We tested the performance of strategy improvement vs Gauss-Seidel method on max-linear equation systems with $10\,000, 20\,000, \ldots, 100\,000$ variables. For each of the sizes we randomly generated one hundred tests and ran both algorithms on all of them. Every third equation in the tests generated had a max operator in it. Henceforth, each variable will be called a *node* and every equation with a max operator will be called *maximizer player's node*.

The initial strategy for each node belonging to a maximizer player was chosen at random. At each step of the simultaneous strategy improvement we first compute the solution with the strategy fixed for each node. Then for each of the max nodes whose current reward value can be improved by more than a small threshold 10^{-10}, we switch the strategy for that node and pick the best reward neighbor. After that we fix the strategy again to the modified one and compute the LFP of the new linear equation system generated for it. We repeat that until by switching a strategy at one of the nodes the change of the reward for that node was higher than ε (picked to be 10^{-8}). The small threshold 10^{-10} described earlier is crucial since otherwise the floating point errors can make the strategy improvement algorithm believe that it improves the value of a node at each step although it is in fact alternating between two possible strategies for that node and it would repeat that forever.

Gauss-Seidel method returned the correct LFP solution for every single equation system examined. On the other hand strategy improvement in about 10% of the cases did not converge. It was due to a failure in convergence of the underlying sparse linear equation solver – Biconjugate Gradient, and it was despite the fact that all the matrices encountered during that computations were invertible. The average running time of the **successful runs only** for each algorithm is plotted on Figure 1. Notice the big variability of the running times for the Gauss-Seidel method. This follows from that fact that sometimes we randomly generate a very hard instance to solve by a simple iterative algorithm such as Gauss-Seidel. However these instances are not much harder for the strategy improvement than any other instances of that size. The same phenomena was observed for random 1-RMCs ([WE07tacas]). Notice also that the average running time of the strategy improvement algorithm looks linear in the size of the equation system.

In order to evaluate both methods more precisely, the minimum, maximum and the median running time for exactly the same examples is plotted on Figure 2. Also for comparison the average running time is plotted on the same graph. Notice that the Gauss-Seidel method has a huge variability in its running time among different equations of the same size. For instance it can solve some examples with $40\,000$ variables within a time as as little as one second and for some others it requires as much as 225 seconds. The most surprising fact is that the median running time for Gauss-Seidel method turned out to be much lower than for the strategy improvement. For the examined sizes (apart from the size $40\,000$) the median is about three times lower than the median for the Newton's method. On the other hand the constant performance of strategy improvement is impressive. Its median running time for a given size is very close to the minimum running time for that size and the variability in the running time is really small.

That allows us to predict more accurately the time needed to find the solution for a given instance. We should also remember that the average running time of strategy improvement is much lower than using Gauss-Seidel method, hence although Gauss-Seidel's median running time is lower, it seems more preferable to use strategy improvement algorithm with linear sparse solver whenever we can.

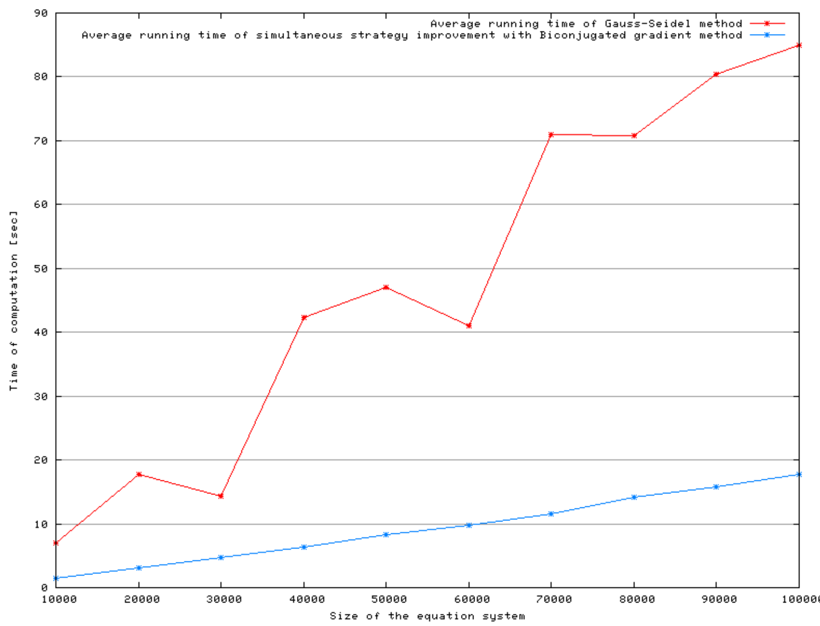

Fig. 1. The blue line shows the average successful running time of simultaneous strategy improvement that uses Biconjugated gradient method as its sparse linear solver. The red line is the average running time of the Gauss-Seidel method.

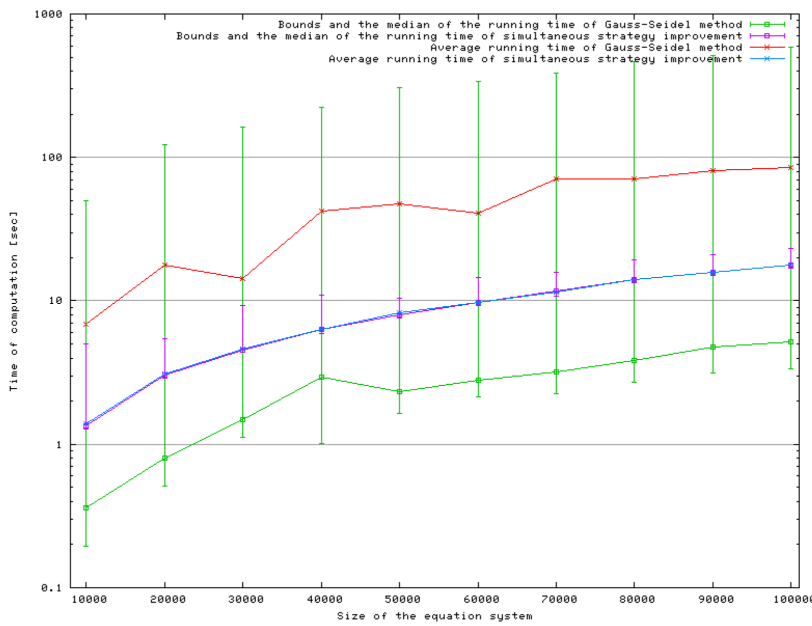

Fig. 2. The minimum, maximum, median and average running times of each of the tested numerical approximation algorithm. Notice that the X-axis is linear, but the Y-axis (the running time) is logarithmic in the powers of 10. The green bars shows the minimum and maximum running time of the Gauss-Seidel method for each of the sizes. A green cross shows the median running time. The magenta bars shows the successful minimum and maximum running time of simultaneous strategy improvement that uses Biconjugated Gradient method as its sparse linear solver. The blue line shows the average successful running time of simultaneous strategy improvement that uses Biconjugated Gradient method as its sparse linear solver. Notice that is line is almost identical to the magenta line connecting the median values for each of the sizes. The red line is the average running time of the Gauss-Seidel method. Notice the huge difference between the red line and the green line connecting the median running time for Gauss-Seidel method.