Rodrigo I. Silveira

Maike Buchin

Maarten Löffler

Boris Aronov

Tom de Jong

Feed Links for Network Extensions

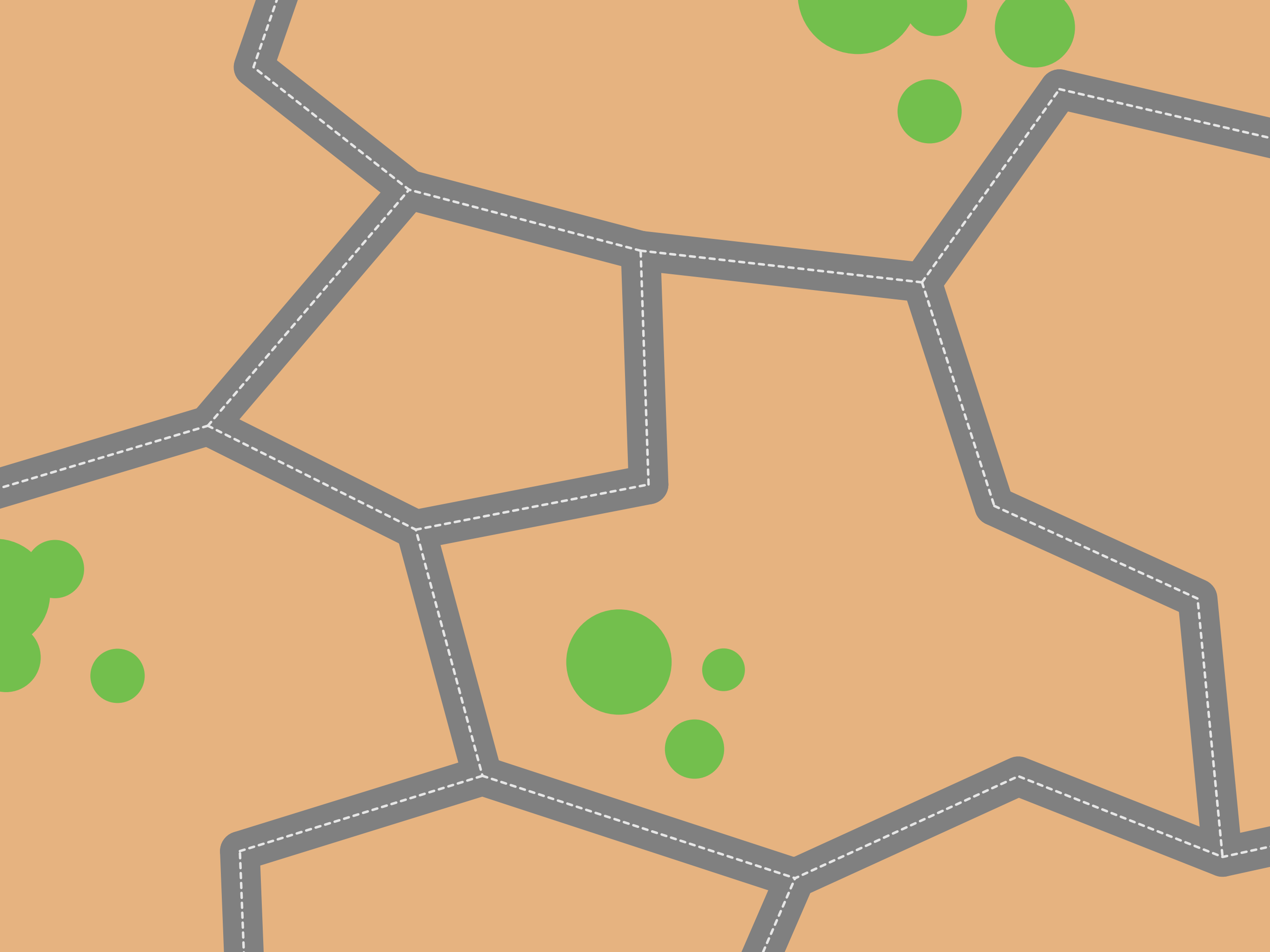
Jun Luo

Bettina Speckmann

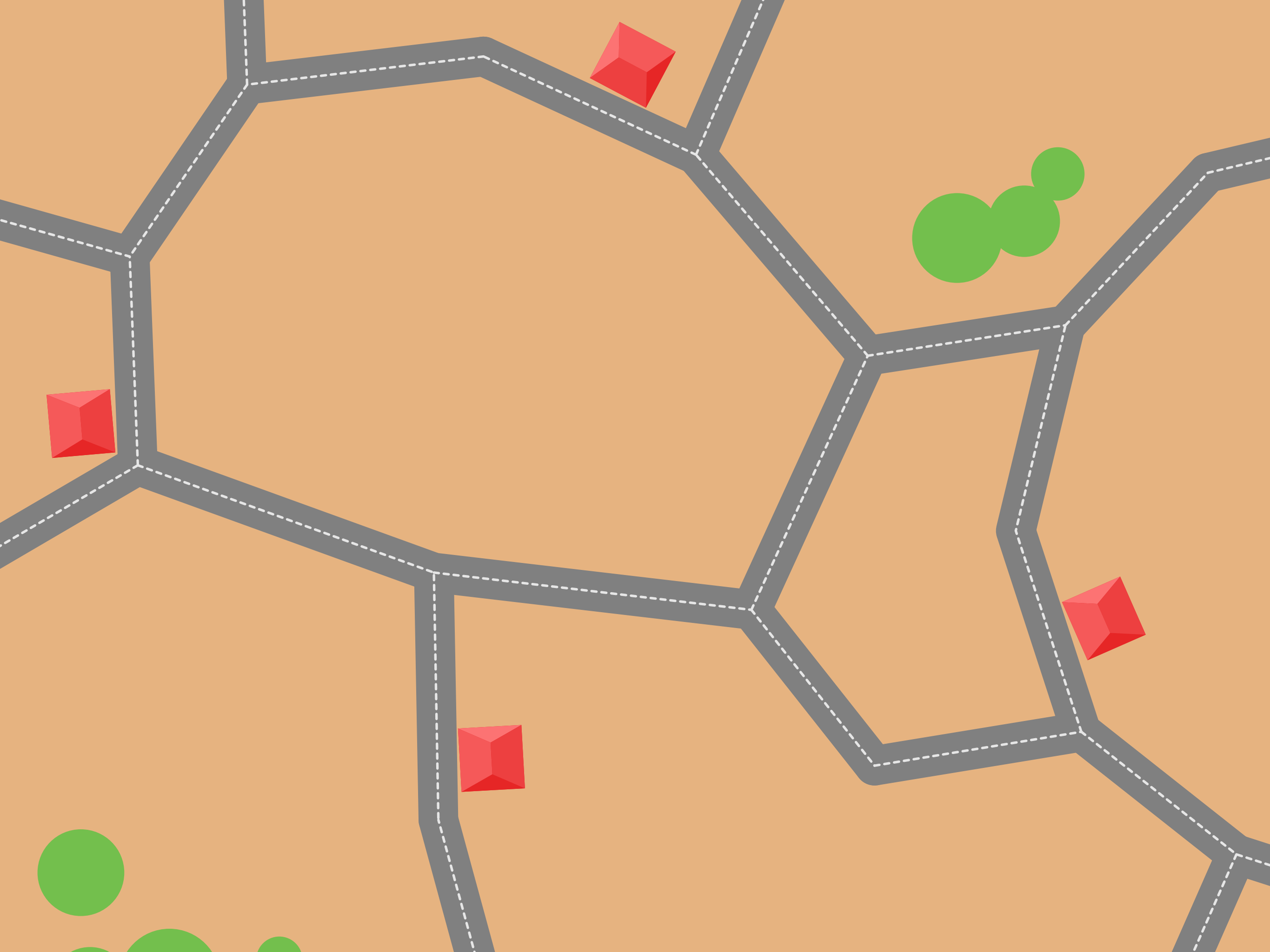
Bart Jansen

Kevin Buchin

Marc van Kreveld



Consider a network of roads.



Also consider a set of *interesting* locations.

Also consider a set of *interesting* locations.

In many applications, the network distance between locations is important.

Also consider a set of *interesting* locations.

In many applications, the network distance between locations is important.

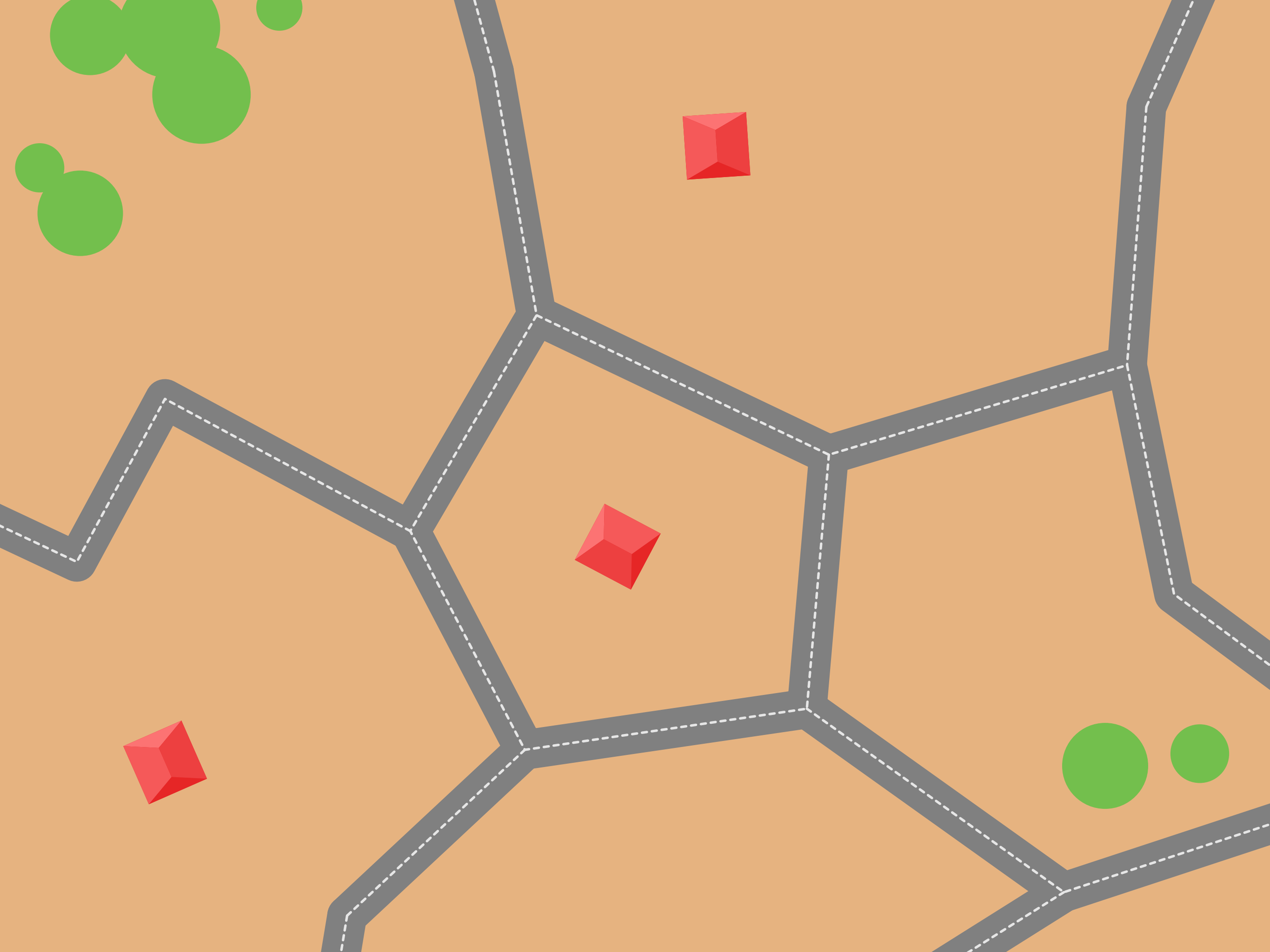
Also consider a set of *interesting* locations.

In many applications, the network distance between locations is important.

Also consider a set of *interesting* locations.

In many applications, the network distance between locations is important.

These are well known and easy to compute.



What to do if the road network data is not complete?

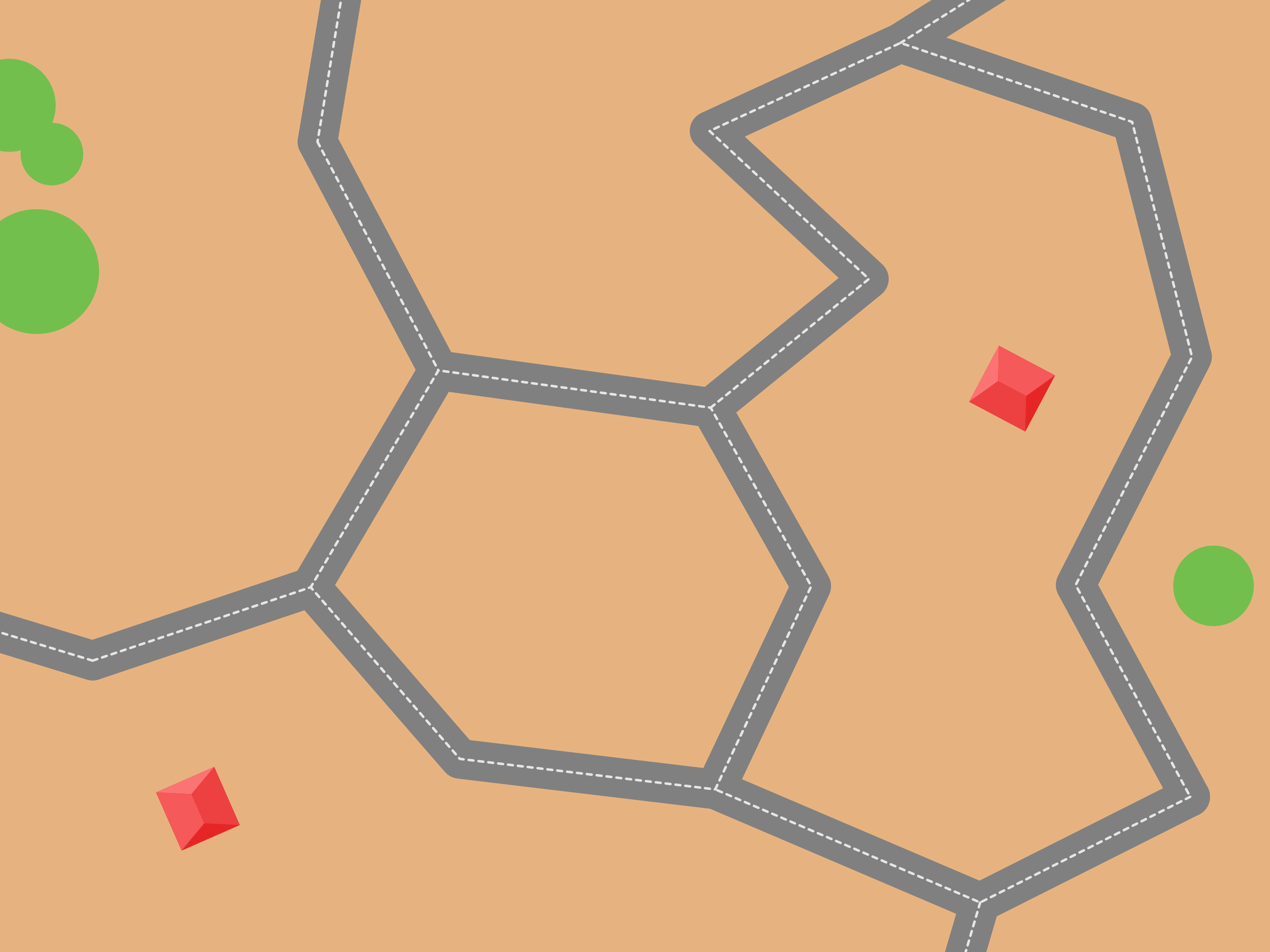
What to do if the road network data is not complete?

Locations may be disconnected from the network.

What to do if the road network data is not complete?

Locations may be disconnected from the network.

Still, we often know that some connection must exist.



There must be some path from p to q.

There must be some path from p to q.

?

There must be some path from p to q.

?

There must be some path from p to q.

?

There must be some path from p to q.

Real paths mostly use the networks, but have no long detours.

?

There must be some path from p to q.

Real paths mostly use the networks, but have no long detours.

?

Finding a reasonable path for each pair of locations is expensive.

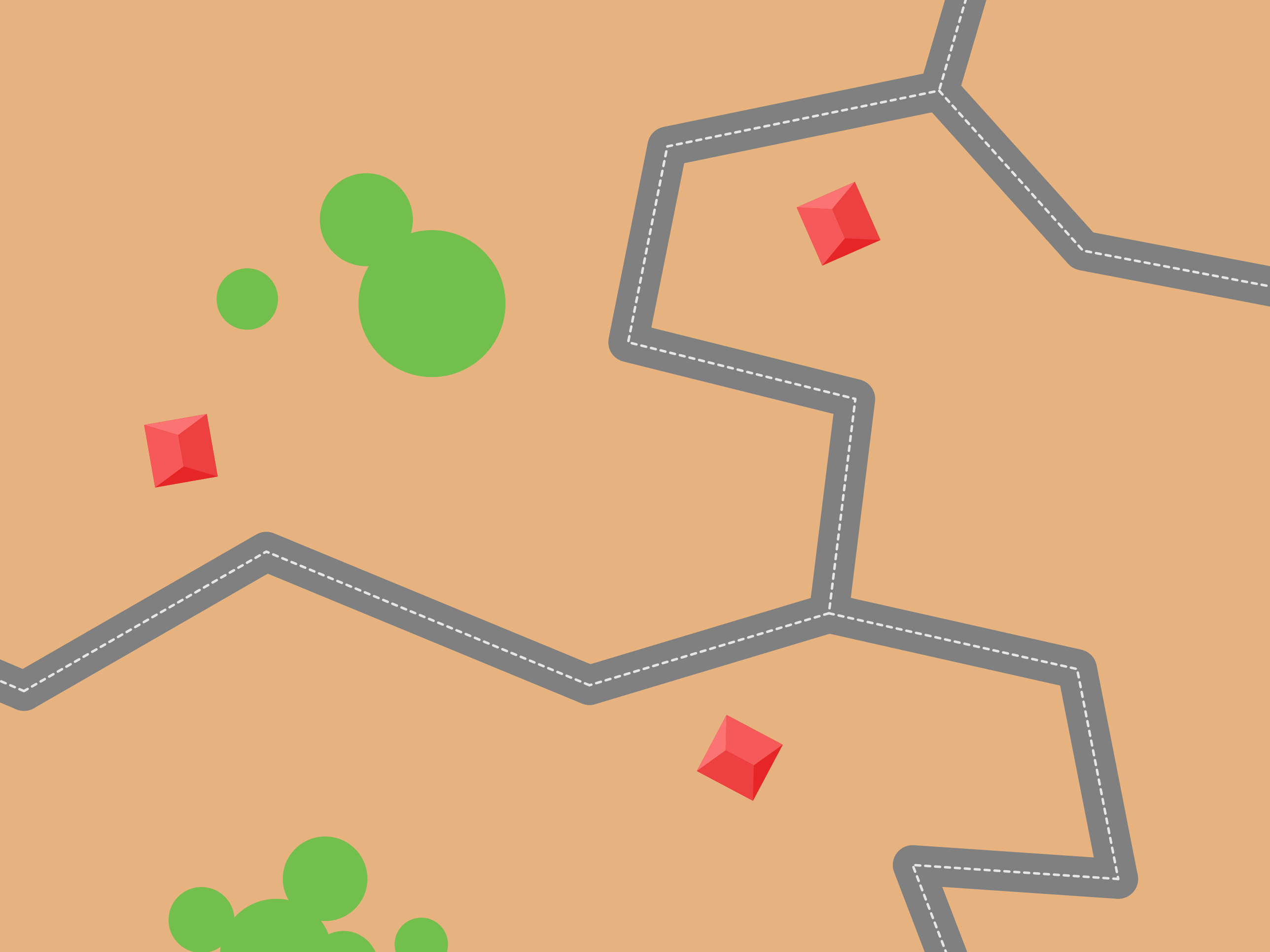
There must be some path from p to q.

Real paths mostly use the networks, but have no long detours.

?

Finding a reasonable path for each pair of locations is expensive.

What can we do?



Possible solution:
connect the points to
the network with
‘virtual’ edges.

Possible solution:
connect the points to
the network with
‘virtual’ edges.

Possible solution: connect the points to the network with ‘virtual’ edges.

We call such new connections *feed links*.

Possible solution: connect the points to the network with ‘virtual’ edges.

We call such new connections *feed links*.

The resulting network may not be a very good estimation of the true distances.

Possible solution: connect the points to the network with ‘virtual’ edges.

We call such new connections *feed links*.

The resulting network may not be a very good estimation of the true distances.

Can we connect the points to the network in a reasonable way?

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

We want to use only a limited number of feed links to join each point to its face.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

We want to use only a limited number of feed links to join each point to its face.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

We want to use only a limited number of feed links to join each point to its face.

Can we connect the points to the network in a reasonable way?

We may need good connections to many other points.

We want to use only a limited number of feed links to join each point to its face.

How do we capture the quality of the feed links?

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

The *dilation* between two points q and q is:

$$\frac{\text{shortestpath}(p, q)}{\text{distance}(p, q)}$$

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

The *dilation* between two points q and q is:

$$\frac{\text{shortestpath}(p, q)}{\text{distance}(p, q)}$$

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

The *dilation* between two points q and q is:

$$\frac{\text{shortestpath}(p, q)}{\text{distance}(p, q)}$$

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

The *dilation* between two points q and q is:

$$\frac{\text{shortestpath}(p, q)}{\text{distance}(p, q)}$$

How do we capture the quality of the feed links?

We want to add feed links that do not induce long detours.

For this point, the dilation is 7.

The *dilation* between two points q and q is:

$$\frac{\text{shortestpath}(p, q)}{\text{distance}(p, q)}$$

How do we capture the quality of the feed links?

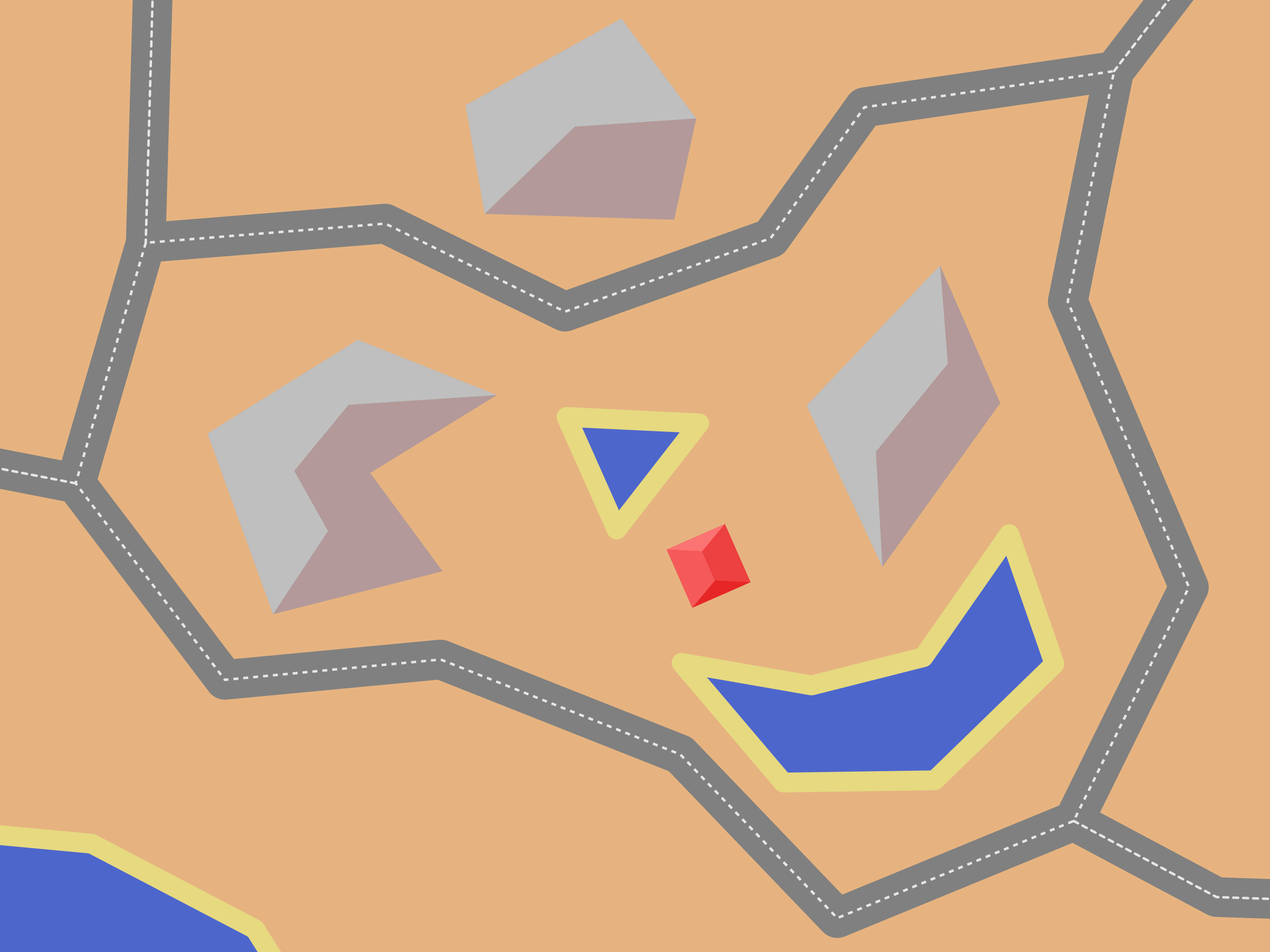
We want to add feed links that do not induce long detours.

For this point, the dilation is 7.

In real road networks, the dilation is typically around $1\frac{1}{2}$.

The *dilation* between two points q and q is:

$$\frac{\text{shortestpath}(p, q)}{\text{distance}(p, q)}$$



Usually, not all land can be walked on.

Usually, not all land can be walked on.

In the presence of obstacles, we can still define the dilation in the same way.

Usually, not all land can be walked on.

In the presence of obstacles, we can still define the dilation in the same way.

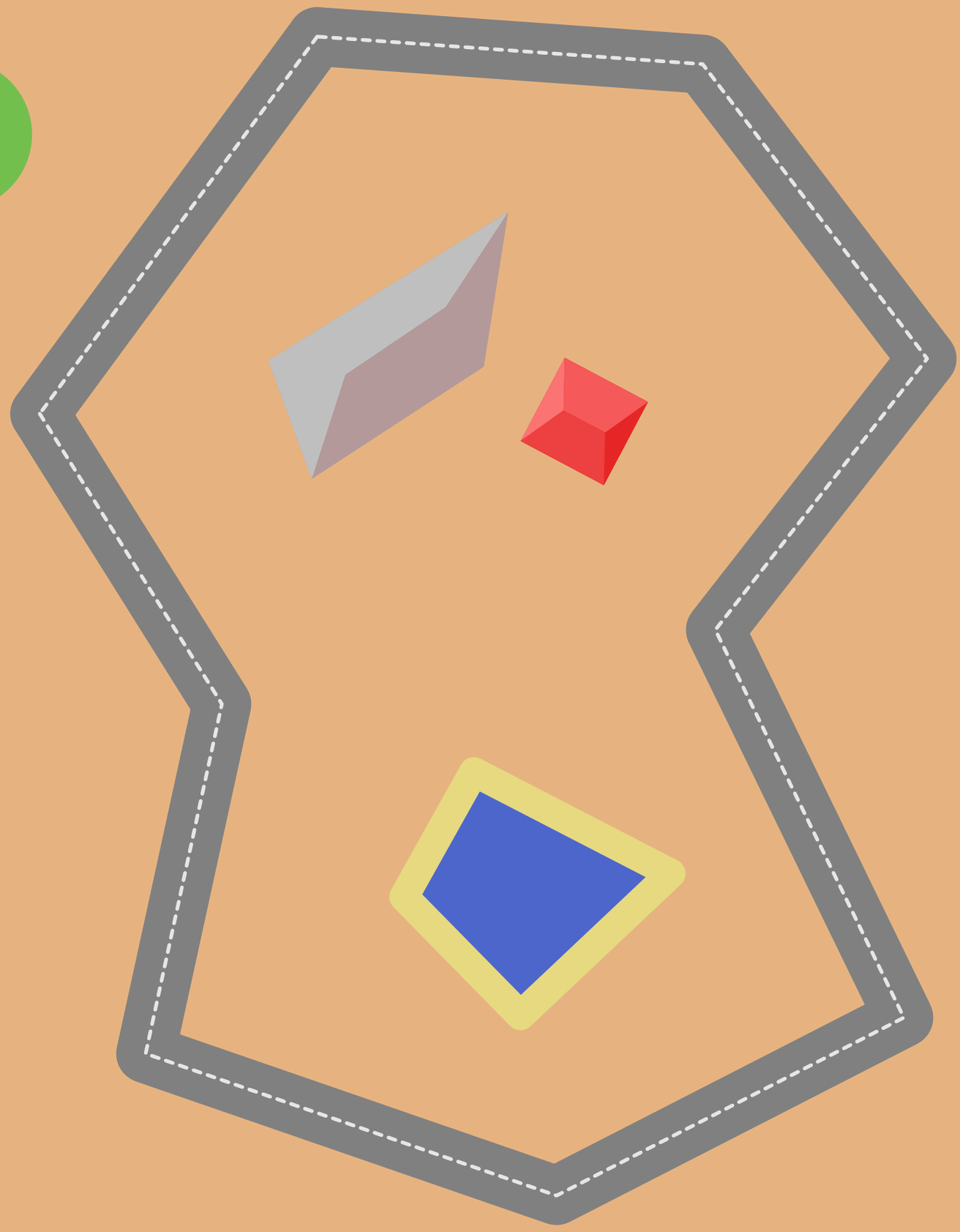
Usually, not all land can be walked on.

In the presence of obstacles, we can still define the dilation in the same way.

Problem definition.

Given are a polygon P, a set of obstacles $\mathcal{B}$, and a point p inside P.

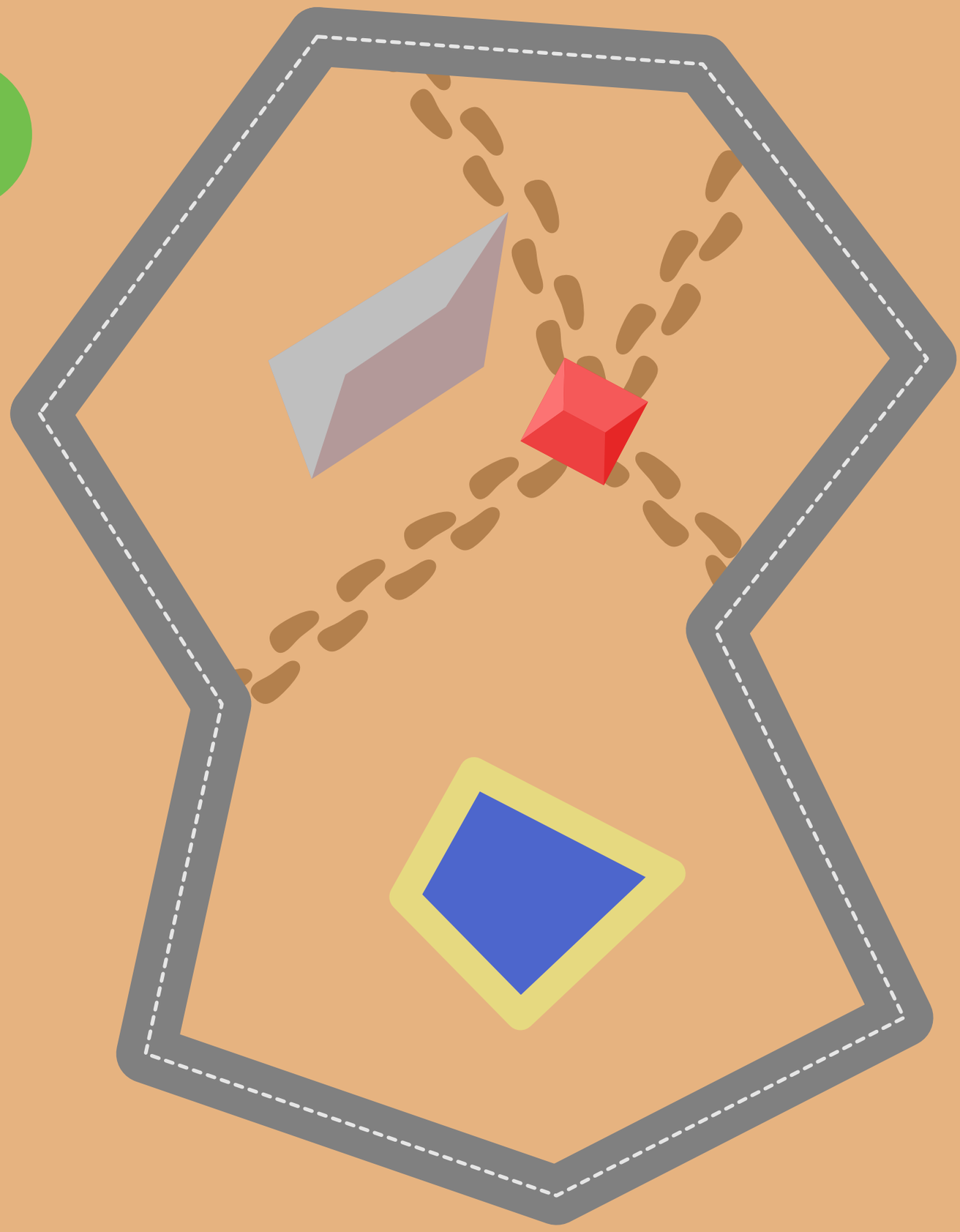
Add k feed links from p to P such that the worst dilation between p and any point on P is minimal.



Problem definition.

Given are a polygon P, a set of obstacles $\mathcal{B}$, and a point p inside P.

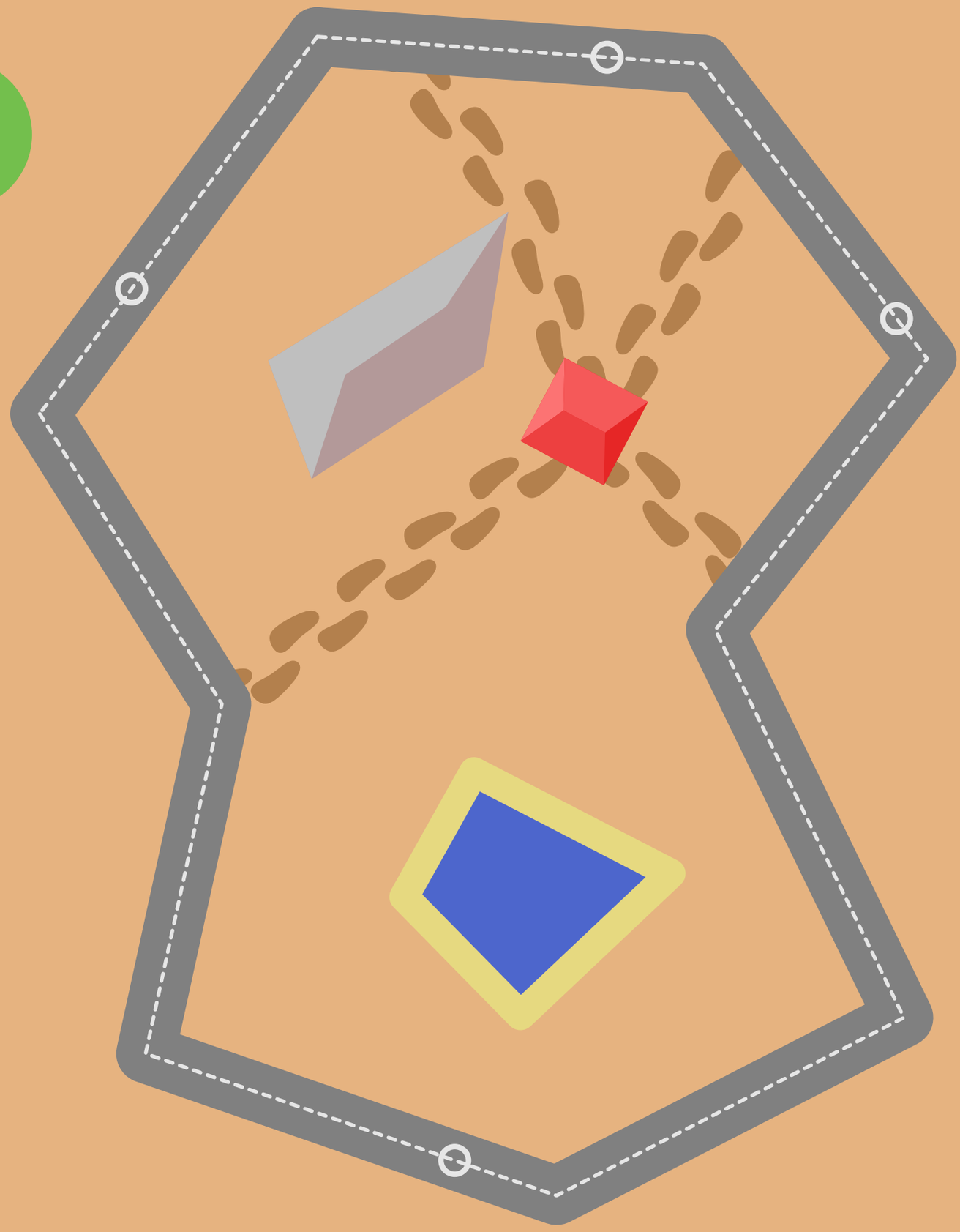
Add k feed links from p to P such that the worst dilation between p and any point on P is minimal.



Problem definition.

Given are a polygon P, a set of obstacles $\mathcal{B}$, and a point p inside P.

Add k feed links from p to P such that the worst dilation between p and any point on P is minimal.

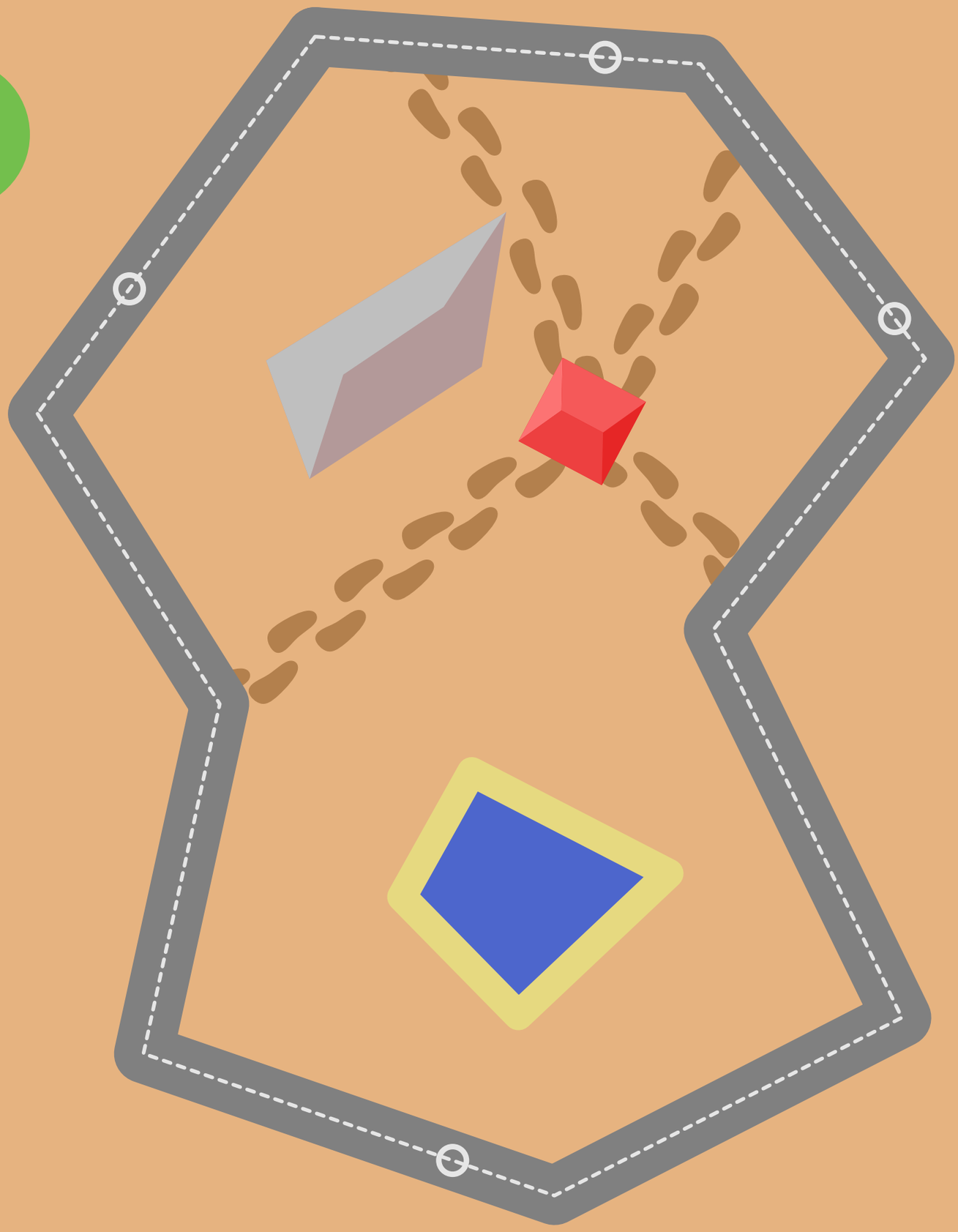


Problem definition.

Given are a polygon P, a set of obstacles $\mathcal{B}$, and a point p inside P.

Add k feed links from p to P such that the worst dilation between p and any point on P is minimal.

We cannot solve the problem exactly in reasonable time.



What results do we have?

What results do we have?

Given a set of feed links, we can *compute* the worst dilation.

What results do we have?

Given a set of feed links, we can *compute* the worst dilation.

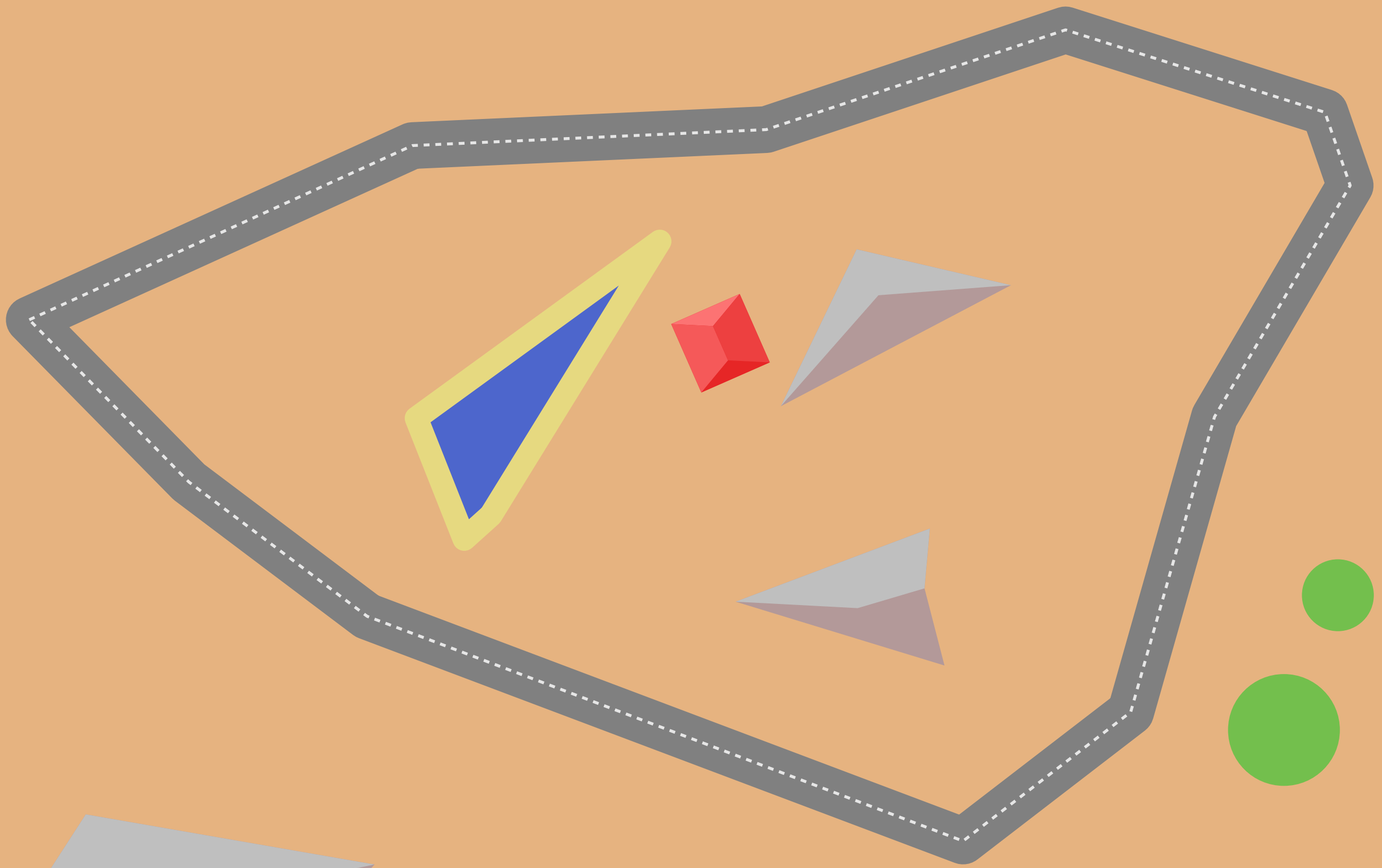
Given a dilation treshold, we can compute $OPT + 1$ feed links that obtain this.

What results do we have?

Given a set of feed links, we can *compute* the worst dilation.

Given a dilation treshold, we can compute $OPT + 1$ feed links that obtain this.

We implemented two heuristics for placing k feed links, and analysed the results.



In the *greedy* heuristic, the first feed link is placed to the closest point on P.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

Each subsequent feed link is placed to the point with the worst dilation so far.

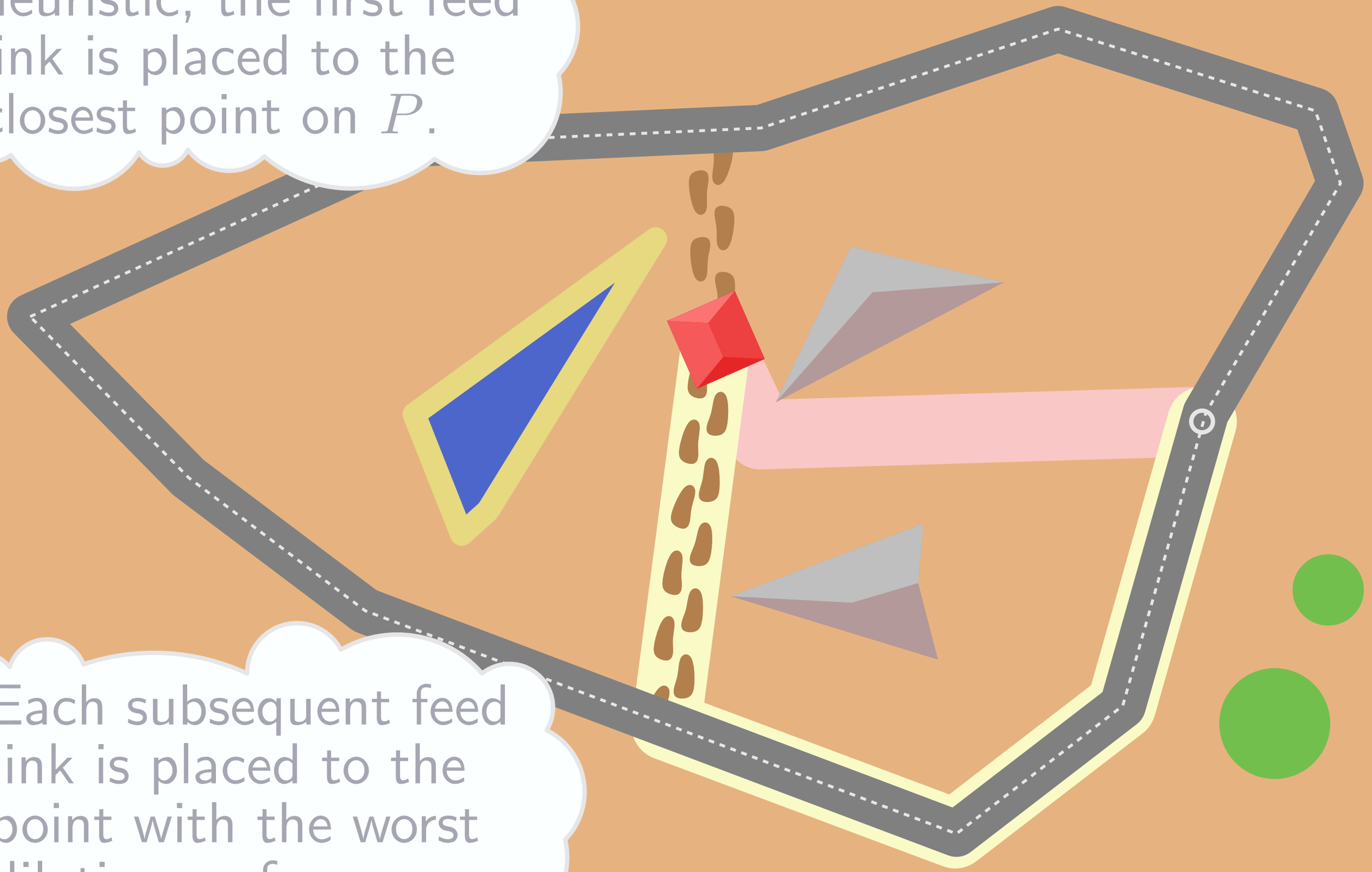
In the *greedy* heuristic, the first feed link is placed to the closest point on P.

Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

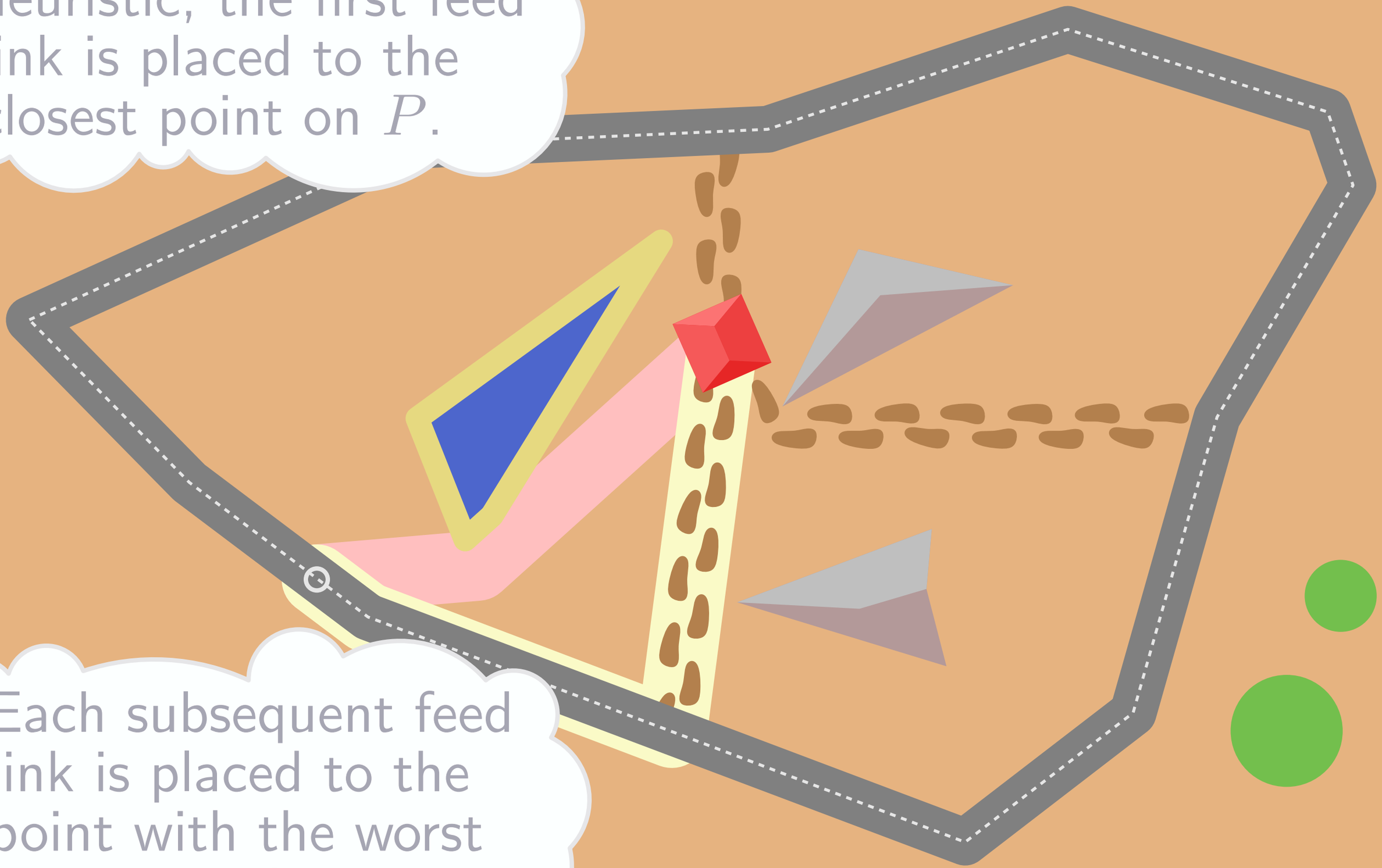


Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

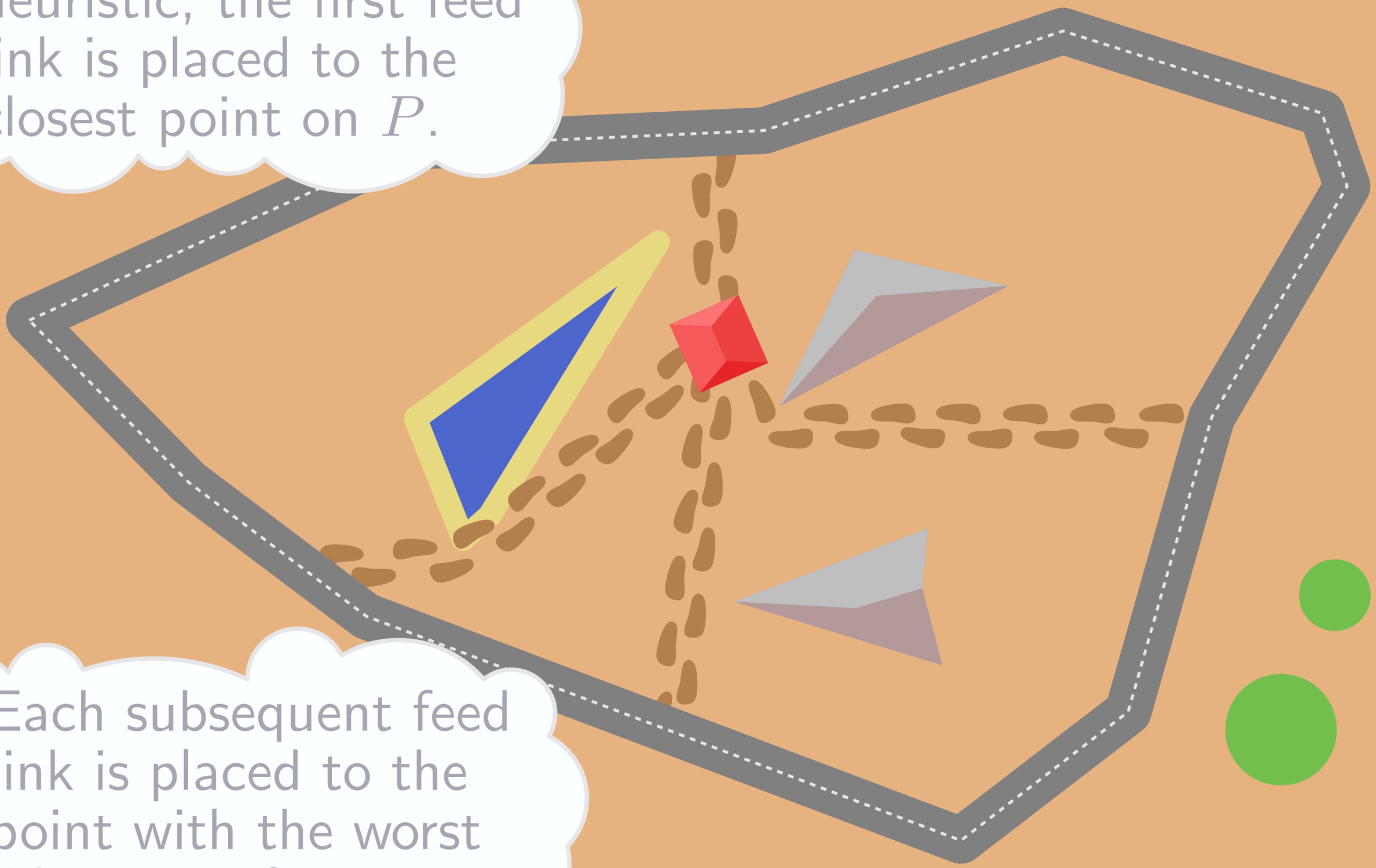
Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.



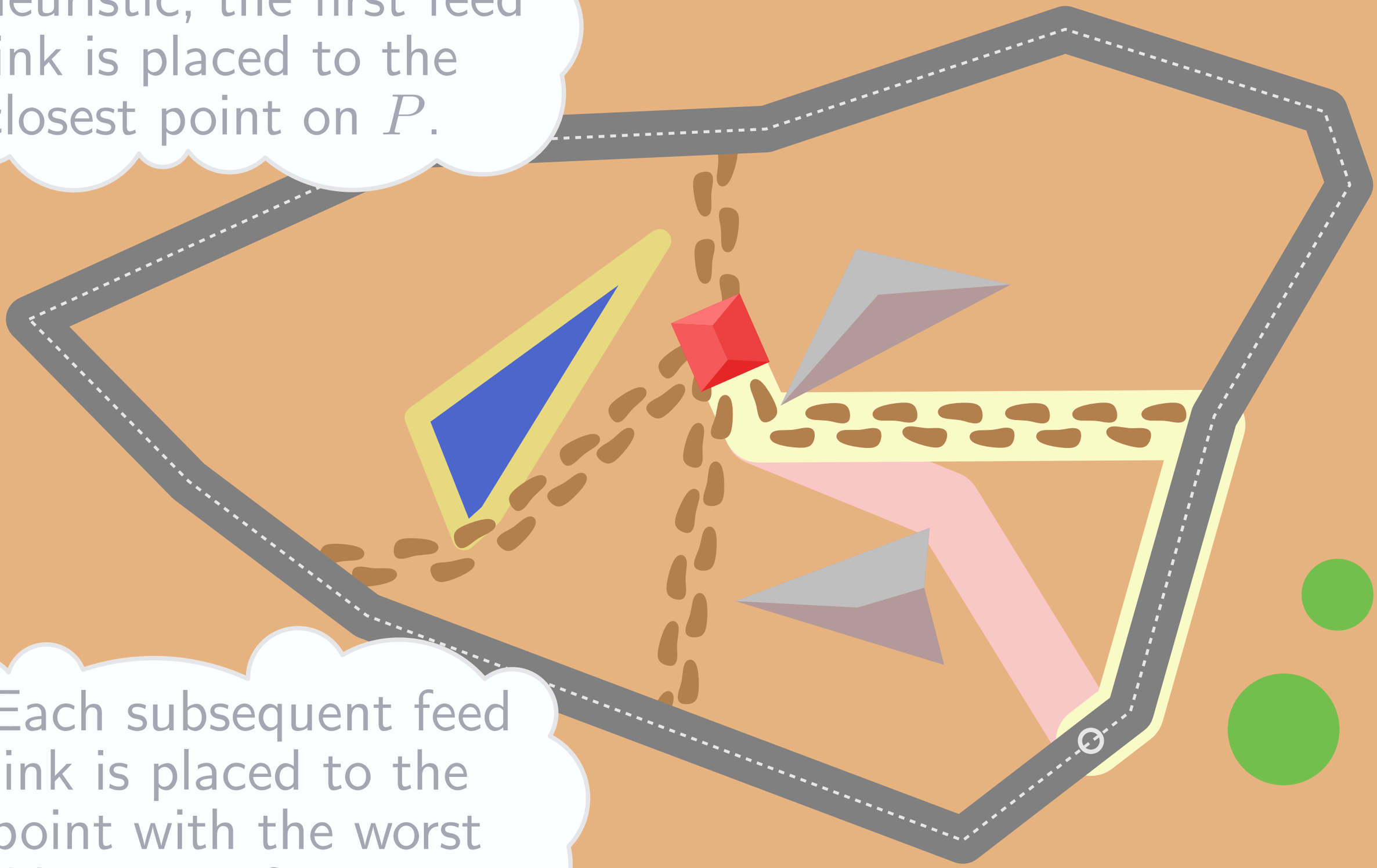
Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.



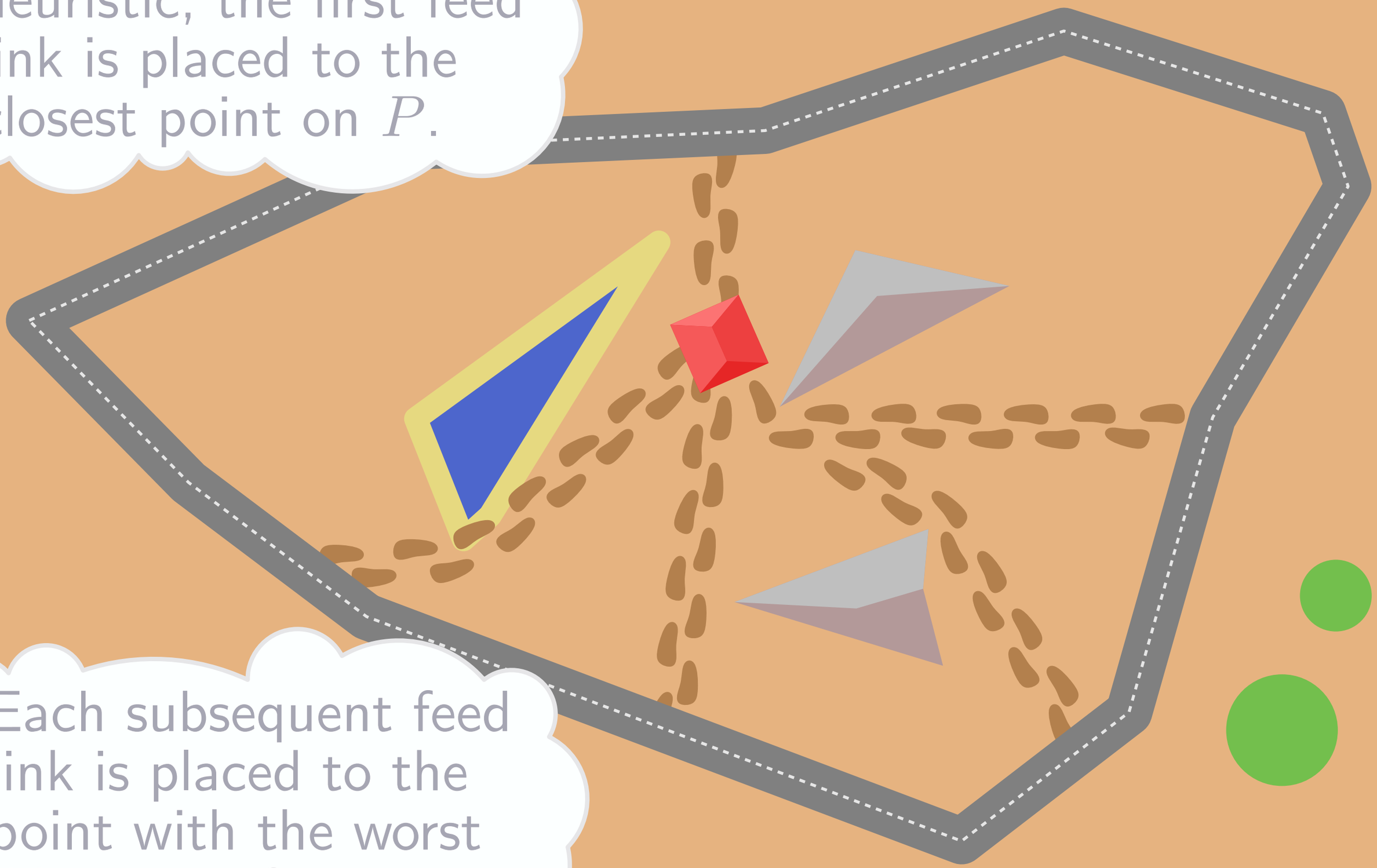
Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

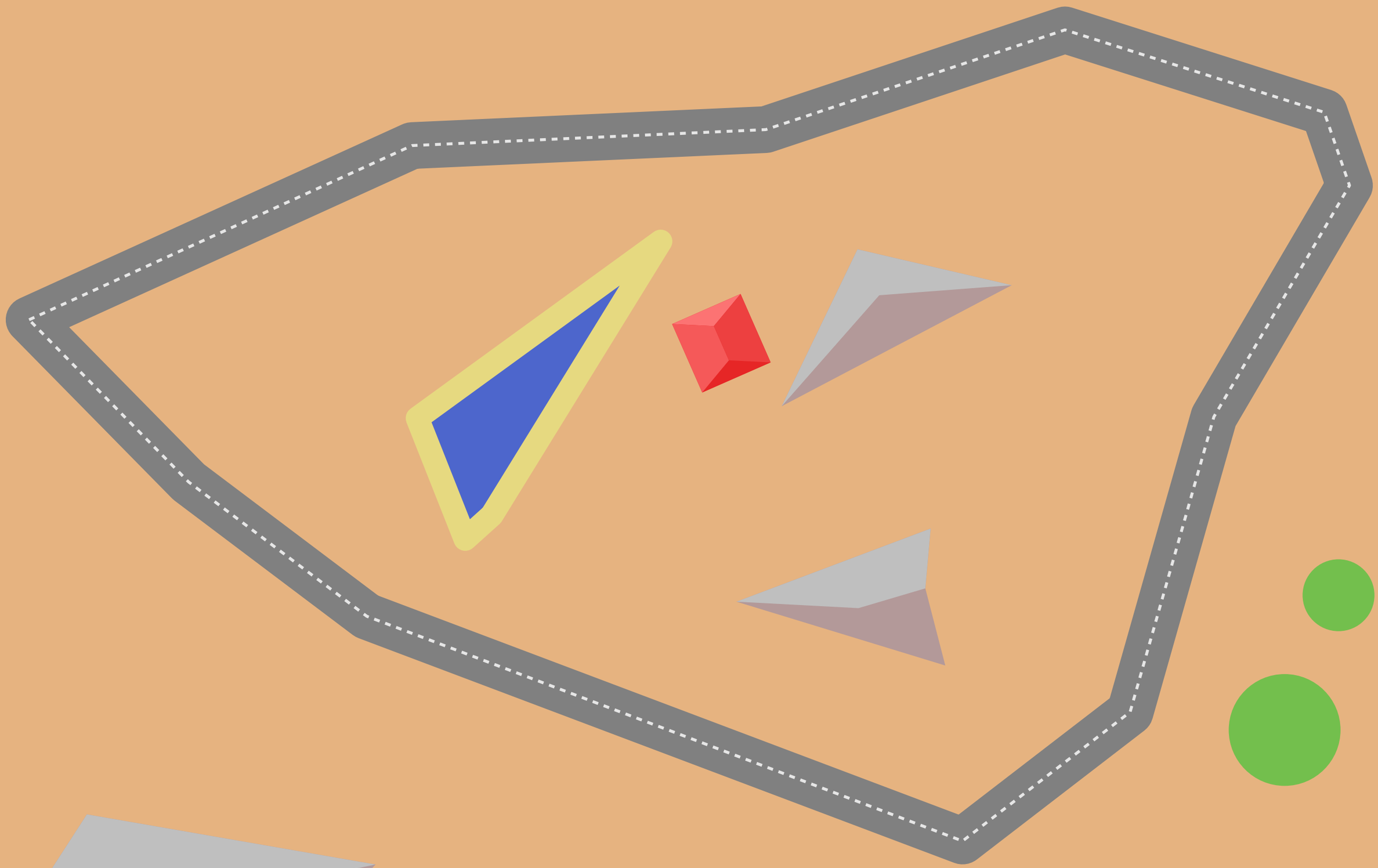


Each subsequent feed link is placed to the point with the worst dilation so far.

In the *greedy* heuristic, the first feed link is placed to the closest point on P.

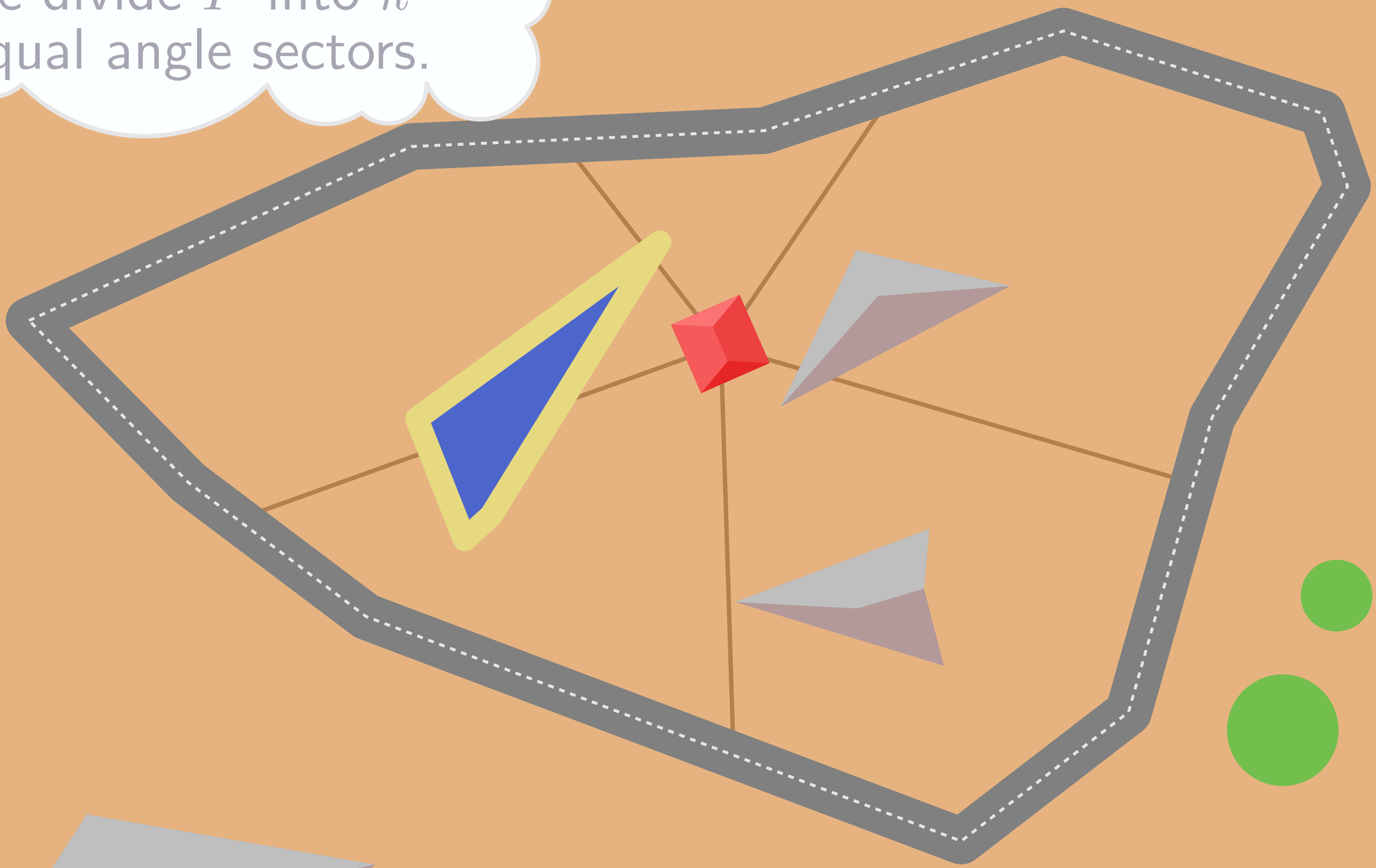


Each subsequent feed link is placed to the point with the worst dilation so far.



In the *sector* heuristic, we divide P into k equal angle sectors.

In the *sector* heuristic, we divide P into k equal angle sectors.



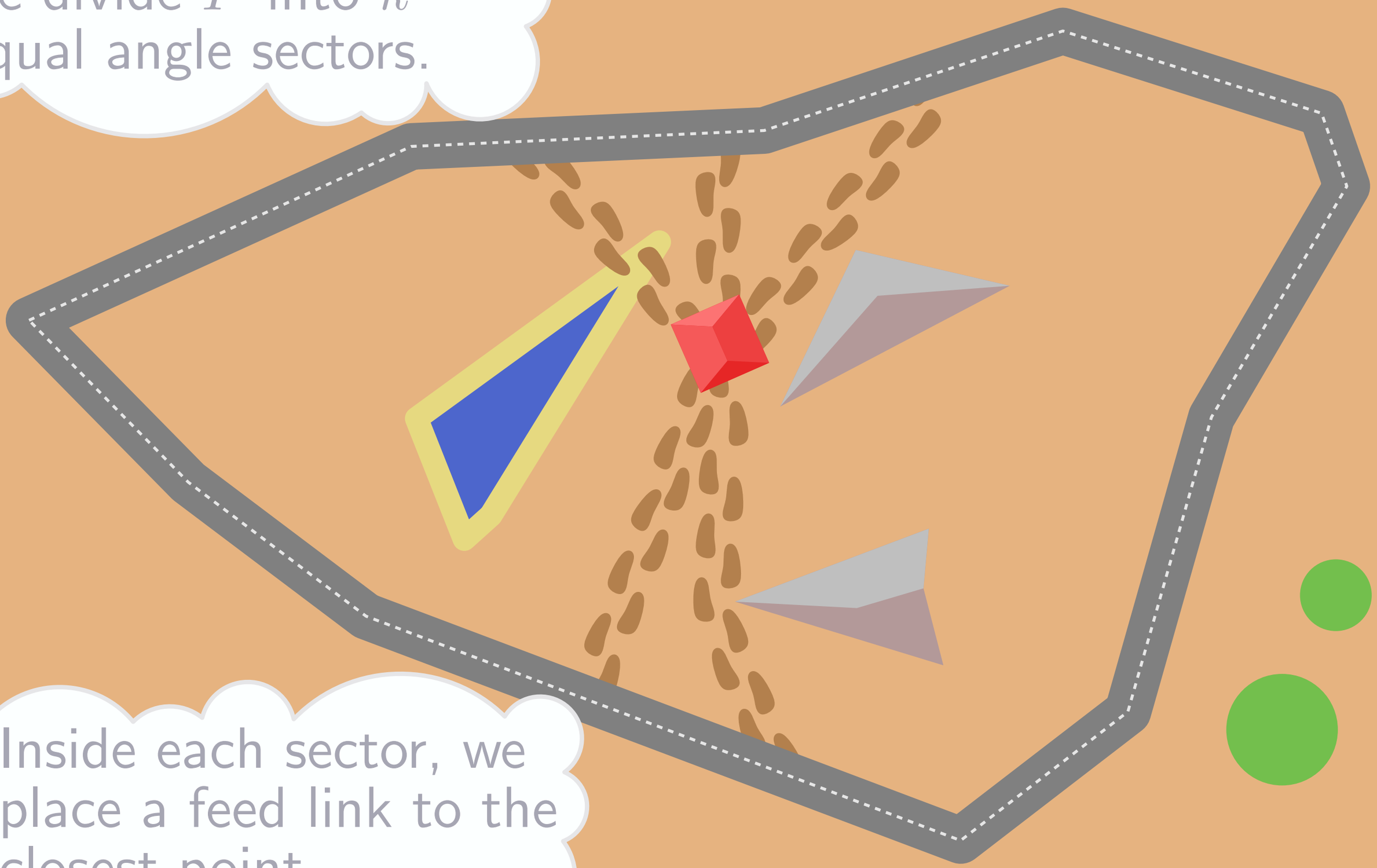
In the *sector* heuristic, we divide P into k equal angle sectors.

Inside each sector, we place a feed link to the closest point.

In the *sector* heuristic, we divide P into k equal angle sectors.

Inside each sector, we place a feed link to the closest point.

In the *sector* heuristic, we divide P into k equal angle sectors.



Inside each sector, we place a feed link to the closest point.

We performed some experiments.

We performed some experiments.

We generated 100 random polygons with two obstacles inside each.

We performed some experiments.

We generated 100 random polygons with two obstacles inside each.

For each polygon, we placed 1 to 10 feed links with both heuristics.

We performed some experiments.

We generated 100 random polygons with two obstacles inside each.

For each polygon, we placed 1 to 10 feed links with both heuristics.

We computed the point with the worst dilation in each case, and took the average.

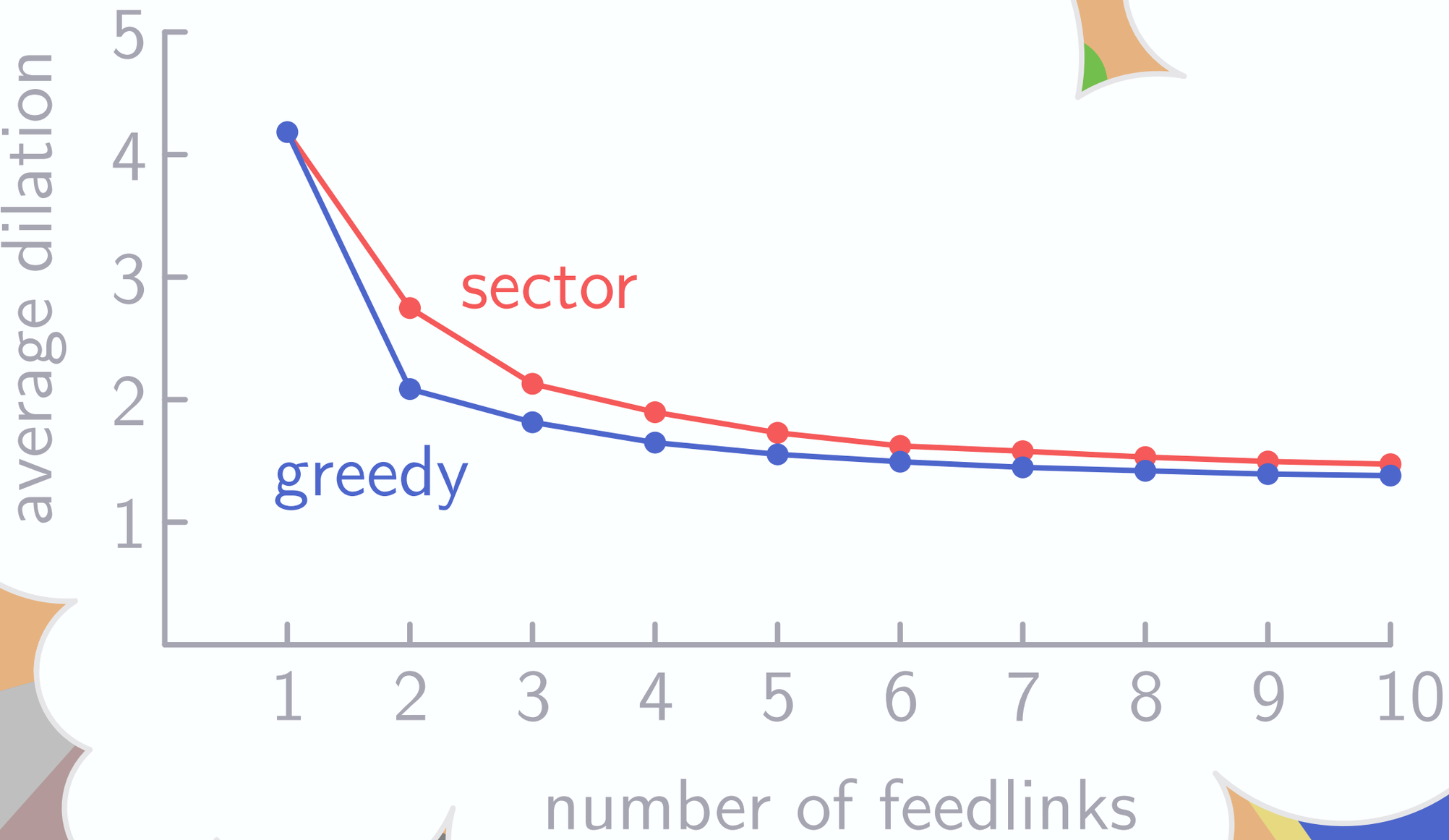
We performed some experiments.

We generated 100 random polygons with two obstacles inside each.

For each polygon, we placed 1 to 10 feed links with both heuristics.

We computed the point with the worst dilation in each case, and took the average.

And the results are...


average dilation
5
4
3
2
1
sector
greedy
1
2
3
4
5
6
7
8
9
10
number of feedlinks

Time to conclude.

Time to conclude.

We tested heuristics to extend a network by several feed links.

Time to conclude.

We tested heuristics to extend a network by several feed links.

Of the two heuristics, the greedy one appears to perform best.

Time to conclude.

We tested heuristics to extend a network by several feed links.

Of the two heuristics, the greedy one appears to perform best.

Both heuristics are easy to implement, and produce more realistic networks than ad-hoc methods that do not take dilation into account.

Thank you!

Rodrigo I. Silveira

Maike Buchin

Maarten Löffler

Boris Aronov

Tom de Jong

Thank you!

Jun Luo

Bettina Speckmann

Bart Jansen

Kevin Buchin

Marc van Kreveld