Double-Wheel Graphs Are Graceful

Ronan Le Bras, Carla P. Gomes, and Bart Selman
Computer Science Department
Cornell University
{lebras,gomes,selman}@cs.cornell.edu

Abstract

We present the first polynomial time construction procedure for generating graceful double-wheel graphs. A graph is graceful if its vertices can be labeled with distinct integer values from $\{0, ..., e\}$, where e is the number of edges, such that each edge has a unique value corresponding to the absolute difference of its endpoints. Graceful graphs have a range of practical application domains, including in radio astronomy, X-ray crystallography, cryptography, and experimental design. Various families of graphs have been proven to be graceful, while others have only been conjectured to be. In particular, it has been conjectured that so-called double-wheel graphs are graceful. A double-wheel graph consists of two cycles of N nodes connected to a common hub. We prove this conjecture by providing the first construction for graceful double-wheel graphs, for any $N > 3$, using a framework that combines streamlined constraint reasoning with insights from human computation. We also use this framework to provide a polynomial time construction for diagonally ordered magic squares.

1 Introduction

In graph theory, graceful labelings and graceful graphs have been studied for over forty years, since their introduction by [Rosa, 1966]. Given a graph $G = (V, E)$, a graceful labeling assigns a unique integer in the range from 0 to $|E|$ to each vertex in V such that each edge is assigned a unique number from 1 through E corresponding to the absolute difference of the numbers of its endpoints. Figure 1 gives an example of a graceful labeling. A graph is said to be *graceful* if it admits a graceful labeling. Not all graphs are graceful. Intuitively speaking, the gracefulness property reveals a certain hidden regularity of the graph. More formally, a graceful labeling of a graph is defined as follows:

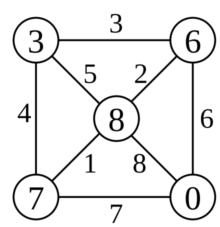

Figure 1: Example of a graceful graph. Each node is assigned a unique label from 0 to 8, where 8 is the number of edges. The label of an edge corresponds to the absolute difference of the labels of its endpoints and is unique among all 8 edges.

Definition (Graceful Labeling)**.** *Given a graph $G = (V, E)$, a labeling of the nodes $l : V \rightarrow \{0, ..., |E|\}$ is graceful if no two vertices share a label and each edge is uniquely identified by the absolute difference of the labels of its endpoints.*

Although graceful graphs are primarily theoretical structures in graph theory and discrete mathematics, they have a broad range of practical application domains, including in radio astronomy, X-ray crystallography, cryptography and experimental design. For example, in the area of radar pulse codes, a graceful labeling (or a semi-graceful labeling, in which the vertices might take values larger than $|E|$) of a complete graph can be used to accurately assess the elapsed time between the emission of a sequence of pulses and its reception. The graceful labeling of the graph then corresponds to the times at which the various pulses of the code should be emitted.

A major open research challenge in discrete mathematics is to characterize what families of graphs are graceful. This challenge is part of a larger effort to fully classify all types of graphs based on different properties. In particular, cycle-related graphs have received a lot of attention. For example, [Frucht, 1979] proved that a *wheel*, obtained when all vertices of a cycle are connected to a central vertex, is graceful. Moreover, a *gear*, obtained from a wheel by adding a vertex between every pair of adjacent vertices in the cycle, is also graceful [Ma and Feng, 1984]. Another family of graceful graphs is *helm* graphs, obtained by attaching a pendant edge at

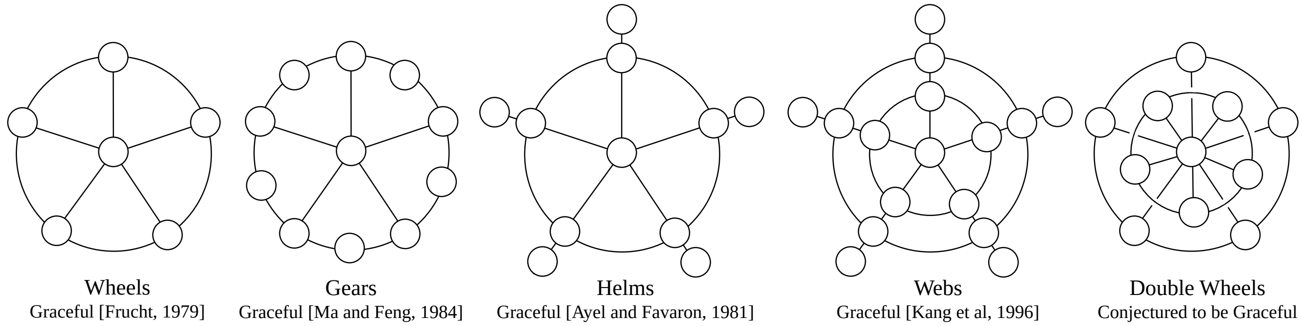

Figure 2: Some classes of graphs and their gracefulness.

each vertex of the cycle of a wheel [Ayel and Favaron, 1981]. Furthermore, a *web* graph, obtained by joining the pendant vertices of an helm to form a cycle and adding pendant vertices to the new cycle, is also graceful [Q.D. *et al.*, 1996]. Figure 2 illustrates these definitions of graphs. These families of graceful graphs, among others, appear in a dynamic survey, that maintains a comprehensive list of the various classes of graceful graphs [Gallian, 1998].

In addition, other families of graphs have been proven not to be graceful, such as cycles of size $n \equiv \{1, 2\} \pmod 4$ [Rosa, 1966] or double cones of size $n \equiv 2 \pmod 4$ [Redl, 2003]. Moreover, Rosa provides necessary conditions for a graph to be graceful based on the number of vertices and edges of the graph, as well as the parity of the degrees of the vertices [Rosa, 1966].

Finally, certain classes of graphs are conjectured to be graceful. In particular, a long-standing open problem in graph theory asks whether all trees are graceful. This problem is well-known as the Ringel–Kotzig conjecture [Bloom, 1979, e.g.] which, despite many attempts, remains unsolved. Another famous conjecture, known as the Truszczynski conjecture, hypothesizes that all unicyclic graphs (namely, graphs with exactly one cycle) are graceful [Truszczynski, 1984]. Finally, it has been conjectured that so-called double-wheel graphs are graceful [Heule and Walsh, 2010]. In this paper, we confirm this conjecture by providing the first constructive procedure for generating double-wheel graceful graphs of any size $N \geq 4$. This result also answers a research challenge posed in 2006 by Barbara Smith, in a panel discussion on "The Next Ten Years of CP," to use constraint programming techniques to prove or disprove the gracefulness of classes of graphs.

A double-wheel graph of size N is defined as follows:

Definition (Double-Wheel Graph). *A double-wheel graph DW_N of size N can be composed of $2C_N + K_1$, i.e. it consists of two cycles of size N, where the vertices of the two cycles are all connected to a common hub.*

Figure 2 shows a double-wheel graph of size 5. Finding graceful labelings for double-wheel graphs has become an interesting challenge problem for constraint solvers. In 2003, Petrie and Smith applied various symmetry breaking methods in constraint programming to graceful graphs and provided solution symmetries for the double-wheel graceful graph problem [Petrie and Smith, 2003]. They show that no labeling of DW_3 is graceful. In addition, they were able to find the number of non-isomorphic graceful labellings for sizes 4 and 5. In 2010, Heule and Walsh analyzed symmetries within solutions in order to boost the search for double-wheel graceful graphs [Heule and Walsh, 2010]. Their approach allowed them to find a graceful double-wheel graph of size 24, the largest reported in the literature so far. The constructive procedure we present in this paper provides graceful labelings for any double-wheel graph with $N \geq 4$. As we describe in the next section, the procedure was discovered using an new approach to constraint solving that incorporates a human computation component. By providing a constructive procedure for labeling double-wheel graphs of any size (≥ 4), we also go beyond standard constraint solving approaches, which generally provide solutions only for specific instances, each of a fixed size.

2 Framework

Constraint reasoning has been successfully applied to find graceful labellings for some particular graphs that were not known to be graceful [Lustig and Puget, 2001; Petrie and Smith, 2003; Smith, 2006; Benhamou *et al.*, 2007; Heule and Walsh, 2010]. In addition, this problem has extensively been studied for symmetry evaluation in constraint programming. For example, [Puget, 2006] proposes an efficent way to break all combinations of variable and value symmetries in graceful graphs. Moreover, [Gent *et al.*, 2005] introduce the concept of conditional symmetry breaking (namely, that occurs during the search itself) and evaluate it on graceful labelings. Furthermore, [Cohen *et al.*, 2005] explore different definitions of symmetry in the graceful labeling problem. Finally, constraint programming has also been used on closely related problems, such as the Golomb ruler [Galinier *et al.*, 2003].

In this work, we use the framework developed by [Le Bras *et al.*, 2012b]. This framework combines streamlining constraint reasoning with insights from human com-

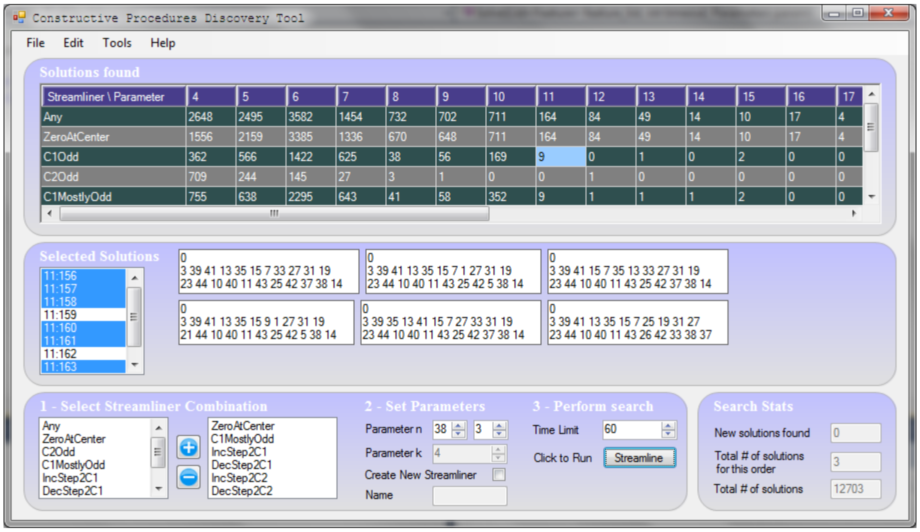

Figure 3: User interface for human-guided search to discover constructions for *graceful double-wheel graphs*. Each selected solution (middle panel) is a labeling of the double-wheel graph of the corresponding order ($N = 11$), where the three rows correspond to the labels of the center, the inner cycle, and the outter cycle, respectively.

putation in order to discover constructions for generating an entire family of combinatorial structures. Via this framework, Le Bras *et al.* discovered the first construction procedure for a combinatorial structure called spatially-balanced Latin square and provided a new lower bound for weak Schur numbers. In this work, we are interested in discovering whole families of graceful graphs.

Streamlining [Gomes and Sellmann, 2004] is an effective combinatorial search strategy that intentionally imposes additional structure to a combinatorial problem in order to focus the search on a highly structured subspace, therefore boosting constraint reasoning and propagation. (For a related search technique, called "tunneling," see [Kouril and Franco, 2005].) In other words, streamlining consists in adding specific desired regularities, such as a partial pattern of a solution, to the search engine, which then proceeds to search for solutions with these regularities. In any case, the effectiveness of streamlining relies on the quality of the suggested regularities.

In this work, we couple streamlining with a human computation component to identify possible patterns in solutions and suggest insightful regularities. This work is in part inspired by the exciting new area of human computation [Law and von Ahn, 2011], that concedes that humans still outperform fully automated approaches on certain tasks, especially those involving visual (pattern recognition) abilities.

The approach combines streamlining combinatorial search with human insights in a complementary and iterative approach, as follows. For a given combinatorial problem, we generate all solutions, or at least a significant fraction of them, for parameter sizes that are within the reach of traditional constraint reasoning techniques. In one or more of these solutions, one may be able to spot some regularity or partial pattern, which will be used to streamline the search. If the streamlined search does not give a larger size solution, the suggested regularity is likely accidental and we look for a new pattern in the smaller-size solutions. However, if the search succeeds, we now have larger size solutions that share some basic regularity. Furthermore, new regularities often reveal themselves at larger scales, and can be used to try and find yet larger solutions. Overall, the goal is to refine a set of regularities that generalizes to larger sizes within the user-allocated search time, until it fully characterizes combinatorial objects for a large number of parameter sizes.

Figures 3 and 6 illustrate the graphical user interface of the framework on the *graceful double-wheel graph problem* and the *diagonally-ordered magic squares* problem (defined in Section 4), respectively. The top panel of the user interface contains a table with the solutions found for each parameter size and each suggested streamliner. The middle panel allows the user to select, visualize and compare a subset of these solutions. The dialog window on the right of Fig. 6 illustrates how the user may specify new observed regularities. Finally, the bottom panel allows the user to select a subset of the suggested regularities, as well as the parameter sizes, for the streamlined search.

In the following, we describe how this framework was applied to the graceful double-wheel graph problem. Table 1 presents the various streamliners that were defined, as well as the largest solution size that was found with these streamliners. Within a 60-second time limit, standard constraint reasoning techniques were able to generate solutions of size up to 9. Enforcing that the inner cycle C_1 only contains odd numbers allows to generate solutions of size 11. Next, observing that many solutions

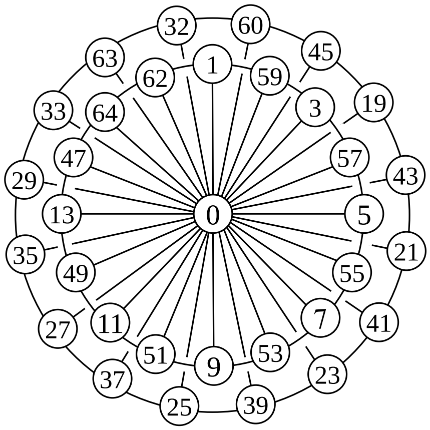

Figure 4: Double-wheel graph of size 16 (DW_{16}) with the graceful labeling generated by the proposed constructive procedure (case N is even).

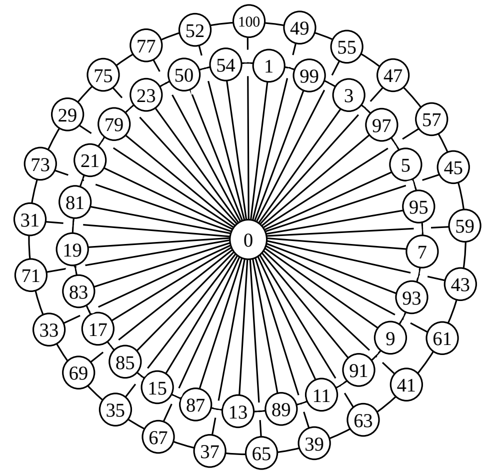

Figure 5: Double-wheel graph of size 25 (DW_{25}) with the graceful labeling generated by the proposed constructive procedure (case N is odd). This is the largest graceful double-wheel graph ever reported in the literature.

have 0 at the center of the wheels, we find solutions of size 19. At this point, some of the generated solutions partially exhibit the pattern named 'Inc. steps of 2', in which a sequence $(x, x+2, x+4, ...)$ appears on the nodes of a circle, skipping a node each time. For example, the inner circle in Figure 4 shows a sequence $(1, 3, 5, 7, ...)$ of length up to $N/2-1$, when looking at every other node, starting from the top and counting clockwise. Nevertheless, when we combine this pattern to the previously formulated streamliners, the search fails, as shown in Table 1. Therefore, we need to weaken the set of streamliners and we decide to require that most of the nodes, namely the first $N-2$ nodes, have an odd label. In that case, the search successfully generates solutions of size 21. Finally, we impose similar sequences (increasing and decreasing) on both circles, which leads to solutions of size 38.

Table 1: Size of the largest graceful double-wheel graphs generated by each set of streamliners. The number of solutions found appears in parenthesis. A 60-second timeout was used.

Set of streamliners	Size
$\Gamma_0 : \emptyset$	9 (251)
$\Gamma_1 : \{C_1 \textit{ is Odd}\}$	11 (8)
$\Gamma_2 : \{0 \textit{ at center}\}$	18 (1)
$\Gamma_3 : \Gamma_1 \cup \Gamma_2$	19 (6)
$\Gamma_4 : \Gamma_3 \cup \{\textit{Inc. steps of 2 in } C_1\}$	-
$\Gamma_5 : \Gamma_2 \cup \{\textit{Inc. steps of 2 in } C_1\}$ $\cup \{C_1 \textit{ Mostly-Odd}\}$	21 (5)
$\Gamma_6 : \Gamma_5 \cup \{\textit{Dec. steps of 2 in } C_1\}$	22 (2)
$\Gamma_7 : \Gamma_6 \cup \{\textit{Inc. steps of 2 in } C_2\}$	23 (1)
$\Gamma_8 : \Gamma_7 \cup \{\textit{Dec. steps of 2 in } C_2\}$	38 (2)

3 Construction for graceful DW_N

The constructive procedure that we propose is based on the various streamliners mentioned in the previous section. We formally define this construction as follows.

Let x_i be the label of vertex i on the first (inner) cycle, y_i the label of vertex i on the second (outer) cycle, and z the label of the central vertex. Our proposed construction distinguishes two cases, based on whether N is odd. In both cases, however, we set $z = 0$. The labels of the vertices on the two cycles are defined as follows:

Case N is even, $N \geq 4$:

$$x_i = \begin{cases} i & \text{if } i \text{ is odd, } 1 \leq i \leq N-3 \\ 4N-3-i & \text{if } i \text{ is even, } 2 \leq i \leq N-2 \\ 4N & \text{if } i = N-1 \\ 4N-2 & \text{if } i = N \end{cases}$$

$$y_i = \begin{cases} 4N-4 & \text{if } i = 1 \\ N+i & \text{if } i \text{ is odd, } 3 \leq i \leq N-3 \\ 3N-1-i & \text{if } i \text{ is even, } 2 \leq i \leq N-2 \\ 4N-1 & \text{if } i = N-1 \\ 2N & \text{if } i = N \end{cases}$$

Figure 4 illustrates this construction in the case where N is even. For the case where N is odd, a similar construction can be derived from the same set of streamliners. Figure 5 shows the graceful graph of size 25 obtained with this construction.

Case N is odd, $N \geq 5$:

$$x_i = \begin{cases} i & \text{if } i \text{ is odd, } 1 \leq i \leq N-2 \\ 4N+1-i & \text{if } i \text{ is even, } 2 \leq i \leq N-3 \\ 2N & \text{if } i = N-1 \\ 2N+4 & \text{if } i = N \end{cases}$$

$$y_i = \begin{cases} 4N & \text{if } i = 1 \\ 2N+2+i & \text{if } i \text{ is odd, } 3 \leq i \leq N-2 \\ 2N+1-i & \text{if } i \text{ is even, } 2 \leq i \leq N-3 \\ 3N+2 & \text{if } i = N-1 \\ 2N+2 & \text{if } i = N \end{cases}$$

We now prove the correctness of the construction in the case where N is even. Interestingly, it has been observed in [Le Bras *et al.*, 2012b] that the proof of the construction of so-called spatially-balanced Latin squares makes extensive use of the proposed streamliners (for a complete proof, see [Le Bras *et al.*, 2012a]). Similarly, the streamliners used to find the construction of graceful double-wheel graphs provide insights on how to prove the construction itself. In particular, the following proof makes extensive use of the parity of vertex labels. First, we prove that the vertices have different labels. Indeed, there are labels for all *odd* numbers from 1 to $(4N-1)$, except for $(N-1)$, $(N+1)$, $(2N-1)$ and $(4N-3)$, which gives $(2N-4)$ distinct odd vertex labels. In addition, there are 4 distinct *even* vertex labels, namely $2N$, $(4N-4)$, $(4N-2)$ and $4N$. Overall, all $2N+1$ vertex labels (including the center vertex) are distinct. Second, we prove that the $4N$ edge labels cover all values from 1 to $4N$. Given that the center is labeled 0, we have edge labels covering all *odd* numbers between 1 and $(4N-1)$, except for $(N-1)$, $(N+1)$, $(2N-1)$ and $(4N-3)$. These edge labels, however, appear on the edges (y_1, y_2), (x_{N-2}, x_{N-1}), (y_{N-1}, y_N), and (x_1, x_N) respectively. Regarding *even* edge labels, the inner circle have all even edge labels from $(4N-6)$ to $(2N+2)$ and the outer circle all edge labels from $(2N-6)$ to 4. Given the four even vertex labels, the only remaining even values to consider are $(2N-4)$ and $(2N-2)$. Nevertheless, these values appear as edge labels for (y_N, y_1) and (y_{N-2}, y_{N-1}), respectively. Therefore, all numbers from 1 to $4N$ appear as an edge label, which completes the proof. A similar proof can be derived for the case where N is odd.

4 Construction for *Diagonally-Ordered Magic Squares*

In this section, we show how we apply the same framework to the *diagonally-ordered magic square* problem and demonstrate that this framework is not application-specific. Diagonally-ordered magic squares have been introduced in [Gomes and Sellmann, 2004] to illustrate the efficiency of streamlined search. We first give a formal definition of a *diagonally-ordered magic square*.

Definition (Magic Squares)**.** *Given a natural number N, a magic square of order N is a square of size N containing all numbers from 1 to N^2 such that the sum of the numbers is the same in each row, each column, and each of the two main diagonals.*

Definition (Diagonally-Ordered Magic Squares)**.** *Given a natural number N, a magic square of order N is diagonally-ordered if both main diagonals, when traversed from left to right, have strictly increasing values.*

Given a natural number N, the *diagonally-ordered magic square* (DOMS) problem consists in the construction of a diagonally-ordered magic square of order N. Figure 7 depicts a DOMS of order 4.

In [Gomes and Sellmann, 2004], the authors first present a straightforward constraint programming model that allows to generate DOMSs of order up to 8 within 10 hours of computation. Nevertheless, within this time limit, their streamlined reasoning approach generates DOMSs of order up to 16, and even report finding DOMSs of order 17, 18, and 19. In addition, they mention that no polynomial-time construction is known for generating arbitrary large DOMSs.

1	**12**	**8**	**13**
15	**6**	**10**	**3**
14	**7**	**11**	**2**
4	**9**	**5**	**16**

Figure 7: Diagonally-ordered magic square of order 4. Each number from 1 to 16 appears exactly once, and each row, column and main diagonal adds up to the same value, namely 34. In addition, both main diagonals are ordered from left to right.

1	**5**	**9**	**13**
2	**6**	**10**	**14**
3	**7**	**11**	**15**
4	**8**	**12**	**16**

16	**12**	**8**	**4**
15	**11**	**7**	**3**
14	**10**	**6**	**2**
13	**9**	**5**	**1**

1	**12**	**8**	**13**
15	**6**	**10**	**3**
14	**7**	**11**	**2**
4	**9**	**5**	**16**

Figure 8: Observed regularity in a diagonally-ordered magic square of order 4. In the blue square (Left), the cells are numbered from 1 to 16, top-down and left-to-right. The red square (Middle) is numbered from 16 down to 1, in the same order. Overlapping both squares, keeping values from the first one for the cells on the two main diagonals, leads to a DOMS (Right).

Using the proposed approach, we are able to generate DOMSs of arbitrary large sizes. Figure 6 illustrates the graphical user interface for this framework applied to the DOMS problem.

A first construction was found by observing the following regularity. For some DOMSs generated with standard constraint reasoning techniques, the cell values are assigned the actual cell index, when counting top-down, left-to-right, or its complement to N^2+1. This pattern is depicted in Figure 8.

We then enforce this regularity. Namely, each cell (i, j) can only be assigned the value $i+(j-1)N$ or $N^2+1-i-(j-1)N$. For clarity purposes, as illustrated in Figure 8, we will call *blue cell* a cell (i, j) taking value $i+(j-1)N$, and *red cell* a cell (i, j) taking value $N^2+1-i-(j-1)N$. At this point, a pattern emerges for orders N such that $N \equiv 0 \pmod 4$, which consists in tiling the 4×4 pattern of Figure 8 to cover the $N \times N$ square, and leads to the following construction, in which $a(i, j)$ represents the value of cell (i, j):

Case $N \equiv 0 \pmod 4$**:**

$$a(i,j) = \begin{cases} i + (j-1)N & \text{if } k = l \text{ or } k = 4 - l, \\ N^2 + 1 - i - (j-1)N & \text{otherwise} \end{cases}$$

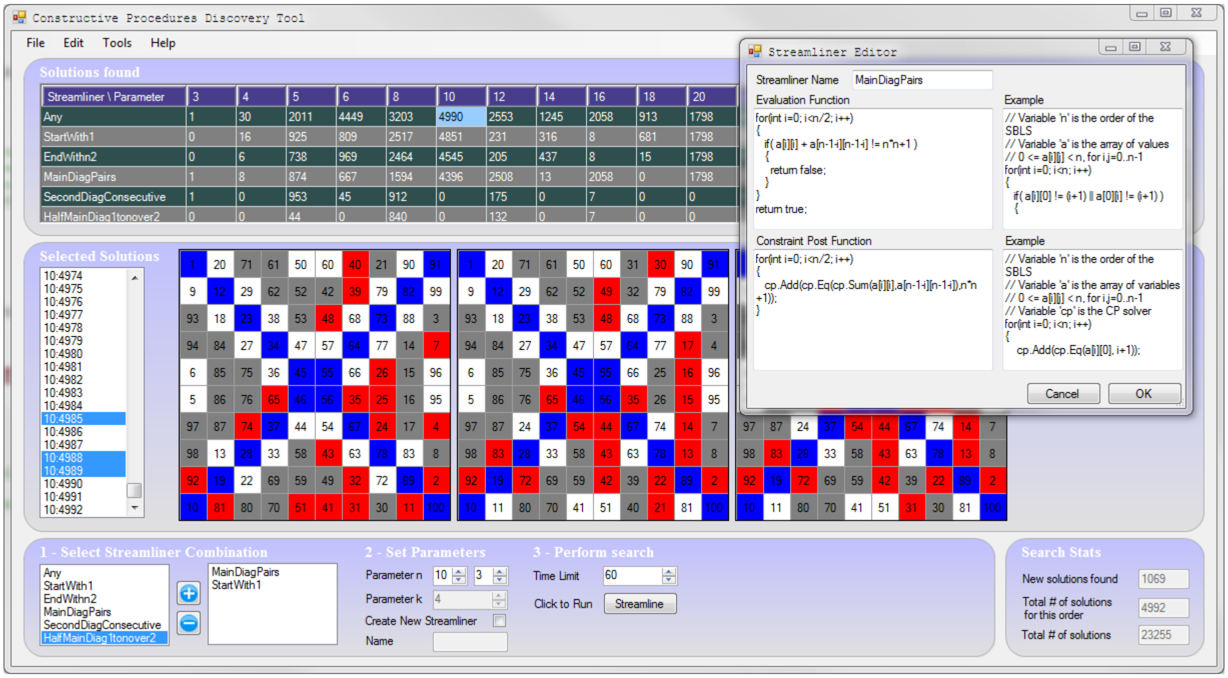

Figure 6: Graphical user interface for human-guided combinatorial search to discover constructions for the *diagonally-ordered magic squares* problem.

where $k = (i-1) \pmod 4$ and $l = (j-1) \pmod 4$.

This first construction of infinite number of DOMSs is in fact a variation of the so-called Durer method for constructing magic squares of doubly-even order [Pickover, 2001, e.g].

Nevertheless, this construction does not apply to either odd orders or singly-even orders. In fact, the streamliner that requires each cell to be either blue or red does not allow to generate DOMSs of order N for any $N \not\equiv 0 \pmod 4$.

In order to find a construction for singly-even DOMSs (i.e, of order $N \equiv 2 \pmod 4$), we relax the previous streamliner and also allow a cell (i,j) to take the value $(N+1-i)+(j-1)N$ (*white* cell) or its complement to N^2+1 (*gray* cell). These cell values correspond to counting the cells bottom-up and from left-to-right, and top-down from right-to-left, respectively. Any solution found with this new streamliner will only have blue, red, white, or gray cells. This feature particularly emphasizes the importance of an appropriate visualization, that allows the user to effectively identify potential patterns in the solutions. In addition, as suggested in some solutions, we define a streamliner that requires that the top-left quadrant contains no red cells. Finally, observing the solutions obtained by combining these streamliners, we also hypothesize that the diagonals of the top-left quadrant are monochromatic.

Although this set of streamliners is not an example of a fully-complete construction for singly-even DOMSs, we believe it provides interesting insights to solve the diagonally-ordered magic square problem for singly-even orders. Indeed, while [Gomes and Sellmann, 2004] report taking about two days to generate a DOMS of order 18, we were able, with this set of streamliners, to generate DOMSs of order up to 18 in seconds.

5 Conclusions

We provide the first constructive procedure for generating graceful labeling for arbitrary large double-wheel graphs. No such procedure was previously known and the largest double-wheel graph known to admit a graceful labeling was of size 24.

This construction was found through a general framework devised to discover efficient, constructive procedures for generating classes of complex combinatorial objects. This framework combines specialized search techniques with a human computation component, through a complementary, iterative approach. We also used this framework to provide a polynomial time construction for diagonally-ordered magic squares.

The integration of combinatorial reasoning and human computation methods is an exciting research direction, and future avenues include, for example, crowdsourcing the identification of patterns within a pool of solutions.

Acknowledgments

This work was supported by the National Science Foundation (NSF Expeditions in Computing award for Computational Sustainability, grant 0832782).

References

[Ayel and Favaron, 1981] J. Ayel and O. Favaron. *The helms are graceful*. Université Paris-Sud, Centre d'Orsay, Laboratoire de recherche en Informatique, 1981.

[Benhamou *et al.*, 2007] F. Benhamou, N. Jussien, B. O'Sullivan, et al. *Trends in Constraint Programming*. Wiley Online Library, 2007.

[Bloom, 1979] Gary S Bloom. A chronology of the ringle-kotzig conjecture and the continuing quest to call all trees graceful. *Annals of the New York Academy of Sciences*, 328(1):32–51, 1979.

[Cohen *et al.*, 2005] D. Cohen, P. Jeavons, C. Jefferson, K. Petrie, and B. Smith. Symmetry definitions for constraint satisfaction problems. *Principles and Practice of Constraint Programming-CP 2005*, pages 17–31, 2005.

[Frucht, 1979] R. Frucht. Graceful numbering of wheels and related graphs. *Annals of the New York Academy of Sciences*, 319(1):219–229, 1979.

[Galinier *et al.*, 2003] Philippe Galinier, Brigitte Jaumard, Rodrigo Morales, and Gilles Pesant. *A constraint-based approach to the Golomb ruler problem*. Montréal: Centre for Research on Transportation, Centre de recherche sur les transports (CRT), 2003.

[Gallian, 1998] J.A. Gallian. A dynamic survey of graph labeling. *Electron. J. Combin*, 5(1):43, 1998.

[Gent *et al.*, 2005] I. Gent, T. Kelsey, S. Linton, I. McDonald, I. Miguel, and B. Smith. Conditional symmetry breaking. *Principles and Practice of Constraint Programming-CP 2005*, pages 256–270, 2005.

[Gomes and Sellmann, 2004] Carla P. Gomes and Meinolf Sellmann. Streamlined constraint reasoning. In *CP*, pages 274–289, 2004.

[Heule and Walsh, 2010] M. Heule and T. Walsh. Symmetry within solutions. In *Proceedings of AAAI*, volume 10, pages 77–82, 2010.

[Kouril and Franco, 2005] Michal Kouril and John Franco. Resolution tunnels for improved sat solver performance. In *In Proc. of 8th International Conference on Theory and Applications of Satisfiability Testing*, pages 143–157, 2005.

[Law and von Ahn, 2011] Edith Law and Luis von Ahn. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

[Le Bras *et al.*, 2012a] R. Le Bras, A. Perrault, and C.P. Gomes. Polynomial time construction for spatially balanced latin squares. Technical report, eCommons Cornell University, 2012.

[Le Bras *et al.*, 2012b] Ronan Le Bras, Carla P. Gomes, and Bart Selman. From streamlined combinatorial search to efficient constructive procedures. In *Proceedings of the 15th international conference on Artificial Intelligence*, AAAI'12, 2012.

[Lustig and Puget, 2001] I.J. Lustig and J.F. Puget. Program does not equal program: Constraint programming and its relationship to mathematical programming. *Interfaces*, 31(6):29–53, 2001.

[Ma and Feng, 1984] KJ Ma and CJ Feng. On the gracefulness of gear graphs. *Math. Practice Theory*, pages 72–73, 1984.

[Petrie and Smith, 2003] K. Petrie and B. Smith. Symmetry breaking in graceful graphs. In *Principles and Practice of Constraint Programming–CP 2003*, pages 930–934. Springer, 2003.

[Pickover, 2001] C.A. Pickover. *The Zen of Magic Squares, Circles, and Stars: An Exhibition of Surprising Structures across Dimensions*. Princeton University Press, 2001.

[Puget, 2006] J. Puget. An efficient way of breaking value symmetries. In *Proceedings of the National Conference on Artificial Inteligence*, volume 21, page 117. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[Q.D. *et al.*, 1996] Kang Q.D., Liang Z.H., Gao Y.Z., and Yang G.H. On the labeling of some graphs. *J. Combin. Math. Combin. Comput.*, 22:193–210, 1996.

[Redl, 2003] T.A. Redl. Graceful graphs and graceful labelings: two mathematical programming formulations and some other new results. *Congressus Numerantium*, pages 17–32, 2003.

[Rosa, 1966] A. Rosa. On certain valuations of the vertices of a graph. *Theory of Graphs (Internat. Symposium, Rome*, pages 349–355, 1966.

[Smith, 2006] B. Smith. Constraint programming models for graceful graphs. *Principles and Practice of Constraint Programming-CP 2006*, pages 545–559, 2006.

[Truszczynski, 1984] M. Truszczynski. Graceful unicyclic graphs. *Demonstatio Mathematica*, 17:377–387, 1984.