Likelihood-Based Tree Search for Low Complexity Detection in Large MIMO Systems

Saksham Agarwal, Abhay Kumar Sah, and A. K. Chaturvedi, *Senior Member, IEEE*

***Abstract*—A recently reported result on large/massive multiple-input multiple-output (MIMO) detection shows the utility of the branch and bound (BB)-based tree search approach for this problem. We can consider strong branching for improving upon this approach. However, that will require the solution of a large number of quadratic programs (QPs). We propose a likelihood based branching criteria to reduce the number of QPs required to be solved. We combine this branching criteria with a node selection strategy to achieve a better error performance than the reported BB approach, that too at a lower computational complexity. Simulation results show that the proposed algorithm outperforms the available detection algorithms for large MIMO systems.**

***Index Terms*—Large MIMO, massive MIMO, branch and bound, integer programming.**

I. Introduction

With the advancement towards the upcoming 5G systems [1], the number of antennas in multiple-input multiple-output (MIMO) systems will be scaled up [2] to achieve higher data rates. Such large antenna systems are popularly referred to as large/massive MIMO systems. The detection of signals in such systems becomes challenging. This is because the existing near maximum likelihood (ML) detectors for conventional MIMO systems like the tree search algorithms in [3] and [4] are infeasible for large/massive MIMO systems owing to their increasing computational complexity with the number of antennas. On the other hand, the performance of low complexity detectors like minimum mean square error (MMSE) deteriorates with increasing number of antennas.

In the literature, various approaches like [5]–[9] have been proposed to address the issue of large MIMO detection. One recent approach [9] has formulated this problem as a mixed integer quadratic programming (MIQP) problem [10] and solved it using the standard branch and bound (BB) algorithm [11]. It has been shown that the BB based approach can perform significantly better than other existing approaches.

BB is a tree search based algorithm and requires strategies for branching and node selection. Among the various branching techniques available, strong branching (SB) is known to provide the optimal solution while doing minimal tree exploration [12]. However, BB using SB comes at a huge computational cost of solving $2N+1$ quadratic programs (QPs) at every node (N denotes the size of the problem) and turns out to be practically infeasible even for small MIMO systems. Out of the $2N+1$ QPs, only one is used towards obtaining the solution and the remaining $2N$ are required for deciding which variable to branch upon. If we can branch without solving QPs, the complexity of this approach can be drastically reduced.

In this letter, we address this issue by using a metric [13] for determining the likelihood of a symbol being in error. Thus, we propose to branch the tree at the index where the magnitude of the likelihood is maximum. As a result, at every node, instead of $2N$ QPs we need to solve only a single QP. We further propose a node selection strategy which explores the tree in a way such that a good performance can be obtained at a reasonable complexity. We verify this using simulations and show that this provides a better error performance than the recently reported BB based technique [9].

The rest of this letter is organized as follows. We describe the system model, problem formulation, and a review of BB algorithm in Section II. Section III proposes the likelihood based tree search algorithm and its complexity analysis has been discussed in Section IV. Simulation results are presented in Section V. Finally, we conclude this letter in Section VI. We use '$\mathbf{A}$' to denote matrices, '$\mathbf{a}$' to denote vectors, and 'a' to denote scalar quantities.

II. Preliminaries

We consider a MIMO system with N_t transmit and N_r receive antennas. The input-output relationship is given by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}}, \tag{1}$$

where $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{N_r}]^T \in \mathbb{C}^{N_r \times 1}$ is the received signal vector and $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{N_t}]^T \in \mathbb{C}^{N_t \times 1}$ is the transmitted signal vector where each $\tilde{x}_i$ belongs to an M-QAM square constellation $\tilde{\mathcal{X}}$. $\tilde{\mathbf{H}} \in \mathbb{C}^{N_r \times N_t}$ denotes the channel gain matrix with $\mathcal{CN}(0,1)$ elements and $\tilde{\mathbf{n}} \in \mathbb{C}^{N_r \times 1}$ denotes the complex additive white Gaussian noise (AWGN) vector with i.i.d. $\mathcal{CN}(0, \sigma^2)$ entries. This system model (1) is valid for both, single user point to point large MIMO systems, as well as multi user massive MIMO systems [14].

The system model in (1) can be expressed as an equivalent real-valued model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2}$$

Manuscript received March 27, 2017; accepted May 1, 2017. Date of publication May 9, 2017; date of current version August 21, 2017. The associate editor coordinating the review of this paper and approving it for publication was J. Wu. *(The authors contributed equally to this work.) (Corresponding author: Abhay Kumar Sah.)*

S. Agarwal and A. K. Sah are with the Department of Electrical Engineering, IIT Kanpur, Kanpur 208016, India (e-mail: saksham@iitk.ac.in; abhaysah@iitk.ac.in).

A. K. Chaturvedi is with the Department of Electrical Engineering, IIT Kanpur, Kanpur 208016, India and also with the Department of Electronics and Communication Engineering, IIT Roorkee, Roorkee 247667, India (e-mail: akc@iitk.ac.in).

Digital Object Identifier 10.1109/LWC.2017.2702639

where $\mathbf{y} = [\Re(\tilde{\mathbf{y}})^T \ \Im(\tilde{\mathbf{y}})^T]^T$, $\mathbf{x} = [\Re(\tilde{\mathbf{x}})^T \ \Im(\tilde{\mathbf{x}})^T]^T$, $\mathbf{n} = [\Re(\tilde{\mathbf{n}})^T \ \Im(\tilde{\mathbf{n}})^T]^T$ and

$$\mathbf{H} = \begin{bmatrix} \Re\left\{\tilde{\mathbf{H}}\right\} & -\Im\left\{\tilde{\mathbf{H}}\right\} \\ \Im\left\{\tilde{\mathbf{H}}\right\} & \Re\left\{\tilde{\mathbf{H}}\right\} \end{bmatrix}.$$

The objective of the optimal (ML) detection problem is to find the transmitted vector $\mathbf{x}^*$ which minimizes the Euclidean cost, that is

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}^{2N_t}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2, \tag{3}$$

where $\mathcal{X} \in \{-\sqrt{M}+1, \ldots, -1, 1, \ldots, \sqrt{M}-1\}$. As shown in [9], by using the linear transformation $z_i = \frac{x_i + (\sqrt{M}-1)}{2}$ $\forall i \in \{1, \ldots, 2N_t\}$, the problem can be converted to the following MIQP problem

$$\mathbf{z}^* = \underset{\mathbf{z}\in\Lambda^{2N_t}}{\operatorname{argmin}} \frac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{g}^T\mathbf{z}, \tag{4}$$

where $\Lambda \in \{0, 1, \ldots, \sqrt{M}-1\}$, $\mathbf{Q} = \mathbf{H}^T\mathbf{H}$ is a symmetric positive semi-definite matrix, $\mathbf{g} = -\mathbf{H}^T(\mathbf{y} + (\sqrt{M}-1)\mathbf{H}\,\underline{\mathbf{1}})/2$ and $\underline{\mathbf{a}}$ represents a column vector of all a's, having size $2N_t$. The problem in (4) can be solved optimally using the branch and bound (BB) algorithm [11]. We will briefly discuss it in the next subsection.

A. Review of Branch and Bound Algorithm

BB is a tree search based algorithm which recursively looks for the optimal solution in a reduced search space. It begins by relaxing the integer constraints in (4) as

$$\begin{aligned} &\underset{\mathbf{z}}{\operatorname{argmin}} \ \ \frac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{g}^T\mathbf{z} \\ &\text{subject to } \ \underline{\mathbf{0}} \le \mathbf{z} \le \left(\sqrt{M}-1\right)\underline{\mathbf{1}}. \end{aligned} \tag{5}$$

This relaxed problem is convex and can be solved using a standard QP solver. We consider this problem as the root node of the tree. If all the elements of the obtained solution vector are integers, it also solves (4) and thus is the required solution. Otherwise, the node branches into other nodes with reduced (but exhaustive in terms of integer points) and mutually exclusive search spaces.

Variable branching is one of the popular strategies used for generating reduced search spaces, as it does not create any extra constraints for the QP solver [12]. Among several variable branching techniques [11], strong branching (SB) is known to provide the solution with minimal tree exploration. It measures the increase in objective function by branching on every solution variable and then chooses the one which maximizes this increase. However, this comes at a cost of solving $4N_t$ QPs for branching itself, whereas only a single QP is required for obtaining the solution. Therefore, application of SB is infeasible for detection in large MIMO systems. On the other hand, the remaining low complexity branching techniques in [11] require a much larger tree exploration, which is again computationally prohibitive for large MIMO systems.

III. Likelihood Based Tree Search Algorithm

We can solve the problem with low complexity if we can avoid solving such high number of QPs for branching, while still retaining a reasonable error performance.

A. Error Likelihood Metric Based Variable Branching

Ideally, for the large MIMO detection problem, branching at a node should take place at the symbols which are in error. This would exclude the search space which would lead to an incorrect solution vector. However, we do not know which symbols are in error. Fortunately, recently a metric which computes the likelihood of a symbol being in error, especially in the context of large MIMO systems, has been reported in [13]. In the MIQP problem (4), the likelihood of the ith symbol being in error can be expressed as [13]

$$\eta_i = \frac{|g_i + \mathbf{q}_i\mathbf{z}|}{\sqrt{q_{ii}}}, \tag{6}$$

where q_{ii} and $\mathbf{q}_i$ are the (i, i) element and ith column of $\mathbf{Q}$, respectively, and g_i denotes the ith element of $\mathbf{g}$. Here η_i is the likelihood of z_i being in error, higher the value of η_i higher is the probability of z_i being in error. Let k be the index for which this likelihood metric is maximum. We branch upon the kth variable of a node S into two search spaces $S_k^{(1)}$ and $S_k^{(2)}$ as defined below

$$\begin{aligned} S &= \{\mathbf{z} : lb_k \le z_k \le ub_k \, \forall k \in \{1, 2, \ldots, 2N_t\}\}, \\ S_k^{(1)} &= S \cap \{\mathbf{z} : lb_k \le z_k \le \lfloor z_k \rfloor\}, \\ S_k^{(2)} &= S \cap \{\mathbf{z} : \lceil z_k \rceil \le z_k \le ub_k\}, \end{aligned}$$

where $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ denote the floor and ceiling functions while lb_k and ub_k are the lower and upper bounds respectively, on z_k.

This helps in searching the tree efficiently and hence reaching the solution with lesser exploration. Moreover, this branching strategy can be used without solving any QP, thereby greatly reducing the computational complexity and still providing a reasonable error performance.

B. Node Selection Strategy

Let us define a tree with depth d and breadth b as a (d, b) tree. For chosen values of d and b, the nodes having depth less than d are allowed to branch further in the tree. Building upon the idea in the previous subsection, we propose to branch upon b indices with the b largest values of the likelihood metric. Thus, during branching at any node, we create $2b$ new nodes with corresponding search spaces $S_k^{(1)}$ and $S_k^{(2)}$ with k being the indices corresponding to the b largest value of the likelihood metric. We would like to search this (d, b) tree such that after exploring the tree, we are likely to have the best possible solution within the tree. In view of this, we propose the following node selection and tree search strategy.

Assume at any depth i, there exist a set of candidate nodes $\{P_i^1, P_i^2, \ldots, P_i^n\}$, where subscript denotes the depth and superscript is used to index the candidate nodes. One of these nodes will be selected to branch further. We choose the node at which the corresponding solution vector has the least value of objective function (4) and denote it as P_i^*. This can

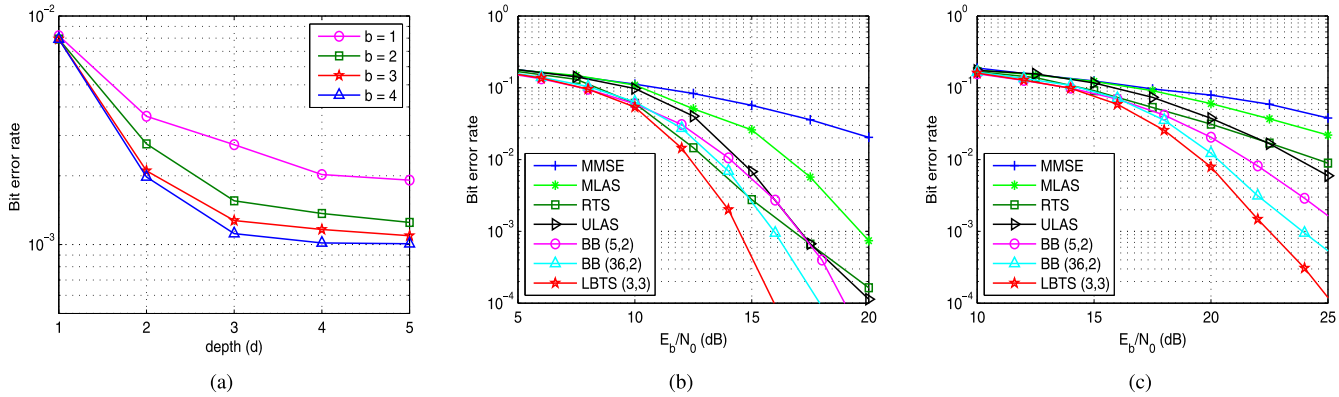

Fig. 1. BER of (a) a 32×32 16-QAM system at $E_b/N_0 = 15$ dB for different d and b values, (b) a 64×64 16-QAM system, and (c) a 32×32 64-QAM system.

Algorithm 1 Likelihood Based Tree Search

Parameters: d, b
Input: $\mathbf{Q}$, $\mathbf{g}$
Initialization: $List := \{P_0^1\}$, $i := 0$, $f^* := \infty$
while *List* is not empty **do**
  Find $P_i^* \in List$ with minimum $f(\mathbf{z}_{P_i^l})$ using (7)
  Keep P_i^* and prune all other nodes
  if $f(\mathbf{z}_{P_i^*}) < f^*$ **then**
    $f^* := f(\mathbf{z}_{P_i^*})$
    $\mathbf{z}^* := \mathbf{z}_{P_i^*}$
  end if
  if node P_i^* is at depth d **then**
    Delete P_i^*
    continue
  end if
  Compute $\eta_i := \frac{|g_i + \mathbf{q}_i \mathbf{z}|}{\sqrt{q_{ii}}}$ $\forall\, i \in \{1, 2, \cdots, 2N_t\}$
  Sort η_i's in descending order and rank corresponding indices
  for $j := 1{:}b$ **do**
    Choose index with rank j
    Branch P_i^* at the chosen index
    Store the newly created nodes as P_{i+1}^{2j-1} and P_{i+1}^{2j}
    Push P_{i+1}^{2j-1} and P_{i+1}^{2j} to *List*
  end for
  Delete P_i^*
  $i := i + 1$
end while
return $\mathbf{z}^*$

be mathematically expressed as

$$P_i^* = \underset{\{P_i^l \,|\, l \in 1, \ldots, n\}}{\operatorname{argmin}} \; f\left(\mathbf{z}_{P_i^l}\right), \tag{7}$$

where $f(\mathbf{r}) = \frac{1}{2}\mathbf{r}^T\mathbf{Q}\mathbf{r} + \mathbf{g}^T\mathbf{r}$ and $\mathbf{z}_{P_i^l}$ is the solution vector of node P_i^l. It may be noted that $\mathbf{z}_{P_i^l}$ is obtained by rounding off the elements of the solution of the relaxed problem (5).

For generating the set of candidate nodes at $(i+1)$th depth, we prune the existing nodes from the (d, b) tree except the node P_i^*, on which we further branch upon. After exploring the tree till depth d, the solution vector $\mathbf{z}^*$ is given by

$$\mathbf{z}^* = \underset{\{\mathbf{z}_k \,|\, k \in P_0^*, \ldots, P_d^*\}}{\operatorname{argmin}} \; f(\mathbf{z}_k). \tag{8}$$

Combining all the above, we summarize the pseudo-code in Algorithm 1, where $\mathbf{z}^*$ returns the desired solution. In the pseudo code P_0^1 represents the lone root node (initial candidate node) in the tree, which corresponds to the problem in (5). We refer to this as the likelihood based tree search (LBTS) algorithm. For a given (d, b) tree, we believe that this tree search strategy can give a good error performance at a reasonable complexity, for large MIMO systems. This has been corroborated using simulations in Section V.

IV. Complexity Analysis of LBTS

The computational complexity of LBTS depends upon the number of QPs required to be solved. We consider the interior point QP solver [15] which has $O(N_t^3)$ complexity per iteration. The advantage of using an interior point solver is that the number of iterations, for a given QAM size, is independent of N_t [16]. Thus, the overall complexity at a given node is of the same the order, i.e., $O(N_t^3)$. Out of approximately $(2b)^{d-1}$ nodes in a (d, b) tree, $d \times 2b$ nodes are explored and since only one QP is required to be solved at every node, the overall complexity is $O(dbN_t^3)$.

V. Simulation Results

In this section, we study the performance of the proposed LBTS algorithm for large MIMO systems of different sizes, considering a Rayleigh fading channel. We first evaluate the effect of d and b on the bit error rate (BER) performance. For this, we consider a 32×32 (i.e., $N_r = N_t = 32$) 16-QAM system and present the results for $E_b/N_0 = 15$ dB in Fig. 1(a). From the figure, it can be viewed that the performance saturates with increasing values of d and b. However, complexity increases linearly with $d \times b$ (explained in Section IV). In the sequel, we choose $d = 3$ and $b = 3$ as this seems to provide a reasonable trade-off between performance and complexity.

Next, we compare the performance for LBTS (d, b) with some recent algorithms for large MIMO detection. Thus, we choose ULAS [17], MLAS [5], and RTS [6], along with BB (l, m) [9] where l and m are simulation parameters. The results have been shown for a 64×64 16-QAM system and

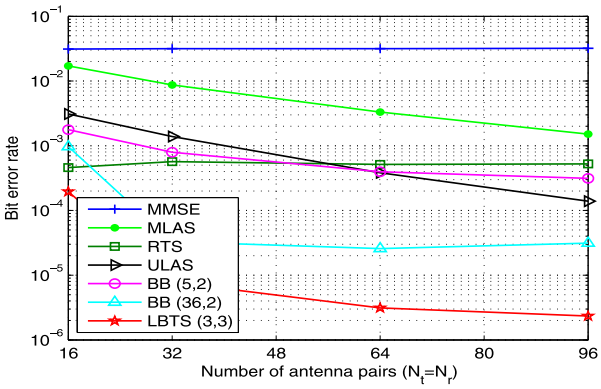

Fig. 2. BER of a 16-QAM at $E_b/N_0 = 18$ dB.

TABLE I
COMPARISON BASED ON ARITHMETIC OPERATIONS ($\times 10^5$) PER BIT

Detection Algorithms	32×32 64 QAM		64×64 16 QAM	
	12 dB	18 dB	8 dB	14 dB
MLAS	34.237	37.250	115.157	80.505
RTS	560.104	292.026	426.956	222.454
ULAS	0.213	0.199	2.993	1.309
BB $(5, 2)$	6.862	6.335	39.807	40.780
BB $(36, 2)$	49.580	50.667	298.033	306.465
LBTS $(3, 3)$	4.725	4.868	27.308	27.784

a 32×32 64-QAM system in Fig. 1(b) and Fig. 1(c), respectively. To start with, we choose the value of (l, m) such that the number of QPs required by BB (l, m) are same as for LBTS with $d = 3$ and $b = 3$. In other words, both have similar complexity. Hence, we choose the value of (l, m) as $(5, 2)$. We prefer this over other possible pairs like $(3, 3)$ and (4,2) having similar complexities, owing to its superior BER performance. From the figures one can observe that LBTS $(3, 3)$ outperforms all of the other algorithms and BB is its best competitor. At a BER of 10^{-3}, compared to BB $(5, 2)$, LBTS $(3, 3)$ has a gain of around 2.5 dB and 3.5 dB for 64×64 16-QAM system and 32×32 64-QAM system, respectively. Next, we allow BB (l, m) to explore more number of nodes (i.e., to solve more number of QPs) by selecting $l = 36$ and $m = 2$. It may be noted that the BER performance of BB (l, m) is more sensitive to l than m, and hence increasing the value of l has more payoff [9]. Even after allowing to solve a significantly higher number of QPs, i.e., 144 compared to 18 for LBTS $(3, 3)$, the latter has still a performance gain of around 1.5 dB, at the same BER of 10^{-3}.

Further, we examine the error performance for increasing number of antennas and show the corresponding BER curves in Fig. 2. From the figure, we can see that LBTS outperforms the rest of the algorithms by a large margin.

Lastly, in Table I, we compare the number of arithmetic operations for a 64×64 16-QAM system and a 32×32 64-QAM system. It can be observed that LBTS $(3, 3)$ requires lowest number of arithmetic operations after ULAS. As an illustration, for a 32×32 64-QAM system at $E_b/N_0 = 18$ dB, the savings in complexities are nearly 23%, 90%, 86%, and 98% compared to BB $(5, 2)$, BB $(36, 2)$, MLAS, and RTS, respectively. Only ULAS has a lower complexity, however it has an inferior BER performance (see Fig. 1(b) and Fig. 1(c)). It may be noted that the complexities of LBTS $(3, 3)$, BB $(5, 2)$, and BB $(36, 2)$ are directly proportional to the number of QPs solved viz. 18, 20, and 144 respectively.

VI. CONCLUSION

We have considered an MIQP based approach to propose LBTS - a low complexity tree search algorithm for detection in large MIMO systems. The algorithm is based on a novel branching strategy using a recently proposed error likelihood metric which helps in searching the tree more efficiently. We combine it with a node selection strategy to obtain a good trade-off between performance and complexity. Simulation results show that LBTS provides a better error performance, at a lower complexity, compared to existing algorithms for large MIMO systems.

REFERENCES

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.
[2] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.
[3] S. Han, T. Cui, and C. Tellambura, "Improved K-best sphere detection for uncoded and coded MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 1, no. 5, pp. 472–475, Oct. 2012.
[4] T. Cui, S. Han, and C. Tellambura, "Probability-distribution-based node pruning for sphere decoding," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1586–1596, May 2013.
[5] S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "A low-complexity near-ML performance achieving algorithm for large MIMO detection," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Toronto, ON, Canada, 2008, pp. 2012–2016.
[6] N. Srinidhi, T. Datta, A. Chockalingam, and B. S. Rajan, "Layered tabu search algorithm for large-MIMO detection and a lower bound on ML performance," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 2955–2963, Nov. 2011.
[7] X. Peng, W. Wu, J. Sun, and Y. Liu, "Sparsity-boosted detection for large MIMO systems," *IEEE Commun. Lett.*, vol. 19, no. 2, pp. 191–194, Feb. 2015.
[8] Q. Zhou and X. Ma, "Element-based lattice reduction algorithms for large MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 274–286, Feb. 2013.
[9] A. Elghariani and M. Zoltowski, "Low complexity detection algorithms in large-scale MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1689–1702, Mar. 2016.
[10] R. Lazimy, "Mixed-integer quadratic programming," *Math. Program.*, vol. 22, no. 1, pp. 332–349, Dec. 1982.
[11] J. Clausen, "Branch and bound algorithms-principles and examples," Dept. Comput. Sci., Univ. Copenhagen, Copenhagen, Denmark, Tech. Rep., 1999. [Online]. Available: http://www.imada.sdu.dk/~jbj/heuristikker/TSPtext.pdf
[12] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming*, vol. 271. Springer, Nov. 2014.
[13] A. K. Sah and A. K. Chaturvedi, "Reduced neighborhood search algorithms for low complexity detection in MIMO systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.
[14] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.
[15] J. Gondzio, "Interior point methods 25 years later," *Eur. J. Oper. Res.*, vol. 218, pp. 587–601, May 2012.
[16] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *J. Optim. Theory Appl.*, vol. 99, no. 3, pp. 723–757, Dec. 1998.
[17] A. K. Sah and A. K. Chaturvedi, "Beyond fixed neighborhood search in the likelihood ascent algorithm for MIMO systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.