

Collisions Search v.0.1a

The *Collisions Search* program is designed to search for interactions that match specified conditions, between objects in Conway's Life or similar Moore-neighborhood cellular automata. Similar programs are *catalyst*, *gencols*, *ptbsearch*, and *CatForce*. The objects are one or more polyplets (or larger groups of cells, not necessarily kingwise connected) in any combination, displayed in a table in the program, or read from a file.

The program is written in Delphi 7 with assembler inserts. This version is available only for Windows. The *Collisions Search* program also allows the search results to be viewed and run. It can save all or part of the results to files in Life 1.05 format. Version 0.1a combines the GUI interface with the search program.

1. Additional folders and files for the program

Projects folder: saves default projects (**Projects\Stepfile.txt** file).

Temp folder: saves search results to files in Life 1.05 format.

Tables folder: holds the tables of objects referred to in **tables.txt** and **eaters.txt** files.

Autoload.txt file (optional): controls the program's appearance and its mode of work.

2. Starting the program

First the program reads the file **Tables\tables.txt**. Lines in this file are read in pairs, as follows:

The first line in each pair specifies the size of the square grid used to separate patterns, in an associated table file. In other words, the first line specifies the spacing between objects, and the spacing must be the same in the X and Y directions. The second line gives the name of the table, which must be located in the **Tables** folder. The format of table files can be Life 1.05, Life 1.06, RLE or BMP picture.

The table files are read sequentially, and each file can describe several objects. The content of the file is divided into rows and columns of a square grid. The size of the grid is controlled by the first line of each pair. For example, the objects in the **table10_hersrch.lif** table all fit easily in a 10x10 square, so 10 is the grid size for that table; the **table20_osc.lif** table includes a 15x15 pulsar, so the grid size there is chosen to be 20x20. Each row contains variants of an object. These may be different phases, different orientations, or alternate forms of the object that tend to have the same catalytic effect. The objects are added to the **Objects** table at the bottom of the program window.

The **eaters.txt** file is commonly used to describe eaters. If the **eaters.txt** file is present, it is read line by line. Each line must contain 4 numbers in the following order:

- the row number from the **Objects** table, describing one or more objects;
- the initial column number of the group of objects;
- the final column number of the group of objects;
- the number of generations that the object(s) may be absent or damaged.

For the fourth parameter, if any cells of the object are not present, the absent/damaged generation count will continue to increase. If this maximum number of generations is exceeded, the eater is considered to have been destroyed, and the current configuration is not tracked any further. If the number of generations is set to 0, it is considered that the maximum number is not limited.

It is assumed that if an object in the **Objects** table is not described in the **eaters.txt** file, then there is no limit to the number of generations that the object may be absent. In other words, the object is considered to be a transparent catalyst.

Common objects such as blocks and beehives are likely candidates to use as transparent catalysts. For eaters such as fishhooks that are very unlikely to reappear naturally, it is better to limit the number of generations of allowable absence, to 10 generations for example, to avoid wasting large amounts of time on hopeless searches.

If the **Autoload.txt** file is present, the program reads it for two fields:

[hide] - the program runs minimized on the system tray.

[start] - the **Automatic search** checkbox is initially checked.

3. Description of program control elements

Load project button: select an existing project for its calculation, view the results, or create a new project based on a saved one.

Save project button: save the current project under a new name. The default extension for new projects is *.txt. If no name is set after opening the program window and creating the new project, the program creates a default project named **stepfile.txt** in the **Projects** folder.

Settings button: display the **Settings** project settings tab.

Export button: display the **Export** tab, and possibly save the search results in the project files in Life 1.05 format.

Reset button: reset the results of the project. Changes to the file associated with the project are reverted, to allow a new search to be started.

Start button: start the search process. Changes are periodically made to the file associated with the project, according to the project settings.

Stop button: stop the search process early. This immediately makes changes to the file associated with the project.

Automatic search checkbox: If a project calculation is not currently running, and this checkbox is checked, then the program checks for the presence of project files in a **Projects\Unsolved** folder, every 5 seconds. If one or more project files are found, a search is run automatically on each, after which the project file is moved to the **Projects** folder.

Hide window button: minimize the program window to the system tray. To restore the window, click on the program icon in the system tray.

4. Description of program search settings

Elements of the **Settings** tab define all search settings for the project. If all search conditions are met at any generation (or at the end of the reaction, depending on the settings), the found configuration is saved in program memory. After a certain number of calculated configurations, or whenever the calculation is stopped manually, the accumulated search results are saved to the project file. The following is a description of the elements of the **Settings** tab.

Initial Conditions

The **Initial filling** table contains information about the initial conditions for all objects of the project. Each row describes a variant of an object that will be placed on the field as part of the initial configuration. The number of objects placed is equal to the number of non-empty rows of the table.

For each configuration, each included object has specific coordinates and a specific chosen variant. For a different configuration, the coordinates and/or variant number will be different for at least one object. The coordinates of the object are the coordinates of the upper left corner of its bounding box.

All coordinates (X and Y) in the program may be in the range $-2^{31} \dots 2^{31}-1$. Each object can have coordinates in a certain bounding rectangle that is defined by the table. Each object can have one of the acceptable forms also defined by the table.

StartX field: the X-coordinate of the upper left corner of the bounding rectangle in which an object can be placed.

EndX field: the coordinate of the lower right corner of the bounding rectangle in which an object can be placed. If this field is not specified, it is considered that it is equal to the value in the **StartX** field.

StartY field: the Y-coordinate of the upper left corner of the bounding rectangle in which an object can be placed.

EndY field: the coordinate of the lower right corner of the bounding rectangle in which an object can be placed. If this field is not specified, it is assumed to be equal to the value in the **StartY** field.

Pat. Row field: contains information describing the object. There are four possible options:

- If the **Pat. Row** field contains an integer, it is considered that it determines the row number from the table of objects at the bottom of the program window, and the variants of the object are determined by a range of columns specified in the **Start pat. №** and **End pat. №** fields. If the **End pat. №** field is not specified, it is considered that it is equal to the value in the field **Start pat. №** field.
- If the **Pat. Row** field contains the string "stab", it is considered that the object represents any stable pattern, the dimensions of which are given by the numbers in the **Start pat. №** (X) and **End pat. №** (Y) fields. These dimensions should not be large, because even $X=Y=8$ would allow millions of possible stable patterns.
- If the **Pat. Row** field contains a longer string that begins with "stab", it is considered to be the name of file a Life 1.05-format text file that describes the mask of the stable pattern. Dead cells are indicated as ".", live cells as "*", and undefined cells as "o". The program ignores the first 3 lines of the file (may begin

with “#”). Two extra lines of cells are reserved along each edge, so it is considered that the mask is 4 cells larger than the stable pattern in each dimension. These reserved cells must be empty (“.”) for a sample without the inductive support of other objects.

- If the **Pat. Row** field contains the name of the file describing the object, then the **Start pat. №** and **End pat. №** fields are not specified.

Rel. row № field (optional): If this field is filled, it must contain an integer specifying the row number of one of the previous rows of the table. In this case, the current coordinates of the object are added to the current coordinates of the object in the row referenced by the **Rel. row №** field.

Total Configurations

The total number of configurations of a project is the product of the number of possible configurations of particular objects in the rows of the table. The number of configurations of a particular object is equal to the size of its bounding rectangle area, times the number of possible variants of the object.

- If the **Pat. Row** field contains a row number from the object table, then the number of possible variants of the object is determined by the range of columns specified in the **Start pat. №** and **End pat. №** fields.
- If the **Pat. Row** field contains the string “stab”, then the number of possible variants of the object is determined by the number of possible stable patterns with the dimensions given in the **Start pat. №** (X) and **End pat. №** (Y) fields.
- If the **Pat. Row** field contains a filename, the number of possible variants is 1.

Find the objects in final filling checkbox: When this checkbox is checked, the program reads conditions of the search for objects and some other search settings from the **Final filling** table (to the right of the **Initial filling** table). Restrictions of the objects in the table will be verified when the program recognizes the end of reaction, and possibly also in other generations. If all the conditions are not satisfied, the result is not saved.

In each step checkbox: When this checkbox and the **Find the objects in final filling** checkbox are checked, the conditions for objects in the table will be checked at each generation in a given search range. It should be remembered that in this mode the search is about two times slower compared to the single checking for conditions at the end of reaction. If the conditions are not met, the result in this generation is not saved.

Number of polyplets edit box: When a number is entered in this field and the **Find the objects in final filling** checkbox is checked, the number of polyplets (i.e., catalyst objects) on the field is verified. If these numbers do not match, the result in this generation is not saved.

Consider each object 1 times checkbox: When this checkbox and the **Find the objects in final filling** checkbox is checked, the program verifies that no duplicate objects are present.

Final Matching Conditions

The **Final filling** table contains information about restrictions on the objects found in the current (or final) pattern configuration. Each row describes one limitation.

- If the **Delete** field contains the string “1”, “del”, or “true”, then the object with restrictions of this row of the table must not appear in the final configuration. If this field is empty, the object with restrictions of this row of the table must appear in the final configuration.
- The **OR group** field may contain a number of groups of objects (an integer number greater than 0), united with logical OR operation. To satisfy the conditions on the group of objects, the condition must be met for at least one of the rows of the table with identical values in the **OR group** field.
- If the **OR group** field contains the string “0”, then if at least one condition from the “0” group is satisfied, the result will be saved and marked with “sig” (signal). These “0” group conditions are optional, not required -- i.e., if all other conditions of the search are satisfied, the result is saved.
- At least one variant of object from a row must be found (or none of the variants of objects from a row should be found if the Delete field is not empty) in a bounding rectangle defined in the table. Each object can have one of the possible variants, also defined in the table.
- The **StartX**, **StartY**, **EndX** and **EndY** fields contain the coordinates of the upper left and lower right corners of the bounding rectangle in which the object must (or must not) be found. If the **EndX** or **EndY** fields are not specified, they are considered to be equal to the corresponding **StartX** or **StartY** value. If Start and End values are both not specified, then that coordinate value must not overlap.
- The **Pat. row** field contains information about the objects that must (or must not) be found and which can be set in two possible ways.
 - If the **Pat. row** field contains an integer number, it is considered that it determines the row number from the table of objects at the bottom of the window and the form of the object is determined by the range of columns specified in the **Start pat. №** and **End pat. №** fields. If the **End pat. №** field is not filled, it is considered that it is equal to the value in the **Start pat. №** field.
 - If the **Pat. row** field contains a non-integer string, this is considered to be the name of a file describing the object that must (or must not) be found. The file can be in Life 1.05, Life 1.06, or RLE format, or can be a BMP picture, and it should be in the same folder as the project file. In this case the **Start pat. №** and **End pat. №** fields are not specified.
- If the **Pat. row** field is empty, the **Start pat. №** and **End pat. №** fields should not be set, but there must be an integer number in the **Curr. pat. №** field.
- If the **Curr. pat. №** field is filled, it must contain the correct row number from the **Initial filling** table. In this case, the coordinates of the bounding rectangle for the object that must (or must not) be found, added to the current coordinates of the object in row which number is in the **Curr. pat. №** field of the **Initial filling** table. If the **Pat. Row** field in a row of the **Final filling** table is empty, it is considered that the object that must (or must not) be found is the object of that row of the **Initial filling** table, which number is in the **Curr. pat. №** field.
- The **Clear** button clears the content of the **Initial filling** and **Final filling** tables.
- When the **Number of cells** checkbox is checked, the number of live cells of the configuration is verified, and if it does not satisfy the conditions given in the associated drop-down and edit boxes, the result in this generation is not saved.

Other Controls

Number of birth cells checkbox: when this checkbox is checked, the number of births cells in the configuration is verified, and if it does not satisfy the conditions given in the associated drop-down and edit boxes, the result in this generation is not saved.

Step range checkbox: When this checkbox is checked, the limits on the number of calculating generations of configuration set. The **from** edit box contains the number of generations that will be calculated before condition begin to be checked. Simulation with checking of conditions will be continued until the generation defined by the **to** edit box, plus the number of generations required for the decision of reaction end.

If the **Step range** checkbox is not checked, then verification of conditions begins immediately and is not bounded above. This can lead to an endless loop, error, or premature end of processing if an unlimited-growth configuration appears, or an oscillator with a period that does not divide the period set for the end-of-reaction decision mechanism. Therefore, be cautious about leaving this checkbox unchecked.

Checking the changing of cells number checkbox: When this checkbox is checked, the program verifies that the number of live cells in the current generation is different from the initial configuration. If this number is not changed, the result in this generation is not saved.

Sift the changing of cells number before limit checkbox: When this checkbox and the **Step range** checkbox are checked, the program verifies that the number of live cells in the generation specified in the **from** edit box is different from the initial configuration. If this number is changed, the result in this generation is not saved.

Do not save solutions before range limit checkbox: When this checkbox and the **Step range** checkbox are checked, the program verifies the current generation number. If it is less than a value specified in the **from** edit box, the result in this generation is not saved.

Do not save solutions after range limit checkbox: When this checkbox and the **Step range** checkbox are checked, the program verified the current generation number. If it is more than a value specified in the **to** edit box, the result in this generation is not saved.

Sift the superposition checkbox: The program at the initial filling whether there is a superposition of objects is checked. When the checkbox is checked, if the superposition present the calculation does not start, and the result is not saved. When the checkbox uncheck, extra live cells with same coordinates are removed, and the calculation of the configuration begins.

Save only changed patterns checkbox: When checked, the program in the calculation of configuration verified if the reaction changes at the removal of at least one fixed object, that location and the form does not change for all configurations of project. If no change, the result in this generation is not saved.

***Sift the prolonged absence of the object* checkbox:** When checked, the program compares a time of absence from the reaction with the maximum allowable time of absence set for the object in the **eaters.txt** file. When the allowable time is exceeded for at least one object, the configuration calculating is terminated and the result is not saved. If the object in **eaters.txt** file is not described or has a value of 0, this option is ignored.

***Period of stabilized pattern* field:** contains a period of stabilized reaction P , that used by the decision mechanism at the end of the reaction. The number of live (and birth) cells are checked at generations $n, n-P, n-2P, \dots n-MP$, where n is number of the current generation, and M is the maximum number of checks. If the same number of live cells and the same number of birth cells in all of these generations, the decision on stabilization of the reaction makes. A more complex analysis – checking whether output ships have a clear path or not, for example -- is not carried out. When the reaction stabilizes, if present the oscillator in the field whose period does not divide P , the mechanism is not determine the reaction end. Therefore P should be chosen taking into account all expected periods of oscillators and ships.

***Number of checks of stabilization* field:** contains the number of tests M for the mechanism of decision of the reaction end.

***Start from* field:** contains the current number of the calculated configuration and ranges from 0 to the maximum number of configurations for calculating in the current project minus 1. This field is automatically updated one time for a specified number of calculated configurations. Before starting the calculation, it can be set manually. For a new project it is equal to 0. It also resets when you press the button **Reset** button. For a fully calculated project is equal to the maximum number of calculated configurations.

***Rules* field:** contains shorthand cellular automata rule strings in a standard format **S.../B...**, where after **S** is recorded the numbers of neighbors of a live cell that allow it to survive in the next generation, and after **B** is recorded the numbers of neighbors of dead cell in which it is born in the next generation. The text following **B** must not contain 0.

***Renewal period* field:** defines how often search progress is updated in the program window.

***Saving period* field:** defines how often the calculated configurations and current state of the project are saved to the associated project file.

***Immediately save solutions to a file* checkbox:** When checked, all results found are immediately saved in .lif files in the **Temp** folder.

1.5. Sample setup and explanation for search program tables

Initial filling Clear

Nº	Start X	End X	Start Y	End Y	Pat. row	Start pat. Nº	End pat. Nº	Rel. row Nº
1	2		5		63	0		
2	-3	7	4	6	63	1	6	
3	-3		2	8	pat1.lif			
4	6		3		stab	4	5	
5	3		0		63	0		1
6								

☒ Find the objects in final filling Total number of objects: =
☒ in each step ☐ Consider each object 1 times

Nº	Start X	End X	Start Y	End Y	Pat. row	Start pat. N	End pat. N	Curr. pat. N	Delete	OR group
1	0		0					1		
2	7	8	3		63	0	2			1
3	6		0					3	1	
4	6	8	3	7	63	3	6			1
5					63	7			1	
6										

Objects

	0	1	2	3	4	5	6	7
63								

The rows of the **Initial filling** table:

- 1: Block with coordinates (2, 5);
- 2: Beehive or loaf (6 variants), placed in the bounding rectangle (-3, 4) - (7, 6);
- 3: An object stored in the **pat1.lif** file, placed on the line (-3, 2) - (-3, 8);
- 4: Any stable object with size not more than 4 to 5, with the coordinates (6, 3);
- 5: Block with coordinates (3, 0) relative to the block described in row 1. This produces a biblock with coordinates (2, 5).

The rows of the **Final filling** table:

1. The object described in row 1 of the **Initial filling** table must be found and shifted by the offset (0, 0) relative to its starting coordinates.
2. (and 4.) Because the **OR group** field contains the same number for rows 2 and 4, at least one of the following objects should be found:
 - A block or beehive in a bounding rectangle (7, 3) - (8, 3);
 - A loaf in a bounding rectangle (6, 3) - (8, 7);
3. The object described in row 3 of the **Initial filling** table must not be found with offset (6, 0) relative to its starting coordinates.
5. The boat orientation shown in column 7 must not be found anywhere in the field.

6. Viewing the program search results

The Export tab allows for viewing search results and exporting them to files in Life 1.05 format. Press the **Export** button to jump to this tab. Return to the settings tab by pressing the **Settings** button.

At the top left of the window is a list of groups of search results. Elements of the active group (marked at this list) are listed at the bottom left of the window. This is the solutions of the current project. The names of solutions in the list are formed as follows:

- The first number is the generation number when a solution was found;
- The second number is the number of consecutive generations, starting from the generation number when a solution was found, which also meet the search criteria;
- To the solution name the label "sig" may be added, if this solution is a "signal", i.e. it satisfies additional conditions;
- To the solution name the label "stab" may be added, if this solution appeared at the end of the reaction (according to the decision mechanism);
- The third number is a 16-digit hexadecimal code generated from the coordinates of birth cells in all previous generations for this configuration;
- The fourth number is a 16-digit hexadecimal code generated from the coordinates of live cells in all previous generations for this configuration;
- The fifth number is the number of birth cells in the solution at this generation;
- The sixth number is the number of live cells in the solution at this generation;
- The seventh number is a unique solution number for this project.

16-digit codes are generated as follows: starting with code=0, the value is updated based on each live (or birth) cell with coordinates (x, y), according to these formulas:

```
code=(code+x)*6364136223846793005+1442695040888963407;  
code=(code+y)*6364136223846793005+1442695040888963407;
```

where the cells of a configuration are added row by row, beginning with smaller Y-coordinates, then the smaller X-coordinates in each row.

The names of the solution groups in the list at the top are formed as follows:

- The first number is the generation number when solutions of group were found;
- The second number is a 16-digit hexadecimal code generated from the coordinates of birth cells that are the same for all solutions in the group in all previous generations;
- The third number is the number of birth cells of each solution in the group;
- To the solution name the label "sig" may be added, if this solution is a "signal", i.e. it satisfies additional conditions;
- To the solution name the label "stab" may be added, if this solution appeared at the end of the reaction (according to the reaction-end decision mechanism).

In the elements of one group all these parameters are identical, but any other parameters are differing. The group unite the same solutions in the sense of the reaction process, but differing by stable parts of eaters, and these parts of eaters do not take part in the reaction.

The solutions can be marked in the lower list of the window manually or automatically by applying the export settings. Only marked solutions will be saved to files when received the appropriate command. The group of solutions also can be marked. The group is automatically marked when at least its one element is marked. If a group is marked manually, all its elements are marked as well as a group.

The further is a description of the elements of the export settings.

Click on the **Update** button to apply the export options.

The **Select if the number of solutions is not more than** edit box contains the maximum number of group elements in which it is marked. All its elements are also marked if needed.

When the **Do not export element number** checkbox is checked the configuration object with the number specified in the edit box to the right of the checkbox will not be saved to the .lif file when the export procedure applies. The number of respective object is taken from the **Initial filling** table.

When the **Only stable** checkbox is checked, the program selects only solutions with the “stab” suffix, which are obtained when the end of the reaction is reached.

When the **Only signals** checkbox is checked, the program selects only solutions with the “sig” suffix that satisfy the additional search conditions.

Click on the **Export** button to save the active solution (gray color) in Life 1.05 format to the **Temp** folder.

Click on the **Export all** button to save all marked solutions of the project in Life 1.05 format to the **Temp** folder.

The exported files will have a .lif extension. The names are generated in a similar way to the solution names in the list at the bottom left of the program window. The coordinates of birth (and live) cells of all previous generations of the current configuration are written in decimal format.

- The first number is a code generated from the coordinates of birth and live cells combined, for all previous generations of current
- The third number is a code generated from the coordinates of birth cells of all previous generations for this configuration;
- The fourth number is the number of birth cells in the solution at this generation;
- The fifth number is a code generated from the coordinates of live cells of all previous generations for this configuration;
- The sixth number is the number of live cells in the solution at this generation.

When a solution in the list on the bottom left of the window is selected, it is marked with a gray color and generation 0 is displayed in the view field on the right. The following is a description of the viewing options and controls.

If the **Start/Stop** button is clicked (or the **Enter** key is pressed on the keyboard), the search starts or stops running in continuous mode. During a run, the current generation number, number of live cells and number of birth cells are displayed at the top of the view field.

The delay in milliseconds between calculations of generations for the continuous viewing of configuration can be set in the **Delay** edit box. At a delay of 0, the current generation information is updated every 100 generations, and the configuration is redrawn every 100 generations only if the **Display** checkbox is checked.

If the **One step** button is clicked (or the right-arrow key is pressed on the keyboard), the search is done in steps of 1 generation.

For the viewing field, the scale (screen pixels per cell) can be set with the **Scale** edit box, and the coordinates of the upper left corner can be set with the **X0** and **Y0** edit boxes. The configuration can be dragged and the coordinates of the cell can be

determined with help of the mouse cursor. The information is displayed on top of the viewing field (the **X:** and **Y:** labels).

If the **Historical mode** checkbox is checked when displaying a configuration, the “new” dead cells are marked in gray color and will not be removed from the screen. This allows the size of the configuration to be determined for all previous generations. This option works with the single-step mode, as well as continuous viewing mode at a delay greater than 0. If the scale or position of the configuration on screen is changed, the information about dead cells is lost.

When the **Select mode** checkbox is checked, the mouse cursor will mark an area of the field with a green rectangle, instead of dragging the view. The coordinates of the rectangle are displayed on top of the viewing field (the **Selected:** label). The two points displayed are the top left corner and the bottom right corner of the selection.

When the **Fix the viewing area** checkbox is checked, the scale of view and coordinates of the upper left corner of viewing area does not change when a new active solution is chosen.

Click the **Copy** button to copy generation 0 of the current solution to the clipboard in Life 1.05 format.

1.7. Limitations of the program

The **Collisions Search** program has hard-coded limits on the following elements:

- The size of **Initial filling** and **Final filling** tables: *500 rows*
- The maximum number of live cells in any configuration: *1000000*
- The maximum number of rows in any configuration that include live cells with the same y-coordinates: *100000*
- The maximum number of live cells in a row in any configuration: *100000*
- The maximum number of live cells in polyplet of any configuration: *100000*
- The maximum number of polyplets in any configuration: *100000*
- The maximum total number of live cells of all table elements: *100000*
- The maximum number of cells in polyplet of a table element: *10000*
- The maximum number of polyplets in the table element: *100*
- The maximum size of a stable pattern: *square, 100 x 100 cells*

These numbers may be increased in the next version of the program.

2. Description of project file format v.7

When a project is saved in the *Collisions Search* or *Collisions Exporter* programs, table elements and edit boxes are encoded as follows:

Checkboxes:

0: unchecked

1: checked

Drop-down lists:

0: =

1: <>

2: >

3: >=

4: <

5: <=

The following is a line by line description of the file:

1. The string "Version 7";
2. **Rules** edit box;
3. **Period of stabilized pattern** edit box;
4. **Number of checks of stabilization** edit box;
5. **Saving period** edit box;
6. **Renewal period** edit box;
7. **Immediately save solutions to a file** checkbox;
8. **Checking the changing of cells number** checkbox;
9. **Sift the changing of cells number before limit** checkbox;
10. **Sift the superposition** checkbox;
11. **Consider each object 1 times** checkbox;
12. **Save only the changed patterns** checkbox;
13. **Do not save solutions before range limit** checkbox;
14. **Do not save solutions after range limit** checkbox;
15. **Sift the prolonged absence of the object** checkbox;
16. **Start from** edit box;
17. **Number of cells** checkbox;
18. **Number of cells** drop-down list;
19. **Number of cells** edit box;
20. **Number of new cells** checkbox;
21. **Number of new cells** drop-down list;
22. **Number of new cells** edit box;
23. **Steps range** checkbox;
24. **Steps range: from** edit box;
25. **Steps range: to** edit box;
26. **Find the objects in final filling** checkbox;
27. **Total number of objects** drop-down list;
28. **Total number of objects** edit box;
29. **in each step** checkbox;
30. A string containing a comma-separated pair of numbers giving the size of the **Initial filling** table, in the form: $\{number\ of\ columns + 1\}, \{number\ of\ rows + 1\}$.
31. Lines describing the contents of the **Initial filling** table. The rows of the table are processed sequentially, and each row is written on multiple lines in the file. The elements of each row are written from left to right in a single line. Empty lines at the end of the table are not saved.
32. The string delimiter "-----";
33. A string containing a pair of numbers giving the size of the **Final filling** table, in the form: $\{number\ of\ columns + 1\}, \{number\ of\ rows + 1\}$.
34. Lines describing the elements of the **Final filling** table. The rows of table are processed sequentially, and each row is written on multiple lines in the file. The elements of each row are written from left to right in a single line. Empty lines at the end of the table are not saved.
35. The string delimiter "-----";
36. Pairs of lines that describe all found solutions. The first line of each pair contains the unique solution number. The second line of the pair contains the solution name, which is equal to the corresponding *.lif file name without the extension.