxations of factorable formulations [60,61] and bounds tightening [62-65]. In [66,67], a complete explanation about deterministic and heuristic techniques has presented.

3.3.1 Spatial branch and bound algorithm
The spatial branch and bound algorithm is a deterministic method used for solving non-convex MINLP problems and by using a convex relaxation process to find the optimal solution.

Step1: This algorithm calculates convex relaxation of the NLP problem which affords the lower bound value.
Step 2: Next, a local or open-source solver is used to solve the NLP problem. If the NLP problem has a feasible solution, then it gives an upper bound value.
Step 3: if the solution is not feasible, then the upper bound value is assumed as infinity.
Step 4: If the differences between the upper bound and lower bound values are within the tolerance limits, the algorithm stops executing and provides an optimal solution for the problem.
Step 5: Otherwise, the two new NLP problems have formed by using branches of variables. Later, each NLP problem computed using the solver to obtain both upper bound and lower bound value by using previous steps. The upper bound has taken from the new region, which is having the lowest upper bound value.
Step 6: This process is repetitive until the upper bound and lowers bound is within the tolerance limits.

3.3.2 Non-Convex MINLP solvers
Many MINLP solvers have been built to solve the nonconvex problems by using the branch and bound algorithm. The solvers used to solve the nonconvex MINLP are α-branch and bound solver (α-BB), Convex over and under envelopes for nonlinear estimation (COUENNE), Branch and reduce optimization navigator (BARON) and Lipschitz global optimizer (LGO). The main advantage of the MILP over MINLP is that convergence of the solution is definite for the given problem [68]. Otherwise, if the obtained solution is optimal, then it is easy to determine the optimum local value for the MILP problems. In case if the constraints oppose each other or if the problem is abundant in the way of the objective function, then the optimal solution cannot be determined [69]. The solution accuracy can be specified using the branch and bound gap method [70]. MILP problem can be solved quickly, extremely fast and effectively using commercially existing solvers such as CPLEX, Gurobi, Xpress and MATLAB [71].

3.4 MINLP and MILP applications
MINLPs has widely used in the field of engineering and scientific applications. MINLP models of electrical engineering applications comprise the efficient management of electricity transmission in [72], transmission expansion in [73], transmission switching and contingency analysis in [74] and blackout prevention of electric power systems in [75]. Chemical engineering applications of MINLP models comprise the design of water in [76], gas distribution networks in [77], and for the optimal design of reliable chemical plants in [78]. Other applications MINLP model comprise like operational reloading of nuclear reactors in [79] and electric distribution networks in [80]. In [81], MINLP models have used for stochastic service system design problems. MILP has used in various fields of engineering and scientific applications. Electrical engineering applications of MILP include optimizing distributed generation locations and sizes, considering different DGs modes of connection to the distribution grid in [82]. In [83] S-MILP operational model is proposed for minimizing the sum of the most associated cost terms (switching costs, emissions, expected costs of operation, and unserved power in the system), by satisfying the number of model constraints. In [84] MILP based technique is implemented for smart self-healing grids. In [85], reconfigurable computing, architectures and design methods are developed based on MILP models. In [86], approximated MILP models are used to AC transmission expansion planning. In [87] MILP model is used to reliability optimization in active distribution networks. In [88] MILP branch flow model is used in synchronize AC multistage transmission expansion and reactive power planning with security constraints. In [89] MILP model is proposed for integrating expansion planning of multi-carrier active energy systems. In [90] MILP model is implemented in the charging coordination problem of electric vehicles in unbalanced distribution systems.

4 CONCLUSION
In this paper an overview of the deterministic MILP and MINLP solution methods with the mathematical description of methodologies, algorithms, software, and solvers had been presented. MIP problem is one of the linear programs where the variables are constrained as integer. However, MILP involves problems with some variables, constrained to be integers and other as non-integers. Whereas, MINLP solution method for problem performs both minimizing and maximizing a nonlinear function subject to nonlinear constraints. Here, various MILP and MINLP methods and algorithms introduced by the researchers along with solvers and software that are used to solve the real time problems in the field of engineering and science had been discussed in detail.

5 ACKNOWLEDGMENT
The authors are extremely grateful to the Chancellor, Vice-Chancellor and Vice-Presidents, of VIT University, Vellore, for providing the excellent infrastructure facilities and encouragement which have made this research work possible.

6 REFERENCES

[1] Harold. Greenberg, “8 Branch and Bound Procedures,” Mathematics in Science and Engineering Elsevier, vol. 76, pp. 177-191, 1971.

[2] Alexander. Schrijver, “Theory of Linear and Integer Programming,” A Wiley-Interscience Publication, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, USA;1999.

[3] Nemhauser. George, et al. “Integer and Combinatorial Optimization,” publisher: Wiley-Interscience, Wiley, ISBN:0-471-82819-X, New York, NY, USA;1988.

[4] Robert J. Vanderbei. “An Interior Point Code for Quadratic Programming,” Journal Optimization Methods and Software, vol.11, no.1-4, pp.451-484, 1999.

[5] R. Bixby and E. Rothberg, “Annals of Operations Research,” Kluwer Academic Publishers-Plenum Publishers, vol.149, no.1, pp.37-41,2007.

[6] IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual, Version 12 Release 6, tech. rep.,2013, https://www.ibm.com/support/knowledgecenter/SSSA5P_12.

[7] Gurobi solver.Gurobi Optimizer Reference Manual 2018.Version 8.1.www.gurobi.com.

[8] FICO XPRESS-MP Reference Manual.tech. rep., 2009.

[9] John K.Karlof. “Integer programming theory and practice. eBook”, CRC Press Taylor and francis group, pp.1–305,2006.

[10] J.Kronqvist and Bernal, E. David and Lundell, Andreas and I.E. Grossmann, “A review and comparison of solvers for convex MINLP,” Optimization and Engineering, vol.20, no.2, pp.397-455, 2019.

[11] P.Bonami, M. Kilinc, et al. , “Algorithms and Software for Solving Convex Mixed Integer Nonlinear Programs,” Chapter in Jon Lee et Sven Leyffer editors,Mixed Integer Nonlinear Programming, IMA Volumes in Mathematics and its Applications,vol.154,pp.1–39,2012.

[12] M.Tawarmalani, N.V. Sahinidis, “Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications,” Kluwer Academic Publishers, vol.65, pp.1-478,2002.

[13] C.Benoit, “Mixed Integer Linear Programming (MILP): Model Formulation (PDF),” ChE 4G03.Optimization in Chemical Engineering, pp.1-7,2018.

[14] C.H. Papadimitriou, K.Stieglitz, “Combinatorial optimization: algorithms and complexity,” Mineola.Publisher: Dover Publications.Unabridged edition,pp.1-496,1998.

[15] L.N.George ,et al., . “Integer and Combinatorial Optimization,” Publication: John Wiley and Sons, Wiley-Interscience New York, NY, USA.ISBN:0-471-82819-X, pp.31-32,1998.

[16] A.Martin ,” General mixed integer programming: Computational issues for branch-and-cut algorithms,” Computational Combinatorial Optimization,2241 of Lecture Notes in Computer Science,pp.1–25,2001.

[17] M.P.W. Savelsbergh, “Pre-processing and probing techniques for mixed integer programming problems,” ORSA Journal on Computing, vol.6, pp.445-454, 1994.

[18] R.E. Gomory, “Outline of an algorithm for integer solutions to linear programs,” Bulletin of the American Mathematical Society, vol.64, pp.275–278,1954.

[19] G. Cornuejols, “Valid inequalities for mixed integer linear programs,” Mathematical Programming, vol.112, pp.3–44, 2008.

[20] Rafael Pastor, Albert Corominas. “Strategies of Node Selection in Search Procedures for Solving Combinatorial Optimization Problems: A Survey and a General Formalization,” Sociedad de Estadistica e Investigacidn Operativa, vol.8, no.1, pp.111-134,2000.

[21] T.Achterberg, T.Koch, et al., “Branching roles revisited,” Operations Research Letters, vol.33,pp.42–54,2005.

[22] M. Benichou,J.M Gauthier,et al., “Experiments in mixed-integer programming,” Mathematical Programming,vol.1, pp.76–94,1971. doi: 10.1007/BF01584074.

[23] J.T. Linderoth, M.P.W. Savelsbergh, “A computational study of search strategies for mixed integer programming,” INFORMS Journal on Computing, vol.11, pp.173–187,1999.

[24] T.G. Crainic, B. Gendron, “Parallel branch and bound algorithms: Survey and synthesis,” Operations Research, vol.42, pp.1042–1066,1994.

[25] T.H.Lai, S. Sahni, “Anomalies in parallel branch and bound algorithms,” In Proceedings of the 1983 International

Conference on Parallel Processing ,pp.183–190,1983.
[26] T.L. Jeffrey, et al., “MILP software,” citeseerx.ist.psu.edu/view, pp.1-17,2001.
[27] J. J. H. Forrest,J.A. Tomlin, “Branch and bound, integer and non-integer programming,” Annals of Operations Research ,vol.149,pp.81–87,2007.
[28] E.Balas, S.Ceria et al., “Gomory cuts revisited,” Operations Research Letters, vol.19,pp.1–9,1996.
[29] R.E. Gomory, “An algorithm for the mixed integer problem,” Technical Report RM-2597, The Rand Corporation 1960.
[30] R. Bixby, E. Rothberg, “Progress in computational mixed integer programming – a look back from the other side of the tipping point,” Annals of Operations Research, vol.149, pp.37–41,2007.
[31] H.D Mittelmann, “Recent Benchmarks of Optimization Software,” 22nd European Conference on Operational Research, vol.4, pp.1-314,2014.
[32] C.Nash, John, “Optimizing Add-Ins: The Educated Guess. What's Best! PC Magazine,” vol.10, pp.130-132,1991.
[33] J.T Linderoth, T.K Ralphs, “Non-commercial software for mixed integer linear programming. In Integer Programming: Theory and Practice,” CRC Press Operations Research Series 2005, pp.253–303,2005. doi: 10.1201/9781420039597.ch10.
[34] Eiji Oki, “Linear Programming and Algorithms for Communication Networks: A Practical Guide to Network Design Control and Management,” CRC Press 2012.
[35] T. Achterberg, “SCIP: solving constraint integer program,” Mathematical Programming Computation, vol.1, no.1, pp.1–41,2009. doi: 10.1007/s12532-008-0001-1.
[36] M.Guzelsoy, T.K Ralphs, “The SYMPHONY callable library for mixed integer programming,” In the Proceedings of the Ninth Conference of the INFORMS Computing Society 2005, pp.61–71,2005.
[37] I.E. Grossmann, “Review of nonlinear mixed-integer and disjunctive programming techniques,” Optimization and Engineering, vol.3, pp.227–252,2002.
[38] S. Nabar, L. Schrage, “Modelling and Solving Nonlinear Integer Programming Problems,” Presented at Annual AIChE Meeting 1991.
[39] B. Borchers, J.E. Mitchell, “An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programming,” Computers and Operations Research, vol.21, pp.359–367,1994.
[40] S. Mehrotra, R.A. Stubbs, “A branch-and-cut method for 0-1 mixed convex programming,” Mathematical Programming, vol.86, pp.515–532,1999. doi: 10.1007/s101070050103.
[41] P. Kesavan , R.J. Allgor and P.I.Barton, “Outer approximation algorithms for separable non-convex mixed-integer nonlinear programs,” Mathematical Programming ,vol.100,pp.517–535,2004.
[42] R. Fletcher,S. Leyffer, “Solving mixed integer nonlinear programs by Outer Approximation,” Mathematical Programming,vol.66,pp.149-327,1994. doi: 10.1007/BF01581153.
[43] O.E. Flippo,A.H.G.R Kan, “Decomposition in general mathematical programming,” Mathematical Programming,vol.60,pp.361–382,1993.
[44] T.Westerlund , F.Pettersson, “An extended cutting plane method for solving convex MINLP problems,” Computers and Chemical Engineering,vol.19,pp.131–136,1995.doi:10.1016/0098-354(95)87027-X.
[45] J.E. Kelley, “The cutting plane method for solving convex programs,” Journal of the SIAM, vol.8, pp.703–712,1960.
[46] M.R. Bussieck, S. Vigerske, “MINLP solver software,” Wiley Encyclopedia of Operations Research and Management Science 2010.
[47] C.D. Ambrosio, A.Lodi, “Mixed integer nonlinear programming tools: a practical overview,” 4OR ,vol.9,pp.329–349,2011. doi: 10.1007/s10288-011-0181-9.
[48] P.Bonami , L.Biegler , et al., “An algorithmic framework for convex mixed integer nonlinear programs,” Discrete Optimization journal, vol.5,pp.186–204,2008. doi: 10.1016/j.disopt.2006.10.011.
[49] J.Viswanathan , I.E. Grossmann, “A combined penalty function and outer-approximation method for MINLP optimization,” Computers and Chemical Engineering, vol.14,pp.769–782,1990. doi: 10.1016/0098-1354(90)87085-4.
[50] H.Y. Benson, D.F. Shanno, “An exact primal-dual penalty method approach to warm starting interior point methods for linear programming,” Computational Optimization and Applications, vol.38, pp. 371–399,2007. doi: 10.1007/s10589- 007-9048-6.

[51] H.Y. Benson, D.F. Shanno, "Interior-point methods for nonconvex nonlinear programming: regularization and warm starts," Computational Optimization and Applications, vol.40, pp.143–189,2008. doi: 10.1007/s10589-007-9089-x.

[52] P.Belotti , Christian Kirches, et al., "Mixed integer nonlinear optimization," Acta Numerica journal,vol.22,pp.1-131,2013.

[53] T.Westerlund , K. Lundqvist, "Alpha-ECP.Version 5.01 An interactive MINLP-solver based on the Extended Cutting Plane Method," Process design laboratory, Abo Akademie university 2001.

[54] Sahinidis, N.V. Baron, "A general-purpose global optimization software package," Journal of Global Optimization, vol.8, pp.201–205,1996. doi: 10.1007/BF00138693.

[55] R.Misener,C. Floudas, "ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations," journal of global optimization, vol.59,pp.503–526,2014.doi: 10.1007/s10898-014-0166-2.

[56] P.Belotti , J.Lee, "Branching and bounds tightening technique for non-convex MINLP," Optimization methods and software, vol.24,no.4,pp.597-634,2009. doi: 10.1080/10556780903087124.

[57] Berthold Timo, Stefan Heinz and Marc E. P fetsch, "Solving Pseudo-Boolean Problems with SCIP," ZIB report 2008, vol.8, no.12, pp.1–15,2008.

[58] C.A. Floudas, "Deterministic global optimization: theory, methods, and applications," eBook Springer, vol.37, pp.1-698,2000.

[59] I.E Grossmann, T.Francisco, "Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods," Chemie Ingenieur Technik journal ,vol.86,pp.991-1012,2014.doi: 10.1002/cite.201400037.

[60] G.P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I Convex underestimating problems," Mathematical programming, vol.10, pp.147–175,1976.

[61] E.Smith , C.C Pantelides, "Global optimization of nonconvex MINLPs," Computers and Chemical Engineering,vol.21,pp.791–796,1997. doi: 10.1016/S0098-1354(97)87599-0.

[62] E.Davis, "Constraint propagation with interval labels," Artificial intelligence,vol.32,no. 3,pp.281-331,1987.

[63] M.W. Savelsbergh, "Pre-processing and probing techniques for mixed integer programming problems," ORSA Journal on Computing, vol.6, pp.445–454,1994.

[64] D. Erling,K. Andersen, "Pre-solving in linear programming," Mathematical Programming ,vol.71,pp.221–245,1995.

[65] F.Messine, "Deterministic global optimization using interval constraint propagation techniques," RAIRO Operations Research, vol.38,pp.277–293,2004. doi: 10.1051/ro:2004026.

[66] M. Gaviano,D.Lera, et.al., "A Parallel Algorithm for Global Optimization Problems in a Distributed Computing Environment," Applied Mathematics,vol.10A,pp.1380-1387,2012.doi:10.4236/am.2012.330194

[67] L. Liberti, "Introduction to global optimization," Technical report. LIX. Ecole Polytechnique, Palaiseau F-91128, France, liberti @lix.polytechnique.fr 2008.

[68] Luca Urbanucci, "Limits and potentials of Mixed Integer Linear Programming methods for optimization of poly-generation energy systems," Energy Procedia, vol.148, pp.1199-1205,2018. doi: 10.1016/j.egypro.2018.08.021.

[69] Ortega Jordi, "Modelling environment for the design and optimization of energy poly-generation systems," Ph.D. Thesis 2010.

[70] Chapin, E. Jonathan, "Review of Integer Programming," by A. Laurence, Wolsey, Wiley and Sons 1998, journal: SIGACT News 2005, vol.36, no.3, pp.10-12,2005. doi: acm.org/10.1145/1086649.1086655

[71] I.E. Grossmann, "Optimization and Engineering," Publisher: Kluwer Academic Publishers, vol.3, pp.227-252,2002.

[72] R.Bacher, "The Optimal Power Flow (OPF) and its solution by the interior point approach," EES-UETP Madrid. Short Course, pp.10-12,1997.

[73] L.L. Garver,"Transmission network estimation using linear programming," IEEE Transactions on Power Apparatus Systems,vol.89,pp.1688–1697,1997.

[74] E.F. Bartholomew, R.P. O'Neill, el.al., "Optimal transmission switching," IEEE Transactions on Power Systems, vol.23, pp.1346–1355,2008.

[75] D.Bienstock, S.Mattia , "Using mixed-integer programming to solve power grid blackout problems," Discrete Optimization,vol.4,pp.115–141,2007.

[76] R.Karuppiah , I.E. Grossmann, “Global optimization for the synthesis of integrated water systems in chemical processes,” Computers & Chemical Engineering ,vol.30,pp.650–673,2006.

A. Martin,M.Moller, “Mixed integer models for the stationary case of gas network optimization,” Mathematical Programming, vol.105,pp.563–582,2009.

[77] Yixin Ye, I.E. Grossmann, et.al., “Mixed-integer nonlinear programming models for the optimal design of reliable chemical plants,” Computers and Chemical Engineering, pp.1-14,2018.

A. J Quist, T. Terlaky, et.al., “Application of nonlinear optimization to reactor core fuel reloading,” Annals of Nuclear Energy, vol.26, pp.423–448,1998.

[78] S.Lakhera , U.V Shanbhag U.V, et.al., “Approximating electrical distribution networks via mixed-integer nonlinear programming,” International Journal of Electric Power and Energy Systems, vol.33,no.2,pp.245–257,2011.

[79] S.Elhedhli, “Service System Design with Immobile Servers, Stochastic Demand, and Congestion,” Manufacturing & Service Operations Management, vol.8,no.1,pp.92–97,2006.

[80] Teshome, F.Dawit, “The comprehensive mixed-integer linear programming model for distribution system reconfiguration considering DGs,” IET Generation Transmission &amp Distribution,vol.12,no. 20,pp.4515-4523,2018,IET Digital Library.

[81] F.V, Dantas, S.F Santos, et.al., “Dynamic Reconfiguration of Distribution Network Systems: A Key Flexibility Option for RES Integration,”2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 13 July 2017.

[82] Al-Owaifeer, Al-Muhaini, Mohammad, “MILP-based technique for smart self-healing grids,” IET Generation Transmission &amp Distribution, vol.12, no. 10, pp. 2307-2316,2018, IET Digital Library.

[83] T.J Todman,G.A Constantinides, et.al., “Reconfigurable computing: architectures and design methods,” IEE Proceedings - Computers and Digital Techniques,vol.152,no. 2,pp.193-207,2005, IET Digital Library.

[84] T.Akbari , Mohammad, “Approximated MILP model for AC transmission expansion planning: global solutions versus local solutions,” IET Generation, Transmission &amp Distribution,vol.10,no.7,pp.1563-1569,2016,IET Digital Library.

[85] Aljohani Nader, Cody Ruben, “MILP Model for Reliability Optimization In Active Distribution Networks,” 2018 North American Power Symposium (NAPS),2018.

[86] Macedo, V. Christian, et.al., “MILP branch flow model for concurrent AC multistage transmission expansion and reactive power planning with security constraints,” IET Generation, Transmission & Distribution,vol.10,no. 12,pp.3023-3032,2016,IET Digital Library.

[87] Ghasemi, Hosein,et.al., “A MILP Model for Integrated Expansion Planning of Multi-Carrier Active Energy Systems,” IET Generation Transmission & Distribution, IET Digital Library 2018, vol.13,no.7,pp. 1177 – 1189,2019.

[88] F.John, Franco, Marcos, et al., “A Mixed-Integer Linear Programming Model for the Electric Vehicle Charging Coordination Problem in Unbalanced Electrical Distribution Systems,” IEEE Transactions on Smart Grid ,vol.6,no. 5,pp.2200-2210,2015.

11 LIST OF ABBREVIATIONS:

CPU	Central Processing Unit
IFS	Integer Feasible Solutions
ILP	Integer Linear Programming
IP	Integer Programming
LP	Linear Programming
MILP	Mixed Integer Linear Programming
MINLP	Mixed-Integer Nonlinear Programming
MIP	Mixed Integer Programming
NLP	Nonlinear Programming
S-MILP	Stochastic Mixed Integer Linear Programming