Jurnal Kejuruteraan 36(6) 2024: 2313–2322
https://doi.org/10.17576/jkukm-2024-36(6)-06

Improving the Efficiency of Minimum Determinant Computation in Space Time Trellis Code with Optimal Subtree Pruning

(Meningkatkan Kecekapan Pencarian Penentu Terkecil dalam Kod Ruang Masa Trelis melalui Pencantasan Substruktur Optimal)

Ungku Azlin Ungku Chulan*, Norbahiah Misran & Mohammad Tariqul Islam

Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

**Corresponding author: proactive.azlin@gmail.com*

Received 14 December 2023, Received in revised form 27 July 2024
Accepted 25 September 2024, Available online 30 November 2024

ABSTRACT

The calculation of minimum determinant plays a crucial role in fulfilling the determinant criterion of a certain code design in space time trellis code. In the heuristic optimization of code construction, the minimum determinant is derived via a variant of the branch and bound algorithm. Although the algorithm is relatively efficient, it is not optimized in terms of the pruning strategy. Search space is pruned when the upper bound is exceeded. The upper bound restricts the area of the search space by acting as a minimize agent. No attempt is made on discerning the potential of different structures within the search space. This paper proposes a new pruning approach to improve the computational efficiency of finding the minimum determinant for a particular genera-tor matrix G. It builds upon the idea of minimal complete cycles. They are the smallest paths that begin and ends with zero. By capitalizing on the mini-mum complete cycle of the search tree, the structure with the highest potential in the search space can be identified. Consequently, it helps the search process to differentiate subtrees in their capacity of yielding a solution. Search can be focused on a certain subtree while others are pruned altogether. This enables approximately 45% reduction of the overall spatial and temporal cost. Despite its potential, the pruning method is inherently probabilistic. There is a 0.0357 risk that it could provide an erroneous minimum determinant.

Keywords: STTC; code design; determinant criterion; minimum determinant; search algorithm

INTRODUCTION

Data rate and transmission reliability are two of the most important factors in a communication system. To achieve a more reliable high data rate and low transmission error over the wireless channel, efficient modulation and coding schemes must be developed. However, this scheme has limitations due to issues such as multi-path fading, interference, and noise. To address this issue, the Space Time Trellis Code technique has been proposed.

The concept was introduced by (Tarokh et al. 1998) which are used in slow Rayleigh fading channel.In Multi-Input Multi-Output (MIMO) Rayleigh fading channels, Space Time Trellis Code (STTC) achieves both diversity and coding gains. A carefully designed STTC using a good design criteria helps to maximize both advantages.The design criteria normally consist of the rank and determinant criteria (RDC)(Tarokh et al.1998). By satisfying these two criteria, the process of code deisgn is optimized. The diversity gain is determine by the minimum rank of the distance matrices while minimum distance of the distance matrices is used to determine the coding gain.

The optimization of code design (Ata & Altunbas 2018; Harun et al. 2013) is one of most heavily researched topic in Space Time Trellis Code. There are many ways to optimize code design but it often revolves around the usage of the design criteria. The design criteria are practically a set of governing factors that determine the manner of which the performance of codes can be maximized under a particular situation. For instance, in the case of two transmitters and one receiver (2T-1R), the coding gain is optimal when the generator matrix G has full rank.

Early code design that uses rank criteria are (Banerjee & Agrawal 2014) and distance criteria (Viland et al. 2010). Other code design criteria that are being used is the determinant criterion. It requires the minimum determinant of the generator matrix G to be found. Finding the minimum determinant of a particular generator matrix G via the tree based approach is a practical way of complying with the determinant criterion in code design. The earliest tree based approach (Fukuda et al. 2006) relies on branch and bound (Jiang et al. 2018; Wang 2018) with the breadth first search as its traversal strategy. Here, pruning strategy is applied when the minimum determinant found at a node in the search tree exceeds the upper bound or largest value that is discovered previously during the search process. This is further improved with depth first search (Harun 2010) where lower search cost is made computationally feasible. Latest research which is the minimal cycle method (MCM) (Harun et al. 2013) improves the tree based approach by introducing a new way of calculating the initial upper bound (IUB).

The remainder of this paper is organized as follows: Methodology reviews the existing methods for computing minimum determinants in space-time trellis codes and also introduces the proposed Optimal Subtree Pruning (OSP) algorithm. Experimentation presents the experimental setup and compares the performance of OSP with MCM. Results and discussion discusses the practical implications of the findings and potential directions for future research. Finally, in conclusion it concludes the paper with a summary of the key contributions.

DOMINANCE

In heuristic search, the employment of dominance or dominant rules (Büyüktahtakın 2022, 2023; Khoudi & Berrichi 2020) is a compelling strategy in pruning. The strategy basically postulates the existence of a path p(i), of which superior or at least similar solutions can be found when compared to another path p(j). In this case, it is said that p(i) is dominant over p(j). To illustrate dominance, imagine the following example (Figure 1). Here, two paths are competing for dominance: p(i) = 0 → 1 and p(j) = 0 →2 where the solutions for both are (a) and (b) respectively. Now, the solutions or minimum determinants for (a) are 4, 4, 8 while (b) are 8, 16, 8. Given that smaller minimum determinants are better than larger ones, it is quite easy to see that the series of solutions in (a) from p(i) are superior or at least similar to the solutions in (b) from p(j). If this is so, then it can be surmised that p(i) is dominant over p(j).

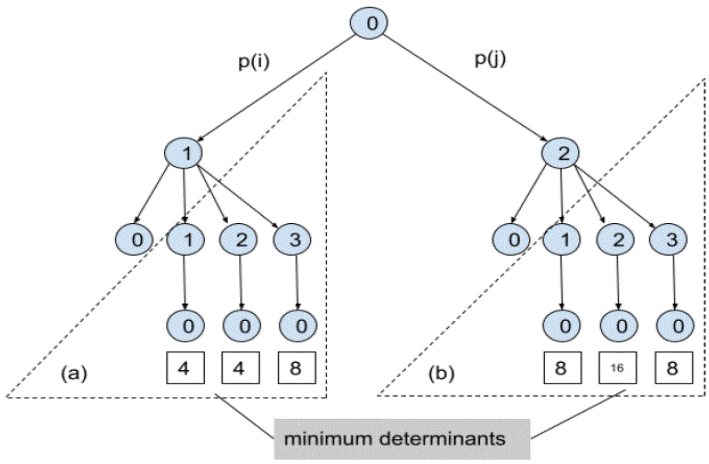

FIGURE 1. Dominance between path p(i) and p(j)

PROBLEM STATEMENT

Although MCM enables a tighter initial upper bound for the search process, the approach itself is not fully optimized. Further introspection would indicate that MCM does not discriminate between the different minimum complete cycles within the subtrees in considering their subsequent impact. That is, the method is not cognizant upon the potential of dominance in pruning the search space. Contemplating upon these minimum complete cycles, it is not implausible to assume that for a tree with different subtrees, there exists one that is dominant over the rest. Nevertheless, this possibility is ignored altogether by MCM. The pruning mechanism is unable to discriminate between varying subtrees within the search space that offer different potentials in yielding a solution for the minimum determinant (Figure 2). In light of this, the paper proposes a better pruning strategy.

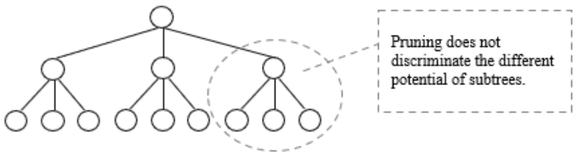

FIGURE 2. Weakness of pruning mechanism

METHODOLOGY

MINIMAL CYCLE METHOD (MCM)

In the tree based approach, the upper bound (UB) is initially set to infinity when the search starts. This implies that until the upper bound holds a finite value, the pruning mechanism of the tree based approach cannot be performed. As such, pruning takes time to commence because it requires the search to progress sufficiently before the upper bound can be updated with a valid value. To solve this problem, MCM would traverse the minimal complete cycles within the search tree prior to the actual search. These cycles mark the shortest path that begins and ends with zero. The minimum determinants are calculated at these cycles such that one of them can be chosen as the initial upper bound. This way, the initial upper bound is no longer set at infinity when the search is initiated. Instead, it contains a value that can be immediately used for pruning in the search process. This significantly reduces the computational complexity of the search.

SUBTREE PRUNE AND REGRAFT (SPR)

Among the pruning approaches, a rather intriguing method is the subtree prune and regraft (SPR), which is employed to determine phylogenetic relationships (Reichler et al. 2021) in molecular biology. By measuring the number of operations involved in transforming one version of a tree to another, the relationship between different trees is ascertained. SPR consists of three main steps: selection, pruning and attachment. For the sake of illustration, consider a simple example below (Figure 3).

1. Selection

   In this step, a subtree is selected. By convention, it is randomly performed (Atkins & McDiarmid 2019). In the example, the subtree from node 1 is selected.

2. Pruning

   Once the subtree is chosen, it is pruned from the tree. As shown, the subtree that is attached to node 1 is disconnected entirely from the tree.

3. Attachment

   The pruned subtree is attached to different parts of the tree. Here, the subtree from node 1 is attached to node 2. This creates a new variation of the tree.

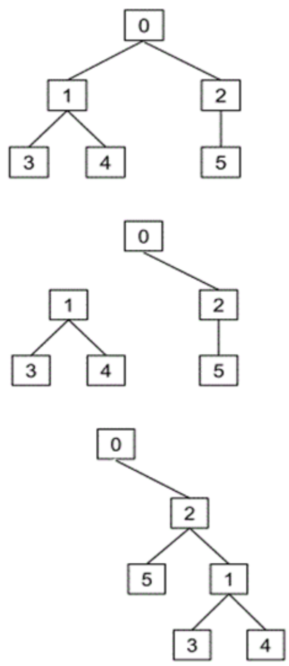

FIGURE 3. Selection, pruning and attachment in SPR

Partially inspired by SPR, the optimal subtree pruning (OSP) is devised. Generally speaking, OSP only applies the selection and pruning process. However, it must be noted that the selection process in OSP is not random but driven by the discovery made by the minimal cycle method (MCM). Furthermore, the pruning process in OSP is an inverse of the one in SPR. Instead of pruning the selected subtree, the other subtrees within the tree are removed to forge a new version of the search space. It is hypothesized that the optimal subtree, the one with the best minimal complete cycle, is dominant over the other subtrees.

In other words, the minimum determinant that can be found within the optimal subtree would be better, or at least as good as the ones in the other subtrees. Reasoning from this perspective, the other subtrees should be pruned altogether. This would allow an optimization of the search process that is far superior to MCM. There is still a caveat to the proposed approach. Given its heuristic nature, OSP is expected to exhibit a certain extent of acceptable error. Nevertheless, the complexity reduction that can be attained through OSP should outweigh the propensity of error as a whole.

OPTIMAL SUBTREE PRUNING (OSP)

The optimal subtree pruning (OSP) builds upon the idea in the minimal cycle method (MCM). In MCM, the minimum complete cycles are used to determine the initial upper bound of the search. OSP goes a step further by utilizing them to determine which subtree is optimal within the search space. The algorithm is shown in (Figure 4).

```
OSP(generator matrix G, total_state N)
    mcm          = minimal complete cycle
    md                  = minimum determinant
    ub                  = upper bound

    for i=1 to N-1
       mcm(i) = 0      i        0
    end
    ub = ∞

    for each mcm(i) in mcm(1) .. mcm(N-1)
        md(i) = minimum_determinant(mcm(i))
        if ( md(i) > 0 && md(i) < ub )
            ub = md(i)
        end
    end

    if ( ub != ∞ )
      for each mcm(i) in mcm(1) .. mcm(N-1)
          if(mcm(i) > iub)
            prune(subtree(mcm(i)))
          else
            ub = search(subtree(mcm(i)))
          end
      end
   else
      ub = search(tree(G,N))
   end
   return ub
end
```

FIGURE 4. Algorithm of optimal subtree pruning

The algorithm accepts the generator matrix G and total state N as inputs. Once these are received, the algorithm constructs the corresponding minimal complete cycles in accordance to the number of states N. If there are N states, then N-1 minimal complete cycles are generated. Similar to MCM, the minimum determinant at each of the cycle is calculated and then analyzed with one another. The minimal complete cycle with the smallest minimum determinant that is larger than zero is chosen as the representative. If none of the value fits this requirement, then the optimal subtree pruning cannot be enforced. On the other hand, if there are more than one minimal complete cycle with the smallest minimum determinant, then the initial one is given priority and chosen.

As for the pruning strategy (Krishnamoorthy 2015; Prasser et al. 2016), the algorithm nominates the subtree with the 'chosen' minimal complete cycle as the optimal subtree (Hirakawa et al. 2017). Thus, search would proceed only on this optimal subtree. For the other subtrees that are related to the non-chosen minimum determinant, they are pruned completely. In other words, the search mechanism would not even consider them.

To understand the approach better, observe the given illustration (Figure 5). Three minimal complete cycles are generated for the search tree of the 4 state 4x2 generator matrix G. The minimum determinant for each minimal complete cycle is 0.0, 8.0 and 4.0 respectively. Among them, the last complete cycle with the value of 4.00 is the one with the lowest non-zero value. It is therefore chosen as the minimal complete cycle. Not just that, the portion related to it becomes the optimal subtree. This implies that the other subtrees are deemed non-optimal. In effect, they are pruned completely. The search would not even consider other branches that are related to these non-optimal subtrees and would only continue at the optimal subtree.

To sum it up, OSP comprises of three main stages (Figure 6). The first stage is the generation stage which involves the generation of all minimal complete cycles within the search tree. This is followed by the selection stage that determines the optimal subtree by analyzing the minimal complete cycle. Finally, the process concludes with the pruning stage that ceases the search at all the non-optimal subtrees and proceeds only at the optimal subtree.

The pruning outcome is highly efficient. It avoids the need to traverse a considerable portion of the search space. However, this benefit often comes with a price. Theoretically speaking, there is a risk that the aggressive pruning method might interfere with the outcome for the search. Comparison between several research work is shown in Table 1.

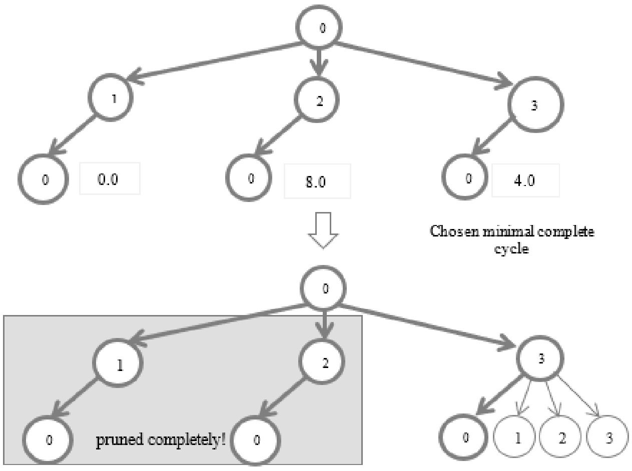

FIGURE 5. Conceptualization of Optimal Subtree Pruning

FIGURE 6. Main Stages of Optimal Subtree Pruning

TABLE 1. Comparison between several research work

Research work	Algorithm Basis	Computational Complexity	Minimum determinant accuracy	Error Probability Reduction	Practical Implications
Subtree Prune and Regraft	Selection, pruning and reattachment of subtrees	Moderate, not specifically designed for determinant computation	Moderate, random selection impacts accuracy	Moderate	Useful in Phylogenetic studies
Minimal Cycle Method	Uses minimal complete cycle to determine intial upper bound	High, due to non discrimitive pruning	High, but may include non optimal subtrees	High	Initial upper bound are defined before search
Optimal Subtree Pruning	Select and prunes all non optimal subtrees based on lowest minimum determinant value	Significantly reduced due to targeted pruning.	High, optimal subtree selection ensures accuracy	High	Enhance communication performance, energy efficiency, and real time implementation

EXPERIMENTATION

The experimentation is done on an intel i3 processor of 4GB RAM with Lubuntu as the OS. To conduct the experimentation, the entire 65536 variations of a 4 state 4x2 generator matrix are generated via the JAVA programming language. JAVA is used instead of the conventional Matlab due to its flexibility in forming the graph data structure. Matlab mainly relies on the array data structure and does not offer the same versatility.

The research uses communication model that involves 2 transmit antenna and 1 receiver antenna The total search space and temporal cost of searching for the minimum determinant between MCM and OSP is then compared (Fig. 7). Not just that, the minimum determinants between these two approaches are inspected as well to ascertain whether any discrepancies exist in terms of the search outcome.

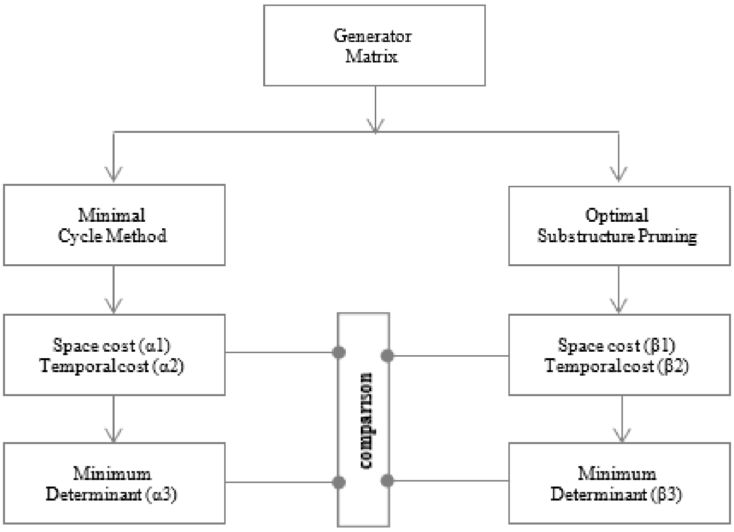

FIGURE 7. Comparison between minimal cycle method and optimal subtree pruning

The experimentation is also performed to analyze the risk of using OSP in calculating the minimum determinant. It is hypothesized that the approach could lack accuracy in certain cases.

RESULTS AND DISCUSSION

ANALYSIS OF SPATIAL AND TEMPORAL REDUCTION

The comparison between the minimal cycle method and optimal subtree pruning in terms of the spatial and temporal cost is shown in Table 2. From the spatial perspective, OSP reduces the spatial complexity of MCM by 45.61%. This is consistent with the temporal enhancement where OSP reduces the processing time of MCM by 47.61%. Apart from the degree of enhancement afforded by OSP, it is also crucial to analyze the breakdown between enhancement, neutrality and regression. This analysis practically explains how frequent OSP would improve or worsen MCM. As shown Table 3, OSP improves MCM by 0.9776 of the time and does not impact the search process or remains neutral at 0.0224 of the time. It must be noted that OSP does not worsen MCM in any of the instances of the experimentation. There is 0.0000 regression. Thus, there is no risk in using OSP.

TABLE 2. MCM vs OSP: Spatial and temporal analysis

Dimension	Total Generator Matrix	Minimal Cycle Method (MCM)	Optimal Subtree Pruning (OSP)	Reduction (%)
Spatial	65536	1662040 nodes	904000 nodes	45.61 %
Temporal	65536	4633 ms	2427 ms	47.61 %

TABLE 3. Breakdown of enhancement, neutrality and regression

	Enhancement	Neutrality	Regression
Probability	0.9776	0.0224	0.0000

ANALYSIS OF ENHANCEMENT

In addition to the former analysis, the extent of enhancement that is succeeded by OSP is further investigated. This is done by analyzing the average, maximum and minimum enhancement Table 4. On average, OSP enhances MCM by 45.77%. The maximum enhancement is 64.00% while the lowest one is 21.05%. Besides that, the probability distribution of the intensity for each of the enhancement that is exhibited by OSP when compared with MCM is studied as well Table 5.Intensity is defined as the measure of how frequently the new algorithm improves the original algorithm by a certain degree or range.Each row signifies the range of improvement displayed by OSP. As shown from the result, OSP improves MCM most frequently by 50.00% - 60.00% (Figure 8). This happens at 0.4312 of the time.The analysis on the improvement intensity gives a rather beneficial overview of how often a particular algorithm can accomplish a certain degree of quality in term of its spatial complexity. This provides a good indicator when estimating the overall performance of a particular enhancement with regard to its likeability of instigation.

TABLE 4. Average, maximum and minimum enhancement

Enhancement	Average	Maximum	Minimum
	45.77%	64.00%	21.05%

TABLE 5. Intensity of enhancement

	Enhancement Range	Total Search Enhanced	Probability
1	[00.00, 10.00)	0	0.0000
2	[10.00, 20.00)	0	0.0000
3	[20.00, 30.00)	6300	0.0983
4	[30.00, 40.00)	4728	0.0738
5	[40.00, 50.00)	22044	0.3441
6	[50.00, 60.00)	27624	0.4312
7	[60.00, 70.00)	3372	0.0526
8	[70.00, 80.00)	0	0.0000
9	[80.00, 90.00)	0	0.0000
10	[90.00, 100.00)	0	0.0000

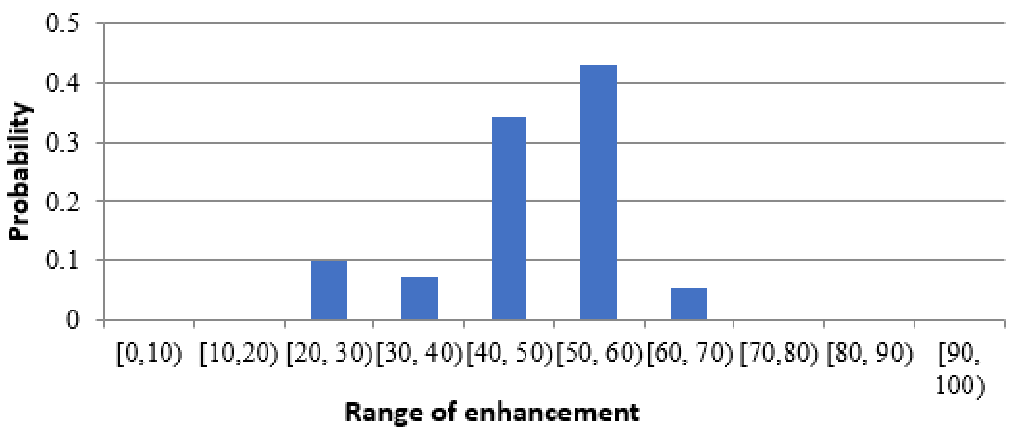

FIGURE 8. Intensity of enhancement

ANALYSIS OF FAILURE

Failure is defined as the erroneous minimum determinant that is given by the optimal subtree pruning. In other words, it transpires when pruning interferes with the ability of the search algorithm to find the minimum determinant accurately. From the 65536 generator matrices that are processed with the optimal subtree pruning, 2340 failures are detected. Thus, the risk of failure is quite marginal, only 0.0357. Failure is seen in the case of the generator matrix G = [3 3 0 1 ; 3 2 2 0]T where the optimal subtree pruning mistakenly states that the minimum determinant is 8.0 when in actuality, it is 4.0, as found by the minimal cycle method. This happens when the minimum determinant resides within the subtree that is perceived as non-optimal to the optimal subtree pruning. Since non-optimal subtrees are pruned entirely, the search process would never come across the right solution. To understand how it could happen, consider a hypothetical scenario below (Figure 9). The shaded nodes are those that are part of the minimal complete cycles. For the sake of argument, assume that OSP generates three minimal complete cycles and finds their minimum determinants to be 16.0, 12.0 and 20.0 respectively. Here, the minimal complete cycle with the least value lies in the middle part of the tree. As such, search only continues within this subtree while the others are pruned. In doing so, the minimum determinant of 8.0 is found.

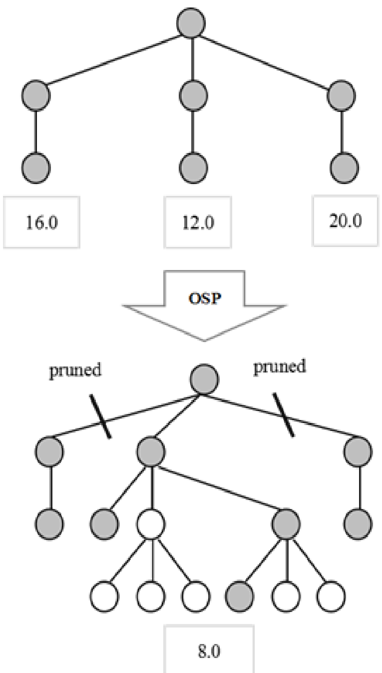

FIGURE 9. Failure of OSP

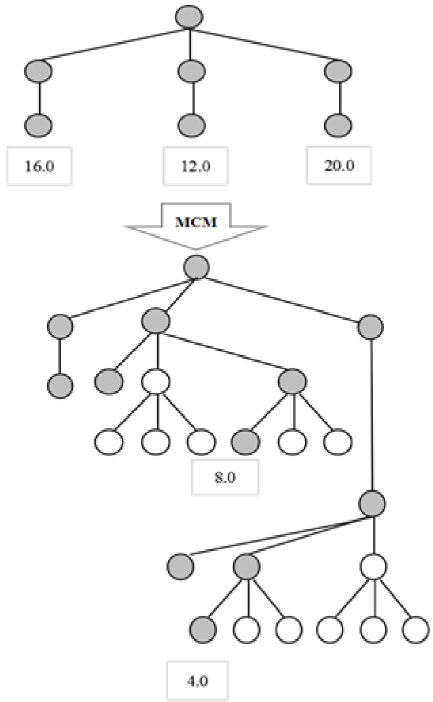

FIGURE 10. Success of MCM

Now, consider the same scenario but this time, it is handled by MCM instead (Figure 10). MCM does not enforce any subtree pruning. Therefore, all subtrees would be searched. This implies that the generation stage for MCM is similar to OSP where three minimal complete cycles are also generated. Again, the node with the minimum determinant of 8.0 is found by MCM. Here however, the search does not simply end. MCM finds a better value of 4.0 for the minimum determinant on a node that belongs to the subtree pruned by OSP.

PRACTICAL IMPLICATIONS AND FUTURE RESEARCH

The findings of this research demonstrate that the Optimal Subtree Pruning (OSP) method significantly enhances the efficiency of minimum determinant computation in space-time trellis codes. This improvement has several practical implications such as reducing error probability. By effectively pruning non-optimal subtrees, OSP reduces the error probability, leading to more reliable data transmission in wireless communication systems. This is particularly beneficial for applications in 5G networks and MIMO systems where high data rates and low error rates are critical.

The reduction in computational complexity and error probability translates to an improved SNR, enhancing overall communication quality.

Based on the findings of this paper, several potential directions for future research can be explored such as extension of OSP to different code structure. Investigating how OSP performs with various types of error-correcting codes can provide insights into its versatility and potential universal benefits.

Implementing the OSP algorithm in real-world communication systems and conducting extensive field tests can validate its practical benefits and identify potential areas for improvement. This includes testing in different frequency bands and under varying environmental conditions.

Another potential direction is the integration of OSP with machine learning algorithms to dynamically adapt the pruning strategy based on real-time data. This could lead to even more efficient and intelligent communication systems capable of self-optimizing their performance.

CONCLUSION

The Optimal Subtree Pruning (OSP) algorithm is designed to enhance computational efficiency by reducing the search space required for minimum determinant computation in space-time trellis codes. Here's how OSP impacts computing time:

1. Pruning Mechanism: OSP employs a selective pruning mechanism based on minimal complete cycles. By identifying and focusing on the optimal subtree, OSP eliminates unnecessary computations associated with non-optimal subtrees. This targeted approach significantly reduces the number of operations required, leading to faster computation times.
2. Comparative Reduction: OSP achieves a spatial complexity reduction of 45.61% and a temporal complexity reduction of 47.61% compared to the Minimal Cycle Method (MCM). These reductions translate to substantial savings in computing time, making OSP a more efficient algorithm for real-time applications.
3. Efficiency Metrics: The efficiency of OSP is measured by its ability to enhance the performance of MCM. OSP improves MCM's efficiency 97.76% of the time, with no instances of regression. This consistent performance improvement underscores OSP's effectiveness in reducing computing time across various scenarios. In spite of these enhancements, the approach suffers a 0.0357 risk of failure. In retrospect, the overall benefits outweigh the risk.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Kebangsaan Malaysia under grant number GUP-2022-017 for financial support.

DECLARATION OF COMPETING INTEREST

None.

REFERENCES

Ata, S.Ö., & Altunbas, I. 2018. STTC design for vehicular communication systems employing fixed-gain AF PLNC over cascaded fading channels. *IET Communications* 12(11): 1283-1289.

Atkins, R., & McDiarmid, C. 2019. Extremal distances for subtree transfer operations in binary trees. Annals of Combinatorics 23, 1-26.

Banerjee, S., & Agrawal, M. 2014. A simple analytical design approach to space time trellis code. *Wireless Personal Communication* 75: 1141-1154.

Büyüktahtakın, I.E. 2022. Stage-t scenario dominance for risk-averse multi-stage stochastic mixed-integer programs. *Annals of Operations Research* 309: 1-35.

Büyüktahtakın, I.E. 2023. Scenario-dominance to multi-stage stochastic lot-sizing and knapsack problems. *Computers and Operations Research* 153: 106149.

Fukuda, T., Otsu, S., Tokunaga, Y., & Zhao, H. 2008. A realization of determinant criterion for STTC design. ITC-CSCC 2008: The 23rd International Technical Conference on Circuits/Systems, Computers and Communications 61-64.

Harun, H. 2010. An improved algorithm for fast evaluation of Space Time Trellis Code (STTC) generator matrix, Doctoral Thesis, University Malaya.

Harun, H., Dimyati, K., & Ungku Chulan, U.A.I. 2013. Optimal generator matrix G. *Aerospace Science and Technology* 24(1): 136-140.

Harun, H., Ungku-Chulan, U.A.N., Ungku-Chulan, U.A.I., & Khazani, K. 2013. Improving the evaluation of generator matrix G by initial upper bound estimation. 2013 Computing, Communications and IT Applications Conference (ComComAp) 85-89.

Hirakawa, T., Tamaki, T., Kurita, T., Raytchev, B., Kaneda, K., Wang, C., Najman, L. 2017. Tree-wise discriminative subtree selection for texture image labeling. *IEEE Access* 5: 13617–13634.

Jiang, W., Wu, J., Qi, H., Feng, W., & Duan, X. 2018. An optimization method for evaluation and decision-making of distribution network planned projects based on branch and bound algorithm. 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) 1-9.

Khoudi, A., & Berrichi, A. 2020. Branch and bound algorithm for identical parallel machine scheduling problem to maximise system availability. *International Journal of Manufacturing Research* 15(3): 199-217.

Krishnamoorthy, S. 2015. Pruning strategies for mining high utility itemsets. *Expert Systems with Applications* 42(5): 2371-2381.

Prasser, F., Kohlmayer, F., & Kuhn, K.A. 2016. Efficient and effective pruning strategies for health data de-identification. *BMC Medical Informatics and Decision Making* 16: 49.

Reichler, S.J., Murphy, S.I., Martin, N.H., & Wiedmann, M. 2021. Identification, subtyping, and tracking of dairy spoilage-associated Pseudomonas by sequencing the ileS gene. *Journal of Dairy Science* 104(3): 2668-2683.

Tarokh, V., Seshadri, N., & Calderbank, A. 1998. Space-time codes for high data rate wireless communication, Performance criterion and code construction. *IEEE Transactions on Information Theory* 44(2): 744-765.

Viland, P., Zaharia, G., & Hélard, J.-F. 2010. Euclidean distance decomposition to generate new 16-QAM and 64-QAM space-time trellis codes. 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 385-390.

Wang, J.-Y. 2018. A branch-and-bound algorithm for minimizing the total tardiness of a three-agent scheduling problem considering the overlap effect and environmental protection. *IEEE Access* 7: 5106-5123.