A Zero-Suppressed Binary Decision Diagram Approach for Constrained Path Enumeration

Renzo Tan, Jun Kawahara, Agnes Garciano, and Immanuel Sin

***Abstract*—Combinatorial optimization over graphs has been the subject of research. Recently, the solution of such problems by enumeration using a compact data structure called the zero-suppressed binary decision diagram was proposed and studied. The paper augments the existing frontier-based search method of construction and puts forth a technique for accommodating additional constraints during computation. The shortest and longest path problems for the Osaka Metro transit network are simultaneously solved as demonstration. Furthermore, a comparison of the approach with a conventional integer programming method is presented towards justifying the effectiveness of the algorithm.**

***Index Terms*—zero-suppressed binary decision diagram, subgraph optimization, enumeration algorithm.**

I. Introduction

The zero-suppressed binary decision diagram [1] is a compact graph-based representation for an identified family of sets. In combinatorial optimization, therefore, feasible solutions to a problem may be stored in a zero-suppressed binary decision diagram economic of space [2]. Moreover, mathematical operations on families of sets may be performed on zero-suppressed binary decision diagrams as the composition is inherently recursive [3]. For example, the intersection and union of two families are easily computed through the said diagram representation. Since the zero-suppressed binary decision diagram contains all sets pertinent to a particular setup, the solution with the maximum or minimum aggregate weight, the number of feasible solutions, the mean and variance of the total weights of the solutions, and other information can also be obtained without manually extracting the set elements [3].

In the context of graphs, a zero-suppressed binary decision diagram can stand in for a collection of subgraphs [3]. The sets of edges that comprise each subgraph are used to distinguish the elements in the collection; thus, the nodes in the zero-suppressed binary decision diagram are consistent with the edges and not the vertices of the given graph. Segueing into construction, a technique for efficient zero-suppressed binary decision diagram creation is the frontier-based search [4]. The algorithm hinges on storing information in the nodes of the zero-suppressed binary decision diagram as it is being generated. Subgraph specifications may then be set in accord with the graph optimization problem. As a consequence, real world problems such as region partitioning for disaster evacuation [5], grid power loss minimization [6], and architectural floor planning [7], among others, were solved through the application of zero-suppressed binary decision diagram approaches.

In brief, the paper extends the frontier-based search algorithm for zero-suppressed binary decision diagram construction. This allows for the inclusion of a category-based constraint in addition to the customary degree constraint. By the proposed routine, one solves a constrained version of the shortest and longest s-t path problems for the Osaka Metro. To close, numerical experiments are done to show the success of the algorithm as opposed to standard methods.

An overview of the study is as follows. The second section provides a summary of the zero-suppressed binary decision diagram and the frontier-based search. Experiment results are in the third section, including a precursory investigation and the main contribution. A report on computational efficiency is also seen in the third section. The fourth section emphasizes the importance of the research and recommends directions for future work.

II. Preliminaries

A. Zero-Suppressed Binary Decision Diagram

A zero-suppressed binary decision diagram is a data structure that represents a family of sets over a finite universe $U = \left\{x_1, x_2, \ldots, x_{|U|}\right\}$ with ordered elements [1]. For the elements in U, $x_i < x_j$ if and only if $i < j$. More formally, the zero-suppressed binary decision diagram is a labeled directed acyclic graph satisfying: (1) There is only one node with indegree 0 called the root node; (2) there are exactly two nodes with outdegree 0 called the 0-terminal and 1-terminal; (3) each nonterminal node has exactly two outgoing arcs labeled by 0 and 1 and these are called the 0-arc and 1-arc, respectively; (4) for $j \in \{0, 1\}$, the destination node of the j-arc of a node n is called the j-child of n and is denoted by $c_j(n)$; (5) each nonterminal node n is labeled by an element of U; and (6) the label of a non-terminal node is strictly smaller than those of its children [1].

A subset U' of U corresponds to a path P from n to n' in a zero-suppressed binary decision diagram if and only if there exists a node n'' labeled with x whose 1-arc is in P for all $x \in U'$ [8]. The possibility of a zero-suppressed binary decision diagram representation of a family of sets $\mathcal{F}$ from U is built on the premise that a subset U' of U is in $\mathcal{F}$ if and only if there is a path from the root node to the 1-terminal in the diagram to which U' corresponds [8].

As an aside, a zero-suppressed binary decision diagram is defined to be reduced whenever: (1) There are no distinct nodes that have the same label, 0-child, and 1-child; and (2) there is no node whose 1-child is the 0-terminal [1]. A unique reduced zero-suppressed binary decision diagram containing the smallest possible number of nodes exists for

Manuscript revised April 5, 2019.

R. Tan and A. Garciano are with the School of Science and Engineering, Ateneo de Manila University, Philippines e-mail: {rrtan,agarciano}@ateneo.edu.

J. Kawahara is with the Division of Information Science, Nara Institute of Science and Technology, Japan e-mail: jkawahara@is.naist.jp.

I. Sin is with the John Gokongwei School of Management, Ateneo de Manila University, Philippines e-mail: isin@ateneo.edu.

a given family of sets $\mathcal{F}$ [1]. An algorithm for reducing a given zero-suppressed binary decision diagram in linear time is found in [3].

Zero-suppressed binary decision diagrams have proven to be capable of handling combinatorial optimization problems. Computing for the maximum and minimum total weights of sets in $\mathcal{F}$ and counting the number of solutions in $\mathcal{F}$, to name a few, may be done in time proportional to the number of nodes in the reduced zero-suppressed binary decision diagram representing it and not to the number of solutions [3]. Family algebra is also conveniently handled because of the recursive structure of the diagram [1].

B. Frontier-Based Search

The frontier-based search is an algorithm for constructing a zero-suppressed binary decision diagram representing a distinctively defined set of subgraphs in a given graph. Diagrams for subgraphs that are paths, trees, and matchings, among others, may be generated through the frontier-based search. A complete discussion is in [4]; nonetheless, an outline of the procedure is given.

Let $G = (V, E)$ be an undirected edge-weighted graph with vertex set V and edge set $E = \{e_1, e_2, \ldots, e_m\}$. Note that E is a collection of 2-element subsets of V, each of which corresponds to a unique edge in G. In other words, an edge $e \in E$ is equivalent to $\{v, w\}$, where $v, w \in V$. That G is simple and connected is also assumed. A subgraph of G is defined as (V', E') with $V' = \cup_{e \in E'} e$ for $E' \subseteq E$. In general, the union $\cup_{j=1}^{i} e_j$ is the set of vertices to which at least one of $e_1, e_2, \ldots, e_i$ is incident. This means that a subgraph has no vertex with degree 0 and that a subgraph is determined by its edge set. The set E with ordered elements $e_1 < e_2 < \cdots < e_m$ is the universe of the zero-suppressed binary decision diagram.

The algorithm starts with labeling the root node as e_1. The construction advances breadth-first, creating nodes e_{i+1} only after all nodes e_i have been generated for $i = 1, 2, \ldots, m-1$. The arcs of the nonterminal nodes e_i must point to nodes labeled e_{i+1}, the 0-terminal, or the 1-terminal. As each node n in the zero-suppressed binary decision diagram is being constructed through the frontier-based search, an array $n.\texttt{deg}$ mapping a particular subset of V to the set of natural numbers is stored into the node. Each path from the root node to n corresponds to a subgraph with vertex degrees specified by the array. If $n.\texttt{deg}$ is equal to $n'.\texttt{deg}$ for two nodes n and n' then the two nodes may be merged. This is defined as node sharing [4].

Let e_i be the edge of largest index among all edges incident to a vertex v. Since the degree of v is independent of edges $e_{i+1}, e_{i+2}, \ldots, e_m$, $\texttt{deg}[v]$ is no longer referred to after e_i is processed. One calls the node v fixed [4]. As convention, $F_0 = F_m = \emptyset$ and the ith frontier is defined as $F_i = \left(\cup_{j=1}^{i} e_j\right) \cap \left(\cup_{j=i+1}^{m} e_j\right)$ for $i = 1, 2, \ldots, m-1$ [4]. For node n with label e_i, one stores $n.\texttt{deg}[v]$ into the node if and only if $v \in F_{i-1}$. On a side note, pruning is also possible using the array $\texttt{deg}$ should there be violations in the degree conditions. Through node sharing and pruning, the zero-suppressed binary decision diagram representing subgraphs of a prescribed type can be created.

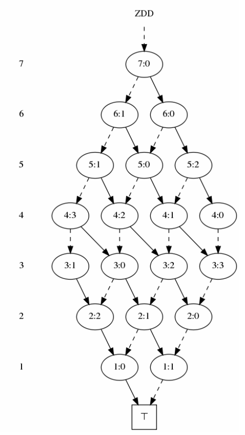

Fig. 1. The zero-suppressed binary decision diagram for the combination problem with $n = 7$ and $k = 3$.

III. Experiments

The results of implementation are discussed in this section. Common combinatorial problems such as the combination and knapsack problems are first used as setting for the utility of the zero-suppressed binary decision diagram. These serve as preview to the investigation process hired in solving the constrained shortest and longest s-t path problems for the Osaka Metro. The subsections correspond to a twofold treatment covering both general examples and a specific real-world case of larger scale.

Computer experiments were done in the C++ programming language with the aid of the TdZdd[1] library contributed primarily by H. Iwashita for the frontier-based search and family operations. The documentation may be read in [9] and [10]. Apart from the supplementary library, the entirety of the code used is committed to the ZDDLines[2] repository by R. Tan. Version 4.4.7 of the g++ was used as compiler. Concerning the machine, the operating system was the Linux CentOS 6.9, the central processing unit was the Intel® Core™ i7-5820K processor working at 3.30GHz, and the memory was 32GB.

A. Solving Combinatorial Problems

1) The Combination Problem: Determining the ways in which k objects can be selected from n distinct objects irrespective of order is known as the combination problem. The family of k-element subsets from a universe with n elements may be represented using a single zero-suppressed binary decision diagram. The diagram in Figure 1 is for the problem of finding all ways to choose exactly 3 items from a cardinality 7 item set.

In the diagram, all nodes on a level labeled by a variable number from 7 to 1 and a node identification number represent an item in the universe. The solid 1-arc and the dashed 0-arc correspond to the item being present or not, respectively. A path formed by 1-arcs and 0-arcs from the root node to the 1-terminal with symbol $\top$ is a set of 3 items from the n given.

[1] The link to the library is https://github.com/kunisura/TdZdd.

[2] The repository is in https://github.com/renzopereztan/ZddLines.

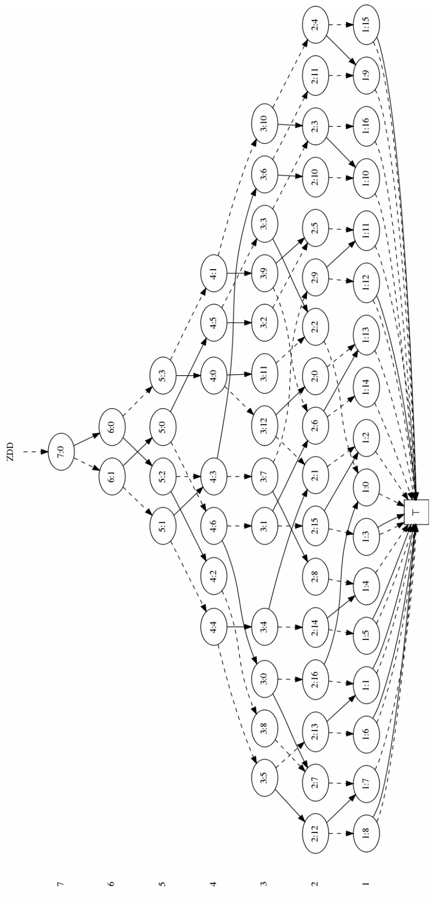

Fig. 2. The zero-suppressed binary decision diagram for the specified knapsack problem.

2) The Knapsack Problem: A familiar problem in combinatorics is the knapsack problem. Given a set of individually weighted items, the task is to resolve which subsets have total weights not exceeding a prescribed weight. These sets dictate which selection of objects may fill a fixed-size knapsack, from which the name derives. If one is to choose from $n = 7$ different items with assigned weight tuple $w = (4, 3, 7, 5, 6, 8, 10)$ and weight constraint $W = 18$, the rotated diagram is in Figure 2.

3) Taking the Intersection: A variant of the knapsack problem above may be solved as well. Suppose 7 items with weights defined by the same sequence $(4, 3, 7, 5, 6, 8, 10)$ are given. To add, the weight capacity of the knapsack is 18 and the restriction that precisely 3 items may be carried is imposed. Noticeably, the formulation simply puts together the constraints for the combination and knapsack problems. Solving the problem is tantamount to taking the intersection of the two diagrams previously produced. Figure 3 shows the resulting zero-suppressed binary decision diagram rotated. Should a value tuple $(7, 2, 8, 3, 6, 9, 5)$ be incorporated into the problem, computing for the 3-set with maximum total value and with weight not exceeding the limit is straightforward. The set formed by taking the first, fifth, and sixth items has total weight 18 and yields a value of 22 as maximum.

B. Finding the Shortest and Longest Paths

In this subsection, constrained variations of the shortest and longest s-t path problems are solved. One operates on the Osaka Metro[3] map of January 2019 and arbitrarily fixes the start and end stations. For the exposition, the Esaka Station of the Midosuji Line and the Kire-Uriwari Station of the Tanimachi Line are chosen. The main goal is to find the routes of minimum and maximum cumulative distance that use each of the lines of the metro network at least once. The two problems are proven to be nondeterministic polynomial-time hard [11].

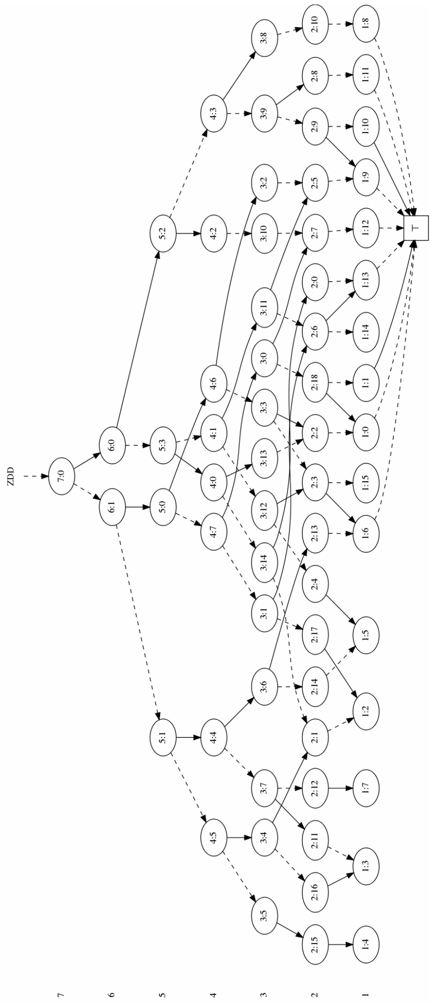

Fig. 3. The zero-suppressed binary decision diagram for the intersection of the combination and knapsack diagrams.

The solution begins with the construction of the zero-suppressed binary decision diagram for all s-t paths. This is done through the frontier-based search and the designation of a degree constraint. For both the initial and terminal vertices, a vertex degree of 1 is set. The remaining vertices in the network are forced to be of degree 2 or 0, guaranteeing subgraphs included in the diagram to be paths. For an idea of scale, the output zero-suppressed binary decision diagram has 5197 nodes and stores 15301 elements. The figure has been omitted for brevity as it is barely visible.

The constraint of having to use each line at least once is then addressed. Attached as appendix is the `Lines` class containing the method used to generate the zero-suppressed binary decision diagram that represents the aforementioned requirement. One takes the intersection of the diagram con-

[3] The route map is found in https://subway.osakametro.co.jp/en/guide/routemap.php.

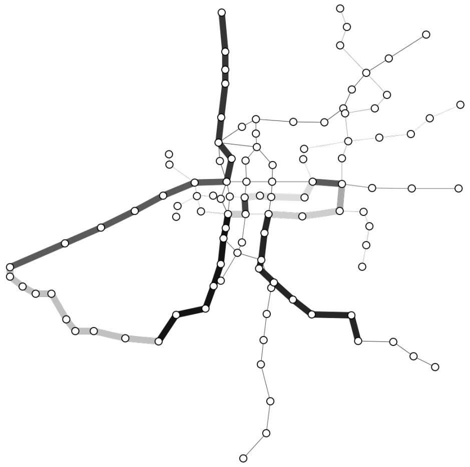

Fig. 4. The shortest constrained s-t path from Esaka to Kire-Uriwari.

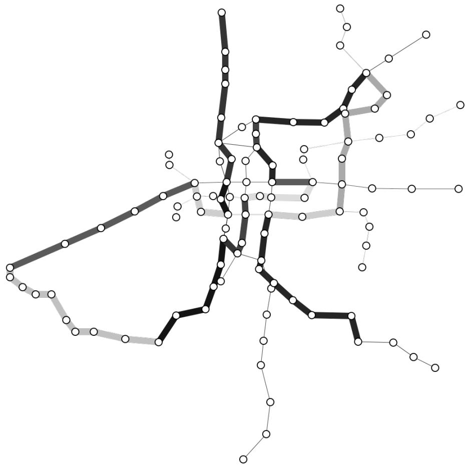

Fig. 5. The longest constrained s-t path from Esaka to Kire-Uriwari.

TABLE I
THE CONSTRAINED PATH WITH MINIMUM DISTANCE

Step	Initial Station	Terminal Station	Line
1	Esaka	Hommachi	M
2	Hommachi	Cosmosquare	C
3	Cosmosquare	Suminoekoen	P
4	Suminoekoen	Namba	Y
5	Namba	Nippombashi	S
6	Nippombashi	Nagahoribashi	K
7	Nagahoribashi	Morinomiya	N
8	Morinomiya	Midoribashi	C
9	Midoribashi	Imazato	I
10	Imazato	Tanimachi 9-Chome	S
11	Tanimachi 9-Chome	Kire-Uriwari	T

TABLE II
THE CONSTRAINED PATH WITH MAXIMUM DISTANCE

Step	Initial Station	Terminal Station	Line
1	Esaka	Hommachi	M
2	Hommachi	Namba	Y
3	Namba	Awaza	S
4	Awaza	Cosmosquare	C
5	Cosmosquare	Suminoekoen	P
6	Suminoekoen	Daikokucho	Y
7	Daikokucho	Dobutsuen-Mae	M
8	Dobutsuen-Mae	Nagahoribashi	K
9	Nagahoribashi	Morinomiya	N
10	Morinomiya	Tanimachi 4-Chome	C
11	Tanimachi 4-Chome	Minamimorimachi	T
12	Minamimorimachi	Tenjimbashisuji 6-Chome	K
13	Tenjimbashisuji 6-Chome	Taishibashi-Imaichi	T
14	Taishibashi-Imachi	Imazato	I
15	Imazato	Tanimachi 9-Chome	S
16	Tanimachi 9-Chome	Kire-Uriwari	T

taining all s-t paths with the diagram congruous with the given line-related constraint. The number of elements is narrowed down to 4029 with the resulting zero-suppressed binary decision diagram of 61102 nodes.

To produce the paths with minimum and maximum weight, the `MinDistData` and `MaxDistData` structures are utilized. The two are based on the algorithm for computing the maximum and minimum weighted sets in [3]. The shortest s-t path with total distance of 52.1 kilometers is illustrated in Figure 4. Further, the longest s-t path with total distance of 73.4 kilometers is shown in Figure 5. The computation time for the simultaneous solution of the shortest and longest s-t path problems using the zero-suppressed binary decision diagram approach is 0.04 seconds. Refer to Table I and Table II for more specific information on the steps in the shortest path and longest path, respectively. A condensed listing of the stations comprising the edges of the optimal paths are in the tables.

As comparison, the problem is posed as an integer programming problem. A formulation motivated by [12] and [13] was run on version 7.5.2 of the Gurobi Optimizer. Such a procedure, however, resulted in subgraph solutions with superfluous loops. This outcome necessitates adding a new constraint to exclude discovered loops and performing the process repeatedly until no loops remain. For instance, if the integer programming problem is solved and a solution including the loop $\{e_1, e_2, \ldots, e_k\}$ is obtained, a constraint prohibiting the use of all edges that compose the loop is added and the program with the added constraint is rerun. The method renders one unable to predict how many times the integer programming problem needs to be solved; accordingly, a solution is not guaranteed to be found.

IV. CONCLUSION

The study has demonstrated through evidence the relative superiority of a zero-suppressed binary decision diagram approach over an integer programming technique. Solutions to two nondeterministic polynomial-time hard problems were reached in a fraction of a second whereas a solution was not assured by using an integer program formulation. The possibility of enumeration of feasible solutions based on the zero-suppressed binary decision diagram is also an advantage.

With regard to future work, the implementation of the algorithm to a larger network is recommended. Solved integer programming problems in [12] for the Paris Metro rapid transit system may return a clearer comparison of the two methods. Lastly, new applications for the zero-suppressed binary decision diagram such as cluster analysis and computational graph theory are to be delved into.

APPENDIX

```
// The constraint of using each line at least once
class Lines: public tdzdd::DdSpec<Lines, int, 2> {
    // The number of edges is n
    // The number of lines is L
    // The lines each edge is part of is in l
    int const n, L;
    int const *l;
public:
    Lines(int n, int L, int *l)
        : n(n), L(L), l(l) {
        }
    int getRoot(int& state) const{
            state = 0;
            return n;
    }
    int getChild(int& state, int level, int value) const{
            if(value == 1) state |= (1 << l[n - level]);
                    level--;
            if(level == 0) {
                    if(state == ((1 << L) - 1) << 1) {
                            return -1;
                    } else{
                            return 0;
                            }
                    }
            return level;
    }
};
```

REFERENCES

[1] S. Minato, "Zero-suppressed BDDs for set manipulation in combinatorial problems," *Proceedings of the 30th International Design Automation Conference*, 1993.

[2] H. Iwashita, Y. Nakazawa, J. Kawahara, T. Uno, and S. Minato, "ZDD-based computation of the number of paths in a graph," *Hokkaido University Division of Computer Science TCS Technical Report TCS-TR-A-12-60*, 2012.

[3] D. Knuth, *The Art of Computer Programming Vol. 4 Fasc. 1*. Addison-Wesley, 2009.

[4] J. Kawahara, T. Inoue, H. Iwashita, and S. Minato, "Frontier-based search for enumerating all constrained subgraphs with compressed representation," *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences Vol. E100-A No. 9*, 2017.

[5] A. Takizawa, Y. Takechi, A. Ohta, N. Katoh, T. Inoue, T. Horiyama, J. Kawahara, and S. Minato, "Enumeration of region partitioning for evacuation planning based on ZDD," *Proceedings of the 11th International Symposium on Operations Research and Its Applications in Engineering, Technology, and Management*, 2013.

[6] T. Inoue, K. Takano, T. Watanabe, J. Kawahara, R. Yoshinaka, A. Kishimoto, K. Tsuda, S. Minato, and Y. Hayashi, "Distribution loss minimization with guaranteed error bound," *IEEE Transactions on Smart Grid Vol. 5 No. 1*, 2014.

[7] A. Takizawa, Y. Miyata, and N. Katoh, "Enumeration of floor plans based on a zero-suppressed binary decision diagram," *International Journal of Architectural Computing Iss. 1 Vol. 13*, 2015.

[8] S. Minato, "Zero-suppressed BDDs and their applications," *International Journal on Software Tools for Technology Transfer Vol. 3 No. 2*, 2001.

[9] H. Iwashita and S. Minato, "Efficient top-down ZDD construction techniques using recursive specifications," *Hokkaido University Division of Computer Science TCS Technical Report TCS-TR-A-13-69*, 2013.

[10] T. Toda, T. Saitoh, H. Iwashita, J. Kawahara, and S. Minato, "ZDDs and enumeration problems: State-of-the-art techniques and programming tool," *Computer Software Vol. 34 No. 3*, 2017.

[11] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

[12] F. Sikora, "The shortest way to visit all metro lines in paris," *arXiv Electronic Preprint arXiv:1709.05948v2*, 2017.

[13] R. Miyashiro, T. Kasai, and T. Matsui, "Strictly solving the longest one-way ticket problem," *Proceedings of the 2000 Fall National Conference of the Operations Research Society of Japan*, 2000.