

Ondrej Such *

EXPERIMENTAL BEHAVIOR OF JURIK'S NEAREST POINT APPROACH ALGORITHM FOR LINEAR PROGRAMMING

T. Jurik recently proposed a new algorithm for solving linear programming problems. The algorithm is iterative, finding points on the boundary of the problem's convex polyhedron. Global convergence of the algorithm is an open question. Jurik benchmarked the performance of the algorithm on standard sets of linear optimization problems, and the algorithm was on par with commonly used ones, and sometimes beating the simplex method by a wide margin. In this note we concentrate on testing local behavior of the algorithm near a vertex in the three dimensional Euclidean space. Our experiments indicate that the behavior of the algorithm is mostly very fast, but there appear to be cases where its behavior is worse than the simplex method.

1. Linear programming problems

In linear programming problems [3] one tries to minimize a linear cost function on a set of higher dimensional vectors bounded by linear constraints. The simplex algorithm [4] to solve this problem was proposed by B. Dantzig in 1947. The algorithm is iterative, moving through vertices of the convex polyhedron bounded by the problem's constraints. It performs quite well in practice, and is still being used. However, it was shown in 1972 by Klee and Minty that in a worst case it takes exponentially many steps. Multiple polynomial time running algorithms were proposed such as L. Khachinyan's ellipsoid method in 1979 and N. Karmarkar's in 1984. The research in this topic of high practical importance is ongoing, with open problems even for the simplex method such as Hirsch's conjecture.

2. Jurik's algorithm

Jurik in [1] proposed a new algorithm for solving linear optimization problems. It starts with constructing a hyperplane H orthogonal to the cost direction that lies sufficiently far from a convex polyhedron P . One then observes that to find the optimal x is the same, as to find pairs of points w, x , where w lies in H and x lies in P whose distance is minimal. The algorithm has an *outer* iterative cycle and embedded within it is an *inner* iterative cycle. Let us briefly describe what is computed at each iteration of the cycles.

The algorithm stipulates how to generate initial points w^0 and x^0 (the latter is generated first). Suppose in the k -th iteration of the outer cycle we generated points w^k lying in the hyperplane H and x^k in P (necessarily on the boundary of P). A line L^k is then

defined as the projection of the line passing through w^k and x^k onto the hyperplane H . New points w^{k+1} and x^{k+1} are then generated in the inner cycle (which Jurik showed terminates in finite number of steps) as the points of minimal distance lying on L^k and P respectively. Because the line L^k contains the projection r^k of point x^k onto H , with increasing k the distance of point x^k to H (the cost function) cannot increase. In fact we have the following inequalities for distances

$$\begin{aligned} d(x^{k+1}, H) &\leq d(x^{k+1}, L^k) && \text{because } L^k \text{ lies in } H \\ d(x^{k+1}, L^k) &\leq d(x^k, L^k) && \text{because } x^{k+1} \text{ is a point in } P \\ &&& \text{minimizing distance to } L^k \\ d(x^k, L^k) &= d(x^k, H) && \text{because } L^k \text{ contains the} \\ &&& \text{projection of } x^k \text{ onto } H \end{aligned}$$

and thus

$$d(x^{k+1}, H) \leq d(x^k, H)$$

At this time, it is not known whether the algorithm always converges. Jurik himself proved that the inner cycle stops after a finite number of iteration. The following conjecture was indicated by Jurik:

Conjecture A Jurik's algorithm converges to the optimum in a finite number of steps.

3. An example needing many line iterations

Jurik in his thesis [2] reports that in the vast majority of cases, he needed to use only two lines to find the optimum. In our experiments we found a 3-dimensional problem which takes many more iterations. Let us define the vectors

* Ondrej Such

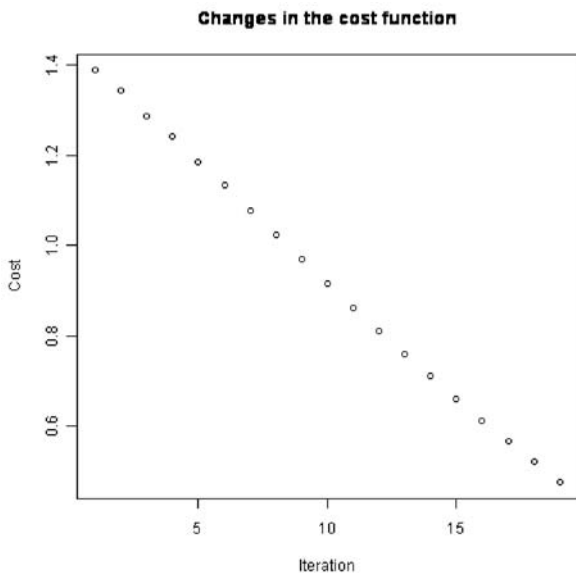
Institute of Mathematics and Computer Science, Banska Bystrica, Slovakia, E-mail: ondrejs@savbb.sk

$$\begin{aligned} V_1 &= (0.18484 \ 0.21435 \ 0.95911) \\ V_2 &= (-0.15919 \ 0.97907 \ 0.12678) \\ V_3 &= (0.07727 \ 0.0453 \ 0.99598) \end{aligned}$$

and the line L_1 passing through the following two points

$$\begin{aligned} P_1 &= (3.09017 \ 9.51057 \ -1) \\ P_2 &= (2.13911 \ 9.81958 \ -1) \end{aligned}$$

Consider the problem of minimizing the cost function $c(x,y,z) = z$ inside the convex hull of nonnegative multiples of vectors V_1, V_2, V_3 . In this case one finds almost linear convergence of Jurik's algorithm towards optimum value of 0, as shown in the following figure.



This leads us to formulate the following conjecture, which indicates that in the worst case the running time of Jurik's algorithm is worse than the running time of simplex algorithm.

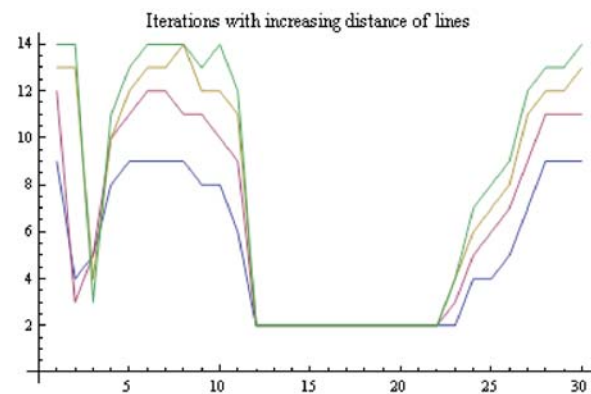
Conjecture B The number of steps in Jurik's algorithm is not bounded by any function of the problem's dimension and the number of constraints.

4. Typical behavior

In this section we present an example, that based on our experience, we believe is quite typical for running behavior of Jurik's algorithm on three dimensional problems. We chose three random vectors:

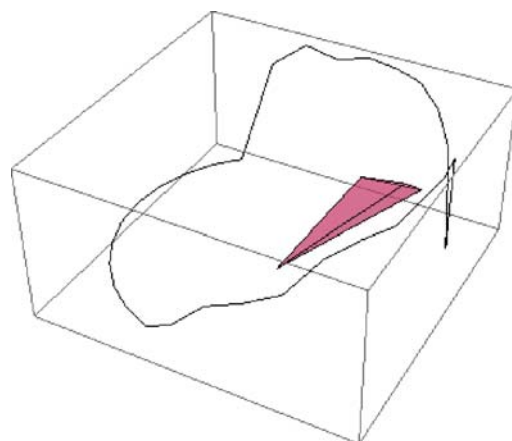
$$\begin{aligned} V_1 &= (0.524169 \ -0.000129183, \ 0.851614) \\ V_2 &= (0.674312 \ 0.285704 \ 0.680938) \\ V_3 &= (0.631712, 0.67865, 0.374667) \end{aligned}$$

We looked at the cost function $c(x,y,z) = z$ and tried to minimize it using Jurik's algorithm on the convex set spanned by non-negative multiples of V_1, V_2, V_3 . For H we chose the hyperplane $z = -1$, and there we chose 30 lines at distances 10, 20, 30 and 40 from the point $(0 \ 0 \ -1)$. The following figure shows that the number of iterations for those 30 lines increases slightly with the increasing distance.



We see that for quite a few starting line positions it took just two iterations to converge to optimum (the precise counts are 12, 13, 13, 13), confirming the behavior observed by Jurik.

The next figure shows how the position of the line relative to the optimizing set affected the number of iterations. The higher the point on the polygonal line the more iterations it took starting with the line (at distance 40 from $(0 \ 0 \ -1)$) passing through its projection to converge to the minimum.



5. Conclusion

Jurik presented in his works [1], [2] compelling evidence that his new algorithm is competitive with commonly used linear opti-

mization algorithms. Based on our, admittedly very limited, experimental experience we stated two conjectures about the behavior of the algorithm. In case the Conjecture B holds, a modification

of the algorithm is needed to deal with some linear programming problems, perhaps by rescaling the problem, a clever choice of hyperplane H, or some other method.

References

- [1] JURIK, T.: *A new efficient two-path algorithm for linear programming problems*, Journal of Applied Mathematics, Statistics and Informatics, 4 (2008), pp. 129-138
- [2] JURIK, T.: *Contributions to the Algorithms of Linear Programming*, Ph.D. thesis, Comenius University, 2009
- [3] http://en.wikipedia.org/wiki/Linear_programming
- [4] http://en.wikipedia.org/wiki/Simplex_algorithm