Which point in that convex polygon is closest to me?

The following Algorithm 1 finds among the points that belong to a given convex polygon the one point with minimum distance to a given goal point. Being very similar to previous work [1], the method sligthly varies incremental linear programming [2], [3]. It receives the convex polygon as a set of halfplanes whose intersection defines the polygon. It requires to know the solution of the same problem with one halfplane less and which is the respective halfplane. The algorithm allows to incrementally build the final solution for any polygon, by solving first the problem without any halfplanes, then adding one more halfplane and solving again, and so on, until it has processed all the halfplanes.

Algorithm 1 Incremental Distance Minimization

Input: halfplane h, goal point g, prior solution $\tilde{s}$, prior halfplanes $\left\{\tilde{h}_j\right\}_{j=1}^{n}$.

Output: solution s being closest to g while belonging to $\bigcap_{j=1}^{n} \tilde{h}_j \cap h$.

```
s ← s̃
if s ∉ h then
  s ← projectPointOrthogonallyOnHalfplaneBoundary(g, h)
  ĥ ← ℝ² //will be a halfplane later
  for j ← 1, ..., n do
    if s ∉ h̃_j then
      s ← intersectHalfplaneBoundaries(h̃_j, h)
      if s ∉ ĥ then
        return "infeasible"
      end if
      ĥ ← h̃_j
    end if
  end for
end if
return s
```

References

[1] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[2] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," *Discrete & Computational Geometry*, vol. 6, no. 3, pp. 423–434, 1991.

[3] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational geometry: algorithms and applications.* Springer, 2008.