Counting Magic Venn Diagrams

Anja Remshagen
Department of Computer Science
University of West Georgia
Carrollton, Georgia 30118, USA

Abstract – A Magic Venn Diagram is a magic figure where regions of a Venn diagram are labeled such that the sums of the regional labels of each set are the same. We have developed a backtracking search to count the number of Magic Venn Diagrams. The algorithm could determine the number of Magic Venn Diagrams for all Venn diagrams with four sets. This paper presents the algorithm with its applied heuristics and lists the computational results.

1. INTRODUCTION

For centuries magic figures, like magic squares and magic graphs, have been studied for mathematical recreation, but also to further mathematical knowledge (Andrews 2014, Block and Tavares 2009, Wallis 2001). Robinson (2016) has introduced a magic figure, called Magic Venn Diagram (MVD), where the nonempty regions of a Venn diagram are labeled such that the sums of the regional labels of each set are the same. MVDs are of general interest as many magic figures can be considered to be special cases of Magic Venn Diagrams.

In order to explore the structure of Magic Venn Diagrams, it is of interest to count the number of MVDs of a given Venn diagram. To the best of our knowledge, there has been no published work on MVDs so far, and MVDs have been counted by hand only. As the number of nonempty regions increases, counting MVDs by hand becomes unfeasible. We have developed and implemented an algorithm that determines and counts all MVDs of a given Venn diagram. The algorithm is based on backtracking search that applies some heuristics to reduce the search space. In our test, we could calculate the number of MVDs for several scenarios that have not yet been determined before.

2. MAGIC VENN DIAGRAMS

A *Venn diagram* of order n and size r is an n-tuple $(A_1, A_2, \dots, A_n)$ of distinct subsets $A_1, A_2, \dots, A_n$ of some set X such that there are exactly r nonempty Venn regions R_I of the form $R_I = (\bigcap_{i\in I} A_i) \cap (\bigcap_{i\notin I} A_i')$ where $I \subseteq \{1, 2, \dots, n\}$. A Venn diagram of order n is called *complete*, if its size is 2^n, that is, the collection of nonempty Venn regions includes region R_I for every subset $I \subseteq \{1, 2, \dots, n\}$.

A Venn diagram is *regular* of *degree* d if each set contains the same number d of regions.

A *labeling* of a regular Venn diagram of order n and size r assigns a distinct label $1, 2, \dots, r$ to each nonempty region. The *label sets* $S_i(L)$ of a labeling L are defined as

$$S_i(L) = \{l \mid L \text{ assigns } l \text{ to } R_I \text{ where } I \subseteq \{1, 2, \dots, n\} \wedge i \in I \wedge R_I \neq \emptyset\}$$

for each $i = 1, 2, \dots, n$. In other words, $S_i(L)$ is the set of regional labels of the subset A_i, $i = 1, 2, \dots, n$.

A given collection of label sets $S_1, \dots, S_n$ uniquely defines a labeling as follows. The region R_I is assigned the label l such that $\forall_{i \in I} l \in S_i \wedge \forall_{i \notin I} l \notin S_i$. Note that not any collection of n subsets of labels represents a labeling. In the following, we often represent a labeling by its label sets.

A *Magic Venn Diagram* (*MVD*) is a regular Venn diagram of order n and size r with a labeling L such that for some number m the sum $\sum\{l \mid l \in S_i(L)\} = m$ for each $i \in \{1, 2, \dots, n\}$. The sum m of labels in a label set is called the *magic sum* of the MVD.

Figure 1 displays a complete MVD of order 3, size 8 and degree 4. In the following, we often represent a region $R_I = (\bigcap_{i \in I} A_i) \cap (\bigcap_{i \notin I} A_i')$ by its subset $I \subseteq \{1, 2, \dots, n\}$. Then the region $\{1, 2\}$ of the MVD in Figure 1 is assigned label 7, for example. The magic sum of the MVD is 20.

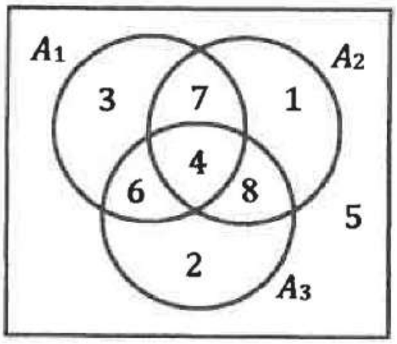

Figure 1: MVD of order 3, size 8 and degree 4

Figure 2 displays a MVD of order 4, size 10, and degree 5. The magic sum is 27. Note that the Venn diagram is not complete. For example, the regions {2}, {3}, {1, 4}, {2, 3}, {1, 2, 4}, and {2, 3, 4} are not labeled, meaning these regions are empty.

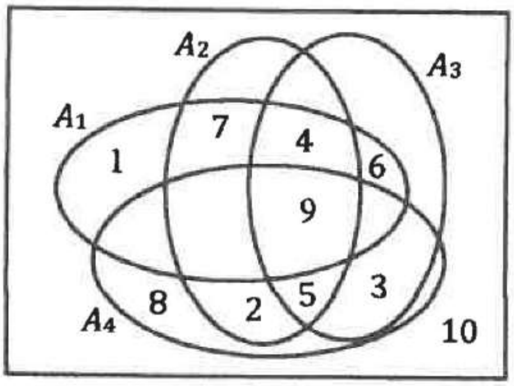

Figure 2: MVD of order 4, size 10, and degree 5

2.1 Magic Figures and Magic Venn Diagrams

Many common types of magic figures can be represented by MVDs. For example, a magic square is an arrangement of the first n^2 natural numbers in an $n \times n$ array such that every row, every column, and both diagonals have the same sum. A magic square can be represented by a MVD where each row, column, and diagonal is represented by a set A_i, $i = 1,2, \dots, n^2 + 2$. The nonempty regions are the corresponding intersections of rows, columns, and diagonals.

In an *edge-magic graph* defined by Kotzig and Rosa (1970), the n vertices and m edges $e_1, \dots, e_n$ of a graph are labeled by the numbers $1,2, \dots, r = n + m$ in such a way that the sum of the label of e_i and of the labels of the vertices incident to e_i is the same for every $i = 1,2, \dots, m$. An edge-magic graph can be represented by a MVD where each edge e_i corresponds to a set A_i, $i = 1,2, \dots, m$. For every edge e_i, region $R_{\{i\}}$ is nonempty, and for every vertex, region $R_{\{i_1,i_2,\dots,i_k\}}$ is nonempty where $e_{i_1}, e_{i_2}, \dots, e_{i_k}$ are the incident edges of that vertex.

2.2 Isomorphic MVDs

Two MVDs with labeling L an L', respectively, are *magically isomorphic*, or short *isomorphic*, if there is a bijection p from $\{1,2, \dots, n\}$ to $\{1,2, \dots, n\}$ such that the corresponding label sets are the same, that is, $\forall_{i=1,\dots,n} S_i(L) = S_{p(j)}(L')$.

Consider, for example, the MVDs displayed in Figure 2 and 3. Let's call the labeling of the MVD in Figure 2 L and the labeling of the MVD in Figure 3 L'. Assume the bijection p maps 1, 2, 3, 4 as follows: $1 \mapsto 4$, $2 \mapsto 3$, $3 \mapsto 2$, $4 \mapsto 1$. Then the label set $S_1(L) = \{1,4,6,7,9\}$ is the same as set $S_{p(1)}(L') = S_4(L')$. Similarly, it holds that $S_2(L) = S_{p(2)}(L') = S_3(L')$, $S_3(L) = S_{p(3)}(L') = S_2(L')$, and $S_4(L) = S_{p(4)}(L') = S_1(L')$.

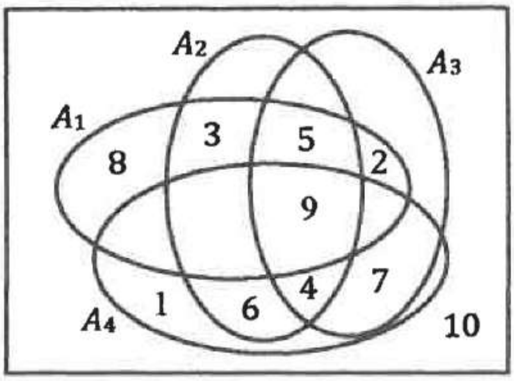

Figure 3: MVD of order 4, size 10, and degree 5

As most research is concerned with the structure of MVDs, we are interested in counting the number of non-isomorphic MVDs. We specify how the number of isomorphic and non-isomorphic MVDs relates to each other.

A *region-preserving* bijection of a Venn diagram of order n and with the subsets with the n -tuple $(A_1, A_2, \dots, A_n)$ is a bijection p from $\{1,2, \dots, n\}$ to $\{1,2, \dots, n\}$ such that for all $I \subseteq \{1,2, \dots, n\}$, the region $R_{\{p(i) | i \in I\}}$ is nonempty if and only if R_I is nonempty.

Let M be a Magic Venn Diagram with labeling L. Then for every region-preserving bijection p of the underlying Venn diagram of M, the labeling L' with label sets $S_i(L') = S_{p(i)}(L)$ specifies a MVD that is isomorphic to M. Similarly, assume M and M' with labeling L and L', respectively, are isomorphic MVDs and p is the bijection such that $\forall_{i=1,\dots,n} S_i(L) = S_{p(j)}(L')$ holds. Then p is a region-preserving bijection on the Venn diagram of M and M'. Thus, if there are k region-preserving bijections on a Venn diagram, then a MVD of that Venn diagram has exactly k isomorphic MVDs.

2.3 The Counting Problem

An instance of the *MVD Counting Problem (MVDC)* is a regular Venn diagram. The MVDC requires determining the number of non-isomorphic Magic Venn Diagrams of the given Venn diagram for every possible magic sum.

MVDC has been solved for all Venn diagrams of order 3. Robinson (2016) has reduced the regular Venn diagrams of order 4 to 19 different types. The solution of any MVDC instance can be derived from the solution of one of the 19 types, which are numbered 1, 2, 3, 4, 5, 6a, 6b, 7, 8, 9, 10, 11, 12a, 12b, 13, 14, 15, 16, 17. MVDC has been solved for the first eight types. See Robinson (2016) for details. Table 1 lists the types of Venn diagrams of order 4 for which MVDC has not been solved so far.

Type	Nonempty regions	Size	Degree	Factor
8	Ø, {1}, {2}, {4}, {1, 2}, {1, 3}, {1, 4}, {2, 3}, {3, 4}, {2, 3, 4}, {1, 2, 3, 4}	11	5	2
9	Ø, {1}, {2}, {3}, {4}, {1, 2}, {2, 3}, {3, 4}, {1, 2, 4}, {1, 3, 4}, {1, 2, 3, 4}	11	5	2
10	Ø, {1}, {2}, {3}, {4}, {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}, {1, 2, 3, 4}	12	5	24
11	Ø, {1}, {4}, {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}, {1, 2, 3}, {2, 3, 4}, {1, 2, 3, 4}	12	6	4
12a	Ø, {1}, {2}, {3}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4}	12	6	2
12b	Ø, {1}, {2}, {3}, {1, 2}, {1, 4}, {2, 4}, {3, 4}, {1, 2, 3}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4}	12	6	2
13	Ø, {1}, {2}, {3}, {4}, {1, 2}, {3, 4}, {1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4}	12	6	8
14	Ø, {1}, {2}, {3}, {4}, {1, 2}, {1, 3}, {2, 3}, {2, 4}, {3, 4}, {1, 2, 4}, {1, 3, 4}, {1, 2, 3, 4}	13	6	4
15	Ø, {1}, {2}, {3}, {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4}	14	7	6
16	Ø, {1}, {2}, {3}, {4}, {1, 2}, {1, 4}, {2, 3}, {3, 4}, {1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4}	14	7	8
17	every subset of {1, 2, 3, 4}	16	8	24

Table 1: Unsolved instances of MVDC of order 4

The second column specifies the nonempty regions of a Venn diagram of the given type. The last column with heading "Factor" lists the number of region-preserving bijections of the Venn diagram. This is the factor by which the number of isomorphic MVDs of a given Venn diagram differs from the number of non-isomorphic MVDs.

There have been some preliminary results by Yang Haifeng for the complete Venn diagram of order 4, which have been communicated in 2014 to David Robinson. Haifeng's results state that the total number of MVDs in this case is 53,860,800. However, the results have not been published, and it is not known if this number includes all MVDs or only non-isomorphic MVDs.

3. THE ALGORITHM

The solution algorithm is based on a backtracking search that traverses the labels from lowest to highest and assigns the current label to a region. If the algorithm detects that a current labeling cannot result in a MVD, the most recently assigned label is assigned to a different region. For details on backtracking search, see Russel and Norvig (2009). In our computational tests, we only track the number of MVDs for each magic sum, but the algorithm can also explicitly output all MVDs.

```
Collection<Region> regions;
Label[] labels;
Integer labelIndex;
Integer[] numberMVDs;

countMVDs(Collection regionSet, Collection labelSet) {
   regions = regionSet;
   for each region r or regions, mark r as unlabeled;
   labels = array with labels of labelSet in increasing order;
   labelIndex = 0;
   for each possible magic sum m, numberMVDs[m] = 0;
   countMVDsRecursively();
}

countMVDsRecursively() {
   if every region in regions is labeled {
      Integer m = magic sum of current MVD;
      numberMVDs[m]++;
   } else {
      Label l = labelArray[labelIndex];
      labelIndex++;
      for each unlabeled region r in regions {
         label r with l;
         if validLabeling() {
            countMVDsRecursively();
         }
         remove label l from r;
      }
      labelIndex--;
   }
}
```

Figure 4: The backtracking search

Figure 4 displays the pseudocode of the backtracking search. The entry point to run the algorithm is the function `countMVDs`. It is passed in a problem instance in the form of a collection of regions and labels. In this research, we only explore label sets of the form $\{1, 2, \ldots, r\}$. However, the algorithm can handle any sets of labels.

A naïve implementation of the function *`validLabeling()`* in the recursive procedure checks that the current labeling represents a MVD if all regions are labeled as show in Figure 5. This implementation will ensure that the solution algorithm correctly counts MVDs by their magic sum. However, the time complexity of this simplistic approach is $O(r!)$ where r is the size of the given Venn diagram. Hence, even for small problem instances of MVDC, the algorithm becomes impractical. We discuss several ways to improve the heuristic *`validLabeling()`* such that the search tree can be pruned before all labels are assigned.

```
validLabeling() {
   if not all regions are labeled or the labeling is a MVD {
      return true;
   } else {
      return false;
   }
}
```

Figure 5: Function `validLabeling`

3.1 Upper and Lower Bounds

The improved algorithm tracks, for each set A_i, a lower bound $\text{lb}(A_i)$ and an upper bound $\text{ub}(A_i)$ for its total possible sum of regional labels given the current partial labeling. In particular, the lower bound is the sum of the k labels currently assigned to k regions of subset A_i and of the d - k lowest labels that are not yet assigned to any region. Analogously, the upper bound for the label sum of subset A_i is the sum of the k currently assigned labels and the d - k highest labels that are not yet assigned to a region. Since the algorithm assigns labels from lowest to highest, the lowest labels $1, 2, \ldots, j$ are assigned to regions for some label $j = 1, 2, \ldots, r$. If $l_1, l_2, \ldots, l_k$ are the labels currently assigned to regions of A_i, then the lower bound for the label sum of subset A_i is calculated as

$$\text{lb}(A_i) = l_1 + l_2 + \cdots + l_k + (j + 1) + (j + 2) + \cdots + (j + d - k)$$

and the upper bound is

$$\text{ub}(A_i) = l_1 + l_2 + \cdots + l_k + \big(r - (d - k - 1)\big) + \cdots + (r - 1) + r.$$

The calculation of the lower and upper bound requires to add d - k consecutive labels. To speed up the calculation of the bounds, the algorithm determines all possible sums of consecutive labels in a preprocessing step and stores the sums in a two dimensional array where the value at index $[i][j]$ is the sum of the j consecutive labels starting with the label i.

Since the sum of regional labels of every subset A_i has to be the same in a MVD, the search tree can be pruned when the largest lower bound exceeds the smallest upper bound, i.e. when

$$\max\{\text{lb}(A_i)|i = 1, \dots, n\} > \min\{\text{ub}(A_i)|i = 1, \dots, n\}$$

holds where n is the order of the Venn diagram.

3.2 The Magic Median

We *complement* the labeling L of a Magic Venn Diagram of size r and degree d by replacing each label l with the label $r + 1 - l$. Let $\{l_1, l_2, \dots, l_d\}$ be the label set $S_i(L)$ of some subset A_i. If L is the labeling of a Magic Venn Diagram with magic sum m, then the sum of regional labels of A_i is

$$m = l_1 + l_2 + \dots + l_d$$

In turn, the sum of regional labels of A_i under the complemented labeling is

$$(r + 1 - l_1) + (r + 1 - l_2) + \dots + (r + 1 - l_d)$$
$$= d(r + 1) - (l_1 + l_2 + \dots + l_d) = d(r + 1) - m$$

Therefore, the complemented labeling is the labeling of a Magic Venn Diagram as well, and its magic sum is $d(r + 1) - m$. Since every MVD with magic sum m has a complement MVD with magic sum $d(r + 1) - m$, the median among all magic sums of a regular Venn diagram of size r and degree d is $d(r + 1)/2$. We call the median of all magic sums the *magic median*. The distribution of magic sums of a given regular Venn diagram is symmetric in the magic median. For instance, Table 2 displays the distribution of magic sums of the MVDC instance of Type 8 as specified in Table 1. Since the Type 8 Venn diagram has degree 5 and size 11, the magic median is $5(11 + 1)/2 = 30$.

Magic sum	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
#MVDs	6	54	118	224	358	424	664	666	652	666	664	424	358	224	118	54	6

Table 2: Distribution of magic sums of Type 8

Due to the symmetry, the algorithm only need to determine either the MVDs with a magic sum less than or equal to the magic median or with a magic sum greater than and equal to the magic median. Since the solution algorithm assigns the smallest labels first, the upper bounds $\text{ub}(A_i)$ are typically tighter than the lower bounds $\text{lb}(A_i)$. In turn, the condition that a current partial labeling results in a label sum less than the magic median can be typically detected earlier. Thus in order to prune the search tree as early as possible, the algorithm counts MVDs with magic sum greater than or equal to the magic median.

3.3 Standard Label Representation

The solution algorithm as proposed in Figure 4 counts all MVDs, not the number of non-isomorphic MVDs. As pointed out in Sections 2.2 and 2.3, the number of all MVDs and the number of non-isomorphic MVDs differs by a factor, which is the number of region-preserving bijections of the underlying Venn diagram. The numbers of non-isomorphic MVDs immediately result from the counts determined by the algorithm by dividing each count by the corresponding factor.

In case of the MVDC instances of Type 10 and 17, every bijection on $\{1,2,\dots,n\}$ is region-preserving. In this case, we can avoid counting isomorphic MVDs by defining a standard form of the labeling representation and by counting a labeling only if it is in that standard form. To define the standard form, we first define the lexicographical order on label sets as follows. Let $S = \{l_1, l_2, \dots, l_d\}$ with $l_1 < l_2 < \cdots < l_d$ and $S' = \{l'_1, l'_2, \dots, l'_d\}$ with $l'_1 < l'_2 < \cdots < l'_d$ be two label sets. Then

$$S < S' \leftrightarrow \exists_{k\in\{1,\dots,d\}} l_1 = l'_1 \wedge l_2 = l'_2 \wedge \dots \wedge l_{k-1} = l'_{k-1} \wedge l_k < l'_k$$

Since label sets represent the regional labels of a subsets A_i, each label set contains d labels where d is the order of the underlying Venn diagram.

A labeling L is in *standard form* if its label sets $S_1(L), S_2(L), \dots, S_n(L)$ are in lexicographical order, i.e. $S_1(L) < S_2(L) < \cdots < S_n(L)$.

If every bijection on $\{1,\dots,n\}$ is region-preserving, then any permutation of the label sets of a MVD represents an isomorphic MVD. In particular, in every group of isomorphic MVDs, there is exactly one MVD whose labeling is in standard form. For those MVDC instances, the algorithm counts a MVD only if its labeling is in standard form. In other words, as soon as a partial labeling violates the standard form, the search tree can be pruned.

Remember that the algorithm assigns labels from lowest to highest. Thus, the algorithm can detect as soon as a partial labeling cannot be extended to a labeling in standard form any more, allowing to prune the search tree early.

3.4 Pruning the search Tree

We summarize the pruning conditions. If one or both of the following conditions are true, the search tree is pruned:

- $\max\{ lb(A_i) \mid i=1,\dots,n\} > \min\{ ub(A_i) \mid i=1,\dots,n \}$
- $\min\{ ub(A_i) \mid i=1,\dots,n\} <$ magic median

In case that every bijection on $\{1,\dots,n\}$ is region-preserving, the tree is also pruned in the following case:

- the current partial labeling cannot be extended to a labeling in standard form

The check for the above conditions has been implemented in the function `validLabeling()` of the solution algorithm.

4. COMPUTATIONAL RESULTS

The solution algorithm has been implemented in the programming language Java 10. All problem instances were run on a desktop computer with an Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz processor and with the operating system Windows 10 Education.

Table 3 displays the run times in seconds for a MVD instance for the Types 8-17. The fourth column displays the run-times of the algorithm version that counts all isomorphic MVDs. The fifth column displays the run-times where only labelings in standard form are counted. This can only be applied to Type 10 and 17. In these two cases, the run-time is reduced by an order of magnitude. The last column displays the number of non-isomorphic MVDs.

Type	Size	Degree	CPU Time (sec) (no standard labeling)	CPU Time (sec) (standard labeling)	# MVDs
8	11	5	0.109375		2,840
9	11	5	0.062500		5,208
10	12	5	0.468750	0.046875	936
11	12	6	0.546875		11,760
12a	12	6	0.687500		32,304
12b	12	6	0.531250		19,600
13	12	6	0.656250		8,076
14	13	6	4.093750		82,232
15	14	7	34.812500		681,544
16	14	7	37.843750		565,536
17	16	8	4990.062500	258.671875	29,313,344

Table 3: Run-times for MVDC instances of order 4

Tables 4 - 14 display the counts of non-isomorphic MVDs by magic sum for each type. The counts are listed only for magic sums that are less than or equal to the magic median. It turns out that Haifeng solved the MVDC instance with an underlying Venn diagram of order 4 and size 15 where all regions are nonempty, except for the region $R_{\emptyset}$. His total count of 53,860,800 includes all MVDs.

Magic sum	22	23	24	25	26	27	28	29	30
# MVDs	3	27	59	112	179	212	332	333	326

Table 4: Non-isomorphic MVDs of Type 8 (magic median: 30)

Magic sum	21	22	23	24	25	26	27	28	29	30
# MVDs	4	30	94	149	260	303	420	501	578	530

Table 5: Non-isomorphic MVDs of Type 9 (magic median: 30)

Magic sum	25	26	27	28	29	30	31	32
# MVDs	6	22	28	54	64	85	97	112

Table 6: Non-isomorphic MVDs of Type 10 (magic median: 32.5)

Magic sum	30	31	32	33	34	35	36	37	38	39
# MVDs	3	48	103	420	407	820	876	1,240	1,071	1,784

Table 7: Non-isomorphic MVDs of Type 11 (magic median: 39)

Magic sum	30	31	32	33	34	35	36	37	38	39
# MVDs	64	288	424	1,184	1,156	2,276	2,304	3,496	2,920	4,080

Table 8: Non-isomorphic MVDs of Type 12a (magic median: 39)

Magic sum	29	30	31	32	33	34	35	36	37	38	39
# MVDs	34	92	232	376	488	1,008	1,390	1,500	3,848	1,924	1,328

Table 9: Non-isomorphic MVDs of Type 12b (magic median: 39)

Magic sum	28	29	30	31	32	33	34	35	36	37	38	39
# MVDs	6	26	68	136	165	330	330	534	590	773	634	892

Table 10: Non-isomorphic MVDs of Type 13 (magic median: 39)

Magic sum	31	32	33	34	35	36	37	38	39	40	41	42
# MVDs	36	222	535	1,128	2,048	2,881	4,088	5,174	6,024	7,325	7,869	7,572

Table 11: Non-isomorphic MVDs of Type 14 (magic median: 42)

Magic sum	40	41	42	43	44	45	46	47	48	49	50	51	52
# MVDs	40	426	1,784	3,994	8,034	13,488	21,462	29,490	38,530	45,952	54,676	59,830	59,830

Table 12: Non-isomorphic MVDs of Type 15 (magic median: 52.5)

Magic sum	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
# MVDs	12	204	812	2,064	4,170	6,866	10,524	14,816	19,862	25,496	31,008	35,800	40,652	44,294	46,188

Table 13: Non-isomorphic MVDs of Type 16 (magic median: 52.5)

Magic sum	51	52	53	54	55	56	57	58	59	60	61	62	63	64
# MVDs	327	3,506	11,457	34,568	70,930	148,256	230,288	381,686	503,732	829,020	903,036	1,203,986	1,371,212	1,843,868

Table 14: Non-isomorphic MVDs of Type 17 (magic median: 64)

5. SUMMARY AND CONCLUDING REMARKS

We have implemented an algorithm that determines the number of MVDs for all magic sums of a problem instance. The algorithm is based on backtracking search and applies several heuristics to reduce the search space. Computational results have been presented where the consecutive labels $1, 2, \ldots, r$ are assigned. The implementation can handle the more general problem where any set of labels is given, not just the labels $1, 2, \ldots, r$. The given number of labels may be greater than the number of nonempty regions. However, if the given labels are not consecutive values the distribution of the magic sums may not be symmetric anymore, and therefore the heuristic using the magic median cannot be applied.

The algorithm has solved all problem instances of MVDC of order 4. Among the instances of order 4, there are 11 types that have not been solved before. The run-time of the MVDC instance of size 16 exceeds an hour when counting all MVDs. Instances with larger size will become impractical to solve. We can likely improve the heuristics to reduce the search space further and to solve MVDC instances of slightly larger size effectively. However, it is expected that the number of MVDs of the MVDC instance with the complete Venn diagram of order 5 is too large to be explicitly enumerated by a single-thread algorithm.

6. REFERENCES

Andrews, W. S. 2004. *Magic Squares and Cubes.* Cosimo Classics.

Block, S. and S. Tavares. 2009. *Before Sudoku: The World of Magic Squares.* Oxford University Press.

A. Kotzig and A. Rosa. 1970. *Magic Valuations of Finite Graphs.* Canadian Mathematical Bulletin, Vol. 13(4), pp. 451-461. doi:10.4153/CMB-1970-084-1.

Robinson, D. G. 2016. *Magic Venn Diagrams.* Lecture Notes.

Russel S. J. and P. Norvig. 2009. *Artificial Intelligence: A Modern Approach.* Pearson.

Wallis, W. D. 2001. *Magic Graphs.* Springer.