

Stelvio

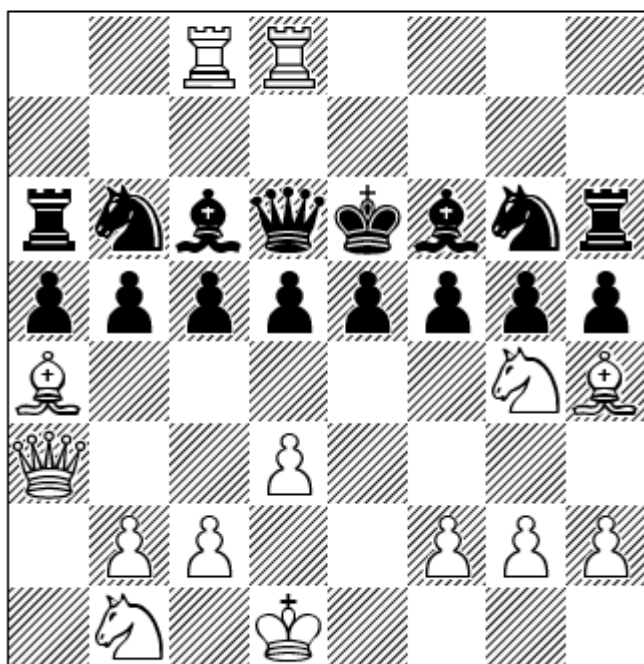
Introduction

Stelvio is a program devoted to solving orthodox SPGs (Shortest Proofgames). To make things bit clearer, its easiest to consider an example. Let us look at a masterpiece from Finland.

Unto Heinonen

Probleemblad 1998

1. Prize



SPG in 23.0 moves

This can be checked by Stelvio, and after around 1.5min, Stelvio comes to the conclusion, that there is exactly one way to reach this position in 23.0 moves:

```

Stelvio 1.0
14+16 Stelvio 1.0 Copyright 2023 Reto Aschwanden
23.0 00:01:30
r s b q k b s r
p p p p p p p p {S,P,P,P,WB,P,P,P} (22/22)
B B {P,P,P,P,P,P,P,P} (1/1)
Q P P P P P
S K 1+0
1 Ke1-d1 2 Ke8-e6 2+0
2 Qd1-a3 1 Qd8-d6 2
4 Ra1-d8 1 Ra8-a6 1+0
4 Rh1-c8 1 Rh8-h6 2
2 Bc1-h4 3 Bc8xPa2|a4-c6 0+0
2 Bf1-a4 3 Bf8xPe2|e5-f6 4
Sb1 2 Sb8-b6
3 Sg1-g4 2 Sg8-g6
1 Pa2-a4 (Bc8) 1 Pa7-a5
Pb2 1 Pb7-b5
Pc2 1 Pc7-c5
1 Pd2-d3 1 Pd7-d5
2 Pe2-e5 (Bf8) 1 Pe7-e5
Pf2 1 Pf7-f5
Pg2 1 Pg7-g5
Ph2 1 Ph7-h5

```

The successful strategy and the unique solution can subsequently be displayed:

```

Stelvio 1.0
14+16 Stelvio 1.0 Copyright 2023 Reto Aschwanden
23.0 00:01:33
r s b q k b s r
p p p p p p p p {S,P,P,P,WB,P,P,P} (22/22)
B B {P,P,P,P,P,P,P,P} (1/1)
Q P P P P P
S K Done. Found 1 solution
8/0k/0m/0.2g

w: Ke1-d1, Qd1-a3, Ra1-c8, Rh1-d8, Bc1-h4, Bf1-a4, Sg1-g4, Pa2-a3, Pd2-d3
b: Ke8-e6, Qd8-d6, Ra8-a6, Rh8-h6, Bc8xPe2|e4-c6, Bf8xPa2|a3-f6, Sb8-b6
Pa7-a5, Pb7-b5, Pc7-c5, Pd7-d5, Pe7-e5, Pf7-f5, Pg7-g5, Ph7-h5

1.Sg1-f3 Ph7-h5 2.Sf3-e5 Rh8-h6 3.Se5-g4 Pe7-e5 4.Pa2-a3 Bf8xa3
5.Pe2-e4 Ba3-e7 6.Ra1-a6 Pc7-c5 7.Ra6-c6 Pa7-a5 8.Bf1-b5 Ra8-a6
9.Bb5-a4 Pb7-b5 10.Qd1-f3 Bc8-b7 11.Rc6-c8 Bb7xe4 12.Ke1-d1 Be4-c6
13.Rh1-e1 Pd7-d5 14.Re1-e4 Sb8-d7 15.Re4-f4 Sd7-b6 16.Rf4-f6 Ke8-d7
17.Rf6-e6 Pf7-f5 18.Qf3-a3 Be7-f6 19.Re6-e8 Sg8-e7 20.Pd2-d3 Kd7-e6
21.Bc1-g5 Qd8-d6 22.Bg5-h4 Pg7-g5 23.Re8-d8 Se7-g6

```

The name

Stelvio, with its 2757m of altitude above sea level, is one of the highest and one of the most beautiful mountain passes of the Alps. As a bike rider, I've been up there many times and have good memories of it. I needed a name, and I liked the way Stelvio sounds. This name breaks the

"tradition" of naming the program after a famous mathematician like Jacobi or Euclide. Let's say the idea of naming it Scholze was not very appealing...



Compatibility

Stelvio can run on Windows and also on Mac (Thierry LeGleuher has tested it on Mac - I personally do not own one). For the moment, the UI is broken on Linux (at least for my installation), something I should fix in a later release. It is possible though to run Stelvio on Linux with the provided `stelvio10.sh` file. The UI looks distorted, but at least the solving process is not affected, and the `problems_out.txt` file with the solving result is correctly generated.

Requirements

Solving SPGs is often memory-intensive, especially for SPGs with a lot of free moves. For that reason, it can be very beneficial to run Stelvio with most of the RAM that you have on your machine. You can find the amount of RAM you have in the System properties, common values for a notebook would be 8g-32g. I'd say 8g is bare minimum (can Tetris run with less?).

Btw: Running Stelvio with way more RAM than is needed for a particular SPG will likely increase the solving time for this SPG, as there are costs involved accessing large amounts of memory on hardware level. You can witness this e.g. in the required ramp-up time: When given a lot of memory, Stelvio needs much longer to get going. This is due to the fact that the large cache needs to be initialized. But for SPGs that take a long time to solve, and added 20 seconds or so up front do not matter. When giving Stelvio almost all or all of your RAM, it is advised to close other applications so that the memory is in fact free to use.

Installation

As Stelvio is written in Java, there needs to be a corresponding Java Runtime Environment installed for it to run. You can get your OS-specific version from here: <https://www.oracle.com/ch-de/java/technologies/downloads/>. I have tested Stelvio with Java 11 and Java 17. E.g. for Windows, you can download Java 17 with this link: https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi. You then need to run the downloaded file and install it to a directory of your choice, recommended is "c:\java\java17" (this path should not contain spaces). Now download stelvio<version>.zip and unzip it to a directory of your choice, e.g. "c:\spg\stelvio". The zip file contains the following:

- bin folder containing some *.jar files (the code)
- doc folder containing PDFs (documentation)
- stelvio<version>.bat (the file to run)
- stelvio<version>.sh (the file to run on Linux)
- stelvioUI.ini (the parameter file)
- problems.txt (example input)

Adjust stelvio<version>.bat

You should open stelvio<version>.bat in a text editor and adjust the maximum memory settings. Per default, it says "-Xmx8g" in the file, which means Stelvio is allowed to use 8g of RAM. If you have more memory available, then give most of it to Stelvio, e.g. "-Xmx16g" for 16g of RAM. Also make sure that the path that points to the Java Runtime Environment is correct. The default is "c:\java\java17\", but you need to adjust this in case you opted to install Java at some other location.

To test your installation, just double-click stelvio<version>.bat, which should run Stelvio with the provided problems.txt file.

Troubleshooting

If Stelvio does not start when double-clicking on the stelvio<version>.bat file, then most likely the Java Runtime Environment cannot be found or its version is too old. Check if the path contained in stelvio<version>.bat pointing to java.exe is correct and adjust if needed.

UI

The simple terminal-based UI gives some impression where Stelvio stands in the solving process.

```

Stelvio 0.97
q s 12+14 Stelvio 0.97 Copyright 2023 Reto Aschwanen
p p 31.5 00:02:37
p r r
p b s b {-,P,P,P,-,-,P,Q} (3/319)
k {P,P,P,P,WB,-,-,S} (1/2) Pa3 Sf6 Pf4 Pe5 Sf3 Bc5
b s 3m/268m
P P P P 119/2k/10m/0.3g
R S B Q K B S R 2+0
Ke1 4 Ke8-f4 2+0
Qd1 (Pe7) 1 Qd8-a8 17
Ra1 2 Ra8-d6 1+0
Rh1 2 Rh8-g6 10
Bc1 2 Bc8-b3 0+0
Bf1 1 Bf8-c5 92
Sb1 2 Sb8-e5
2 Sg1-??-g1 2 Sg8-e8
7 Pa2-a8=R-c6 (Pd7) 2 Pa7xPf2|b6-b5
Pb2 Pb7
Pc2 Pc7
Pd2 1 Pd7xPa2|c6
7 Pe2xPf7|f7-f8=Q-g1 (Ph7) 6 Pe7-e2xQd1|d1=B-h5
7 Pf2-f8=S-b6 (Pa7) Pf7 (Pe2)
Pg2 Pg7 (Ph2)
7 Ph2xPg7|g7-g8=Q-d1 6 Ph7xPe2|g1=S-h3

```

What it all means:

- Next to the board:
 - Number of pieces and number of moves are obvious.
 - The green lines: How the pawns are assigned to partitions currently in the strategy seeking process. Partitions are the types of pieces on the board that are mutually exclusive, we have partition values (K, Q, R, WB, BB, S, P). WB stands for white square bishop and BB for black square bishop. Top line is white assignment {-,P,P,P,-,-,P,Q}, lower line is black assignment {P,P,P,P,WB,-,-,S}. This means for white, that a/e/f pawns are captured, b/c/d/g pawns remain pawns and that the h-pawn is a queen in the diagram position. For black, a/b/c/d pawns remain pawns, the e-pawn is a white square bishop visible in the diagram, f/g pawns are captured and h-pawn promotes to knight and is visible in the diagram. At the outset of strategy seeking, all possible assignments of pawns to partitions are calculated, in the example we have 319 for white and 2 for black. During strategy seeking, these assignments are all gone through in a big loop, and we are currently at assignment 3 for white and 1 for black. This information gives you a hint how far strategy seeking and therefore solving already is.
 - 2+0: Number of free moves for the current strategy by color, so 2 for white and 0 for black.
- On the right:
 - Solving time so far (hh:mm:ss).
 - Current move path (not to be taken too seriously, this can be stale/inconsistent data for technical reasons. If garbage is displayed, this has no effect on the solving itself, only the display is invalid).
 - Cache/strategy metrics:
 - 3m/268m: Currently, the playing cache is basically empty (3 million). Cache can hold 268

million positions. This cache is bigger if you have more RAM available, which can be very useful depending on the SPG in question.

- 119/2k/10m/0.3g:
 - 119: The strategy number of the current strategy being played.
 - 2k: The number of strategy found so far in thousands. All but 119 were subsequently filtered by strategy analysis.
 - 10m: Number of moves played for the current strategy in millions.
 - 0.3g: Total number of moves played for all strategies so far, in billions.
- Below the board:
 - The current strategy, i.e. what each piece does, white on the left and black on the right, with associated move count per piece.
 - The column on the far right: Histogram of already played strategies (top 8 entries, as there is limited space). So in the example, 17 strategies with 2+0 free moves have already been played, next to strategies with 1+0 and 0+0 free moves.

When a solution is found or at the end of solving, pressing "Space" displays the solution and the successful strategy:



When there are cooks, Stelvio counts the number of strategies that contribute to cooks, in the example 63. In case strategies which can be uniquely played are found, these are counted as valid solutions and displayed as well.

A brief note on what Stelvio counts as a cook/solution: In case a SPG requires an even number of half-moves, e.g. 22.0, then any solution/cook needs to have an even number of half-moves. Correspondingly for SPGs requiring an odd number of half-moves. This seems a good approach to take, since who is at play is part of the position to reach. So in case in the example one could reach

the diagram position in 21.5 moves, then this does not count for Stelvio. Any shorter move-path with same parity of half-moves does count though, so in the example, Stelvio would count solutions/cooks in {21.0, 20.0, 19.0, ... 1.0} moves.

Stelvio 1.0 Copyright 2023 Reto Aschwanen

13+13 21.0 00:00:25

{P,P,-,-,P,-,P,P} (1/85)
{-,P,P,P,P,P,-,P} (4/15)

63 cooks
Pc4 Pd5 Qa4 Qd7 Qa5 Qh3
0m/134m
905/250k/0m/0.0g

| | | |
|----------------------------|---------------------|-----|
| Ke1 | 3 Ke8-e5 | 6+0 |
| 4 Qd1xRa8 a8-d1 | 2 Qd8-h3 | 8 |
| Ra1 | Ra8 (Qd1) | 5+0 |
| Rh1 | 2 Rh8-g5 | 12 |
| Bc1 | 1 Bc8-e6 | 4+0 |
| Bf1 | 3 Bf8xPd2 d2-e3 | 29 |
| Sb1 | Sb8 | 3+0 |
| Sg1 | 2 Sg8-f7 | 70 |
| Pa2 | Pa7 (Pc2) | 2+1 |
| Pb2 | Pb7 | 8 |
| 8 Pc2-c8=QxPa7 a7-a4 (Pc7) | 3 Pc7xPf2 b6xPc2 a4 | 2+0 |
| Pd2 (Bf8) | 1 Pd7-d5 | 177 |
| Pe2 | Pe7 | 1+1 |
| 7 Pf2xPg7 g5-g8=Q-b6 (Pc7) | 2 Pf7-f4 | 9 |
| Pg2 | 1 Pg7-g5 (Pf2) | 1+0 |
| Ph2 | 1 Ph7-h6 | 316 |

Histogram mode

In order to get a first impression if an SPG can be solved in a reasonable time, it can be useful to simply search for all the strategies without actually playing them. This is what the histogram mode is for. In this mode, strategies are added up and grouped by white/black free moves.

Stelvio 1.0 Copyright 2023 Reto Aschwanen

12+14 31.5 00:00:23

{S,P,P,P,S,WB,P,Q} (319/319)
{P,P,P,P,WB,S,-,-} (2/2)

Done. Histogram completed
2316/17k/0m/0.0g

| | |
|-----|--------|
| 2+0 | : 352 |
| 1+0 | : 612 |
| 0+0 | : 1352 |

So in the example, there are 352 strategies found with 2+0 free moves, 612 with 1+0 and 1352 with 0+0 free moves. Strategy seeking only took 23 seconds, so computing the strategy histogram is much faster than solving the SPG (in this particular case). The SPG looks solvable in reasonable time given this histogram information. It is also possible to attain a partial histogram: You can tell Stelvio to only start adding up strategies after strategy number X. This can be useful if you want to know what is left in terms of strategies, if Stelvio already solved up to strategy X.

User interaction

The playing of the current strategy can be stopped by typing 's'. Stelvio will move on to the next strategy thereafter. Pressing Ctrl-C cancels the solving process.

Input / Output

A simple text file (by default problems.txt) serves as input. It needs to be in the same directory as stelvio<version>.jar. In problems.txt, the SPG needs to be given in FEN notation on the first line, and the number of half moves on the second line, something like:

```
1nbq4/ppk1p3/Rp5p/3npr2/R3P3/2br1B1P/PP2P2P/1NBQNK2
65
```

Pieces are denoted by:

- K/k = King
- Q/q = Queen
- R/r = rook
- B/b = bishop
- N/n/S/s = knight
- P/p = Pawn

The result is written into an output file (by default named problems_out.txt). There is a stelvioUI.ini file for parameters that can be adjusted by the user. If no such file is present, then default values are used. See also StelvioParameters.pdf.