An interactive Matlab programme for Steiner trees

Marcelo Zanchetta do Nascimento, Valério Ramos Batista,
Wendhel Raffa Coimbra

1 Getting started

Download `stree.zip` from `http://hostel.ufabc.edu.br/~marcelo.nascimento/software.html` and extract it in a folder. Enter "stree" at the Matlab prompt. You'll be asked to adjust the window so that figures won't hide it. You may either give an existing datafile of terminal points for the tree or plot one with the mouse left-button. Some test-files with extension ".txt" can be found in `stree.zip`, and we'll take them here for comments.

Press the Enter key after you either finish plotting points or type the filename *without* extension, for instance `test0`. Figure 3 will appear with the tree of Prim, and Figure 4 is for you to draw a Steiner's. Our purpose is to find the *shortest* tree, and Figure 3 can help make good choices.

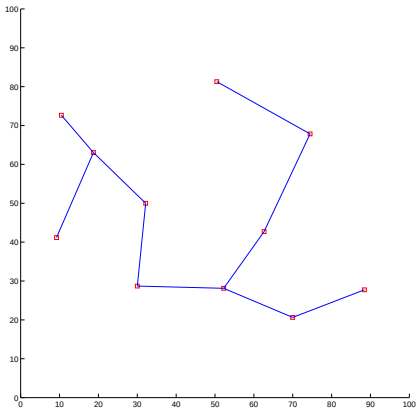

Figure 3: The tree of Prim.

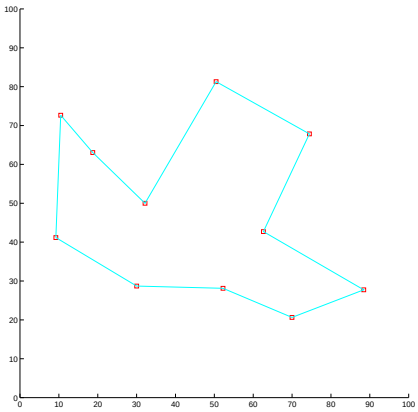

Figure 4: Terminals joined by auxiliary cyan edges.

In Figure 4, the cyan polygon is obtained by the Convex Hull and Lune properties described in [2]. From this point on, some hints and instructions to proceed are printed in the Matlab terminal window, and all actions will be with the mouse. For the tree of Prim, its length *Lprim* and $\frac{\sqrt{3}}{2}$*Lprim* are printed, and so the user gets an estimate on the length of an SMT from Gilbert and Pollak's conjecture. However, according to [3], it might even happen that we get an SMT *under* this ratio.

From the *first mouse menu*, now printed in the Matlab terminal window, you can read the drawing options:

Press button
l(+l)+r to omit point(s);
r alone to try full tree;
middle for more options.

For instance, choose $(75, 68)$ as the first and $(30, 29)$ as the second point with the left button, and then press the right button. A dotted black polygonal will blink to indicate a remaining cyan polygon, of which the vertices are now joined by a Steiner subtree computed by the programme (see Figures 5 and 6).

From the first to the second point, one goes *counterclockwise* along the black polygon. It's a standard of `stree.m`, but if you mistake it *before* clicking the right button, choose extra point(s) and the

programme will treat the latter two as "first" and "second". If you mistake it *afterwards*, the middle button will print the *second mouse menu* in the Matlab terminal window. From it we have that the right button now undoes your mistake, and with the left button you'll be back to the first mouse menu.

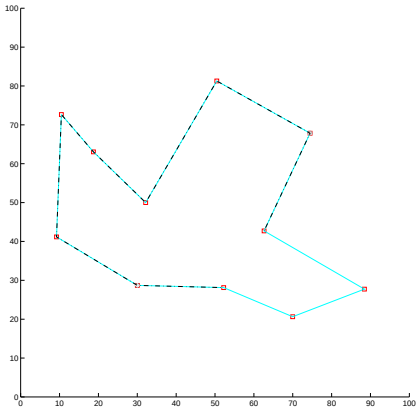

Figure 5: The dotted black polygonal.

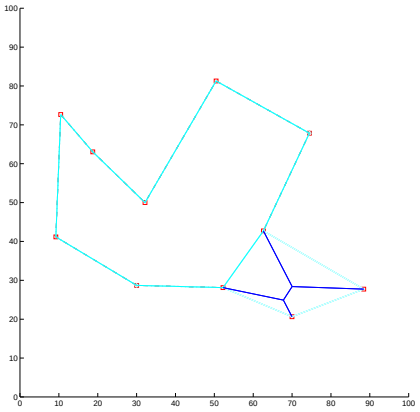

Figure 6: A Steiner subtree.

You might now want to take $(19, 63)$ and then $(52, 29)$ to get a Steiner subtree out of a quadrilateral. That will work out, but Figure 3 doesn't hint this way. So try instead $(32, 50)$ and $(52, 29)$ and you'll get Figure 7 with an "odd" cyan polygon. However, `stree.m` shows it for you to consider other possibilities not hinted by Figure 3, for instance Figure 8. But in this figure, the Steiner tree isn't minimal because it has length $Ltree = 227.1818$, which is bigger than $Lprim = 211.1398$, as printed in the Matlab terminal window.

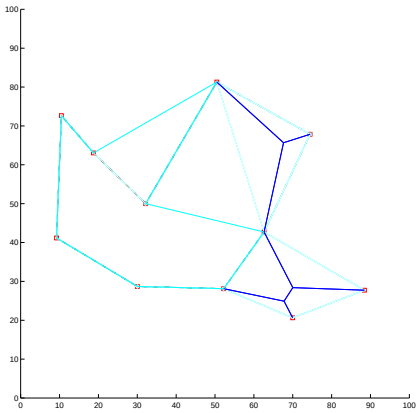

Figure 7: An "odd" cyan polygon.

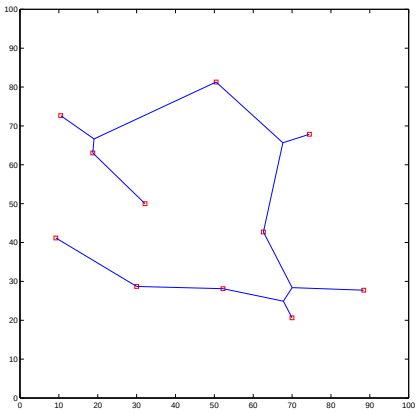

Figure 8: A non-minimal Steiner tree.

Back to Figures 7 and 3, remove now $(50, 81)$ by clicking on it with the left and then right mouse buttons. Remove $(10, 73)$, and then $(62, 43)$ in the same way. The order you remove these single points is unimportant. You'll then be left with a group of terminals very likely to give a full Steiner subtree. It'll be drawn by pressing the right mouse button alone, no matter the cursor is. The Steiner tree will be concluded by connecting the isolated point $(10, 73)$. By the *third mouse menu*, which is also the last one now printed in the Matlab terminal window, you should click left on $(19, 63)$ and then right on $(10, 73)$. The middle mouse button concludes your drawing (see Figure 9) and informs that $Ltree = 206.1456$. Sometimes, as in this case, you go from the first to the third mouse menu without access to the second.

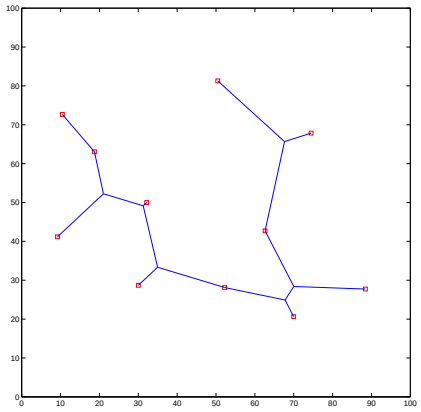

Figure 9: The completed Steiner tree.

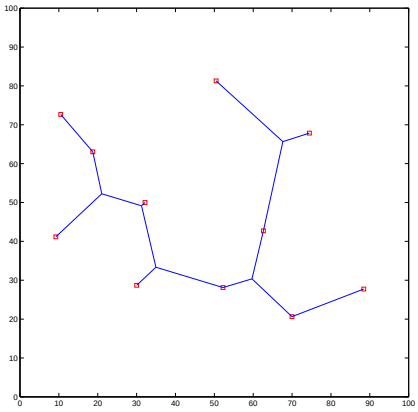

Figure 10: The supposedly minimal Steiner tree.

However, we performed several tests which suggest that the tree in Figure 10 is the shortest one. It has $Ltree = 201.1988$.

2 Full tree stretches

In the previous section we showed some steps that led to a group of terminals, which eventually gave a full Steiner subtree by pressing the right mouse button. Herewith we present some details and hints, so that the intuitive mind will hardly be wrong at identifying such groups.

When a subgroup of terminals will admit connection by a full Steiner tree? For the time being, we answer this question providing each Steiner point is directly connected to a terminal of the subgroup (see Theorem 4.3 below). In general, it's when you have a "good" zigzag polygonal connecting them. For instance, run "Mksaw" for the terminal data `test1.txt`, and do the same with "stree". You'll get Figures 11 and 12.

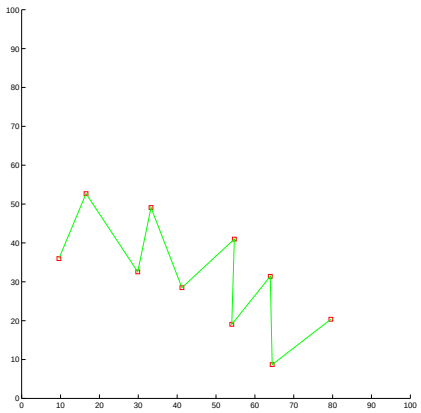

Figure 11: A "good" zigzag.

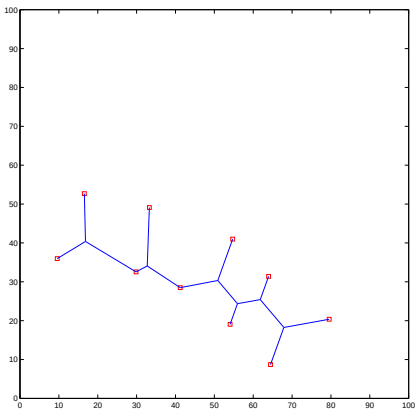

Figure 12: An "almost" full tree.

This example shows that `stree.m` won't let you empty-handed, even if a full tree doesn't exist. However, it relies on your guesses. Whenever they're wrong, choose the "undo" option from the second mouse menu, or simply run again "stree" if most of your terminals were a group like in Figure 11. This second case is illustrated in Figure 13 (run "stree" for `test0` and omit the two uppermost terminals to get it). Figure 14 shows what's wrong: the zigzag is quite irregular.

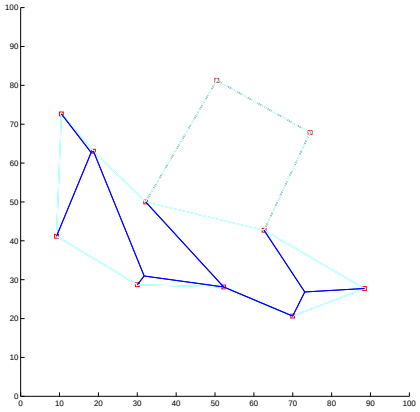

Figure 13: A wrong guess.

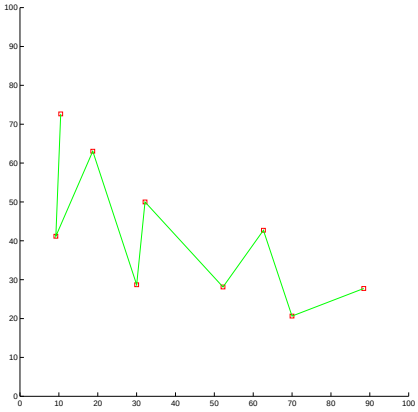

Figure 14: A "bad" zigzag.

Here are some hints to identify good zigzags: the group of points forms a strip, which can scroll in any direction. The strip can be slightly bent or waved, but its width may oscillate even quite a lot. However, despite all hints we give, only the practice will really make you recognise the good zigzags very quickly.

References

[1] D.-Z. Du, F.K. Hwang, A proof of the Gilbert-Pollak conjecture on the Steiner ratio, Algorithmica 7 (1992) 121–135.

[2] E.N. Gilbert, H.O. Pollak, Steiner minimal trees, SIAM Journal of Applied Mathematics 16(1) (1968) 1–29.

[3] N. Innami, B.H. Kim, Y. Mashiko, K. Shiohama, The Steiner ratio conjecture of Gilbert-Pollak may still be open, Algorithmica 57(4) (2010), 869–872.