Surrogate-based optimization

In the computer experiments literature, folks had been using GPs to optimize functions for some time,

however, the best reference for the core idea might be

The methodology is simple:

Observe that Step 2 involves its own inner-optimization,

The problem

Before we continue, lets be clear about the problem.

We wish to find

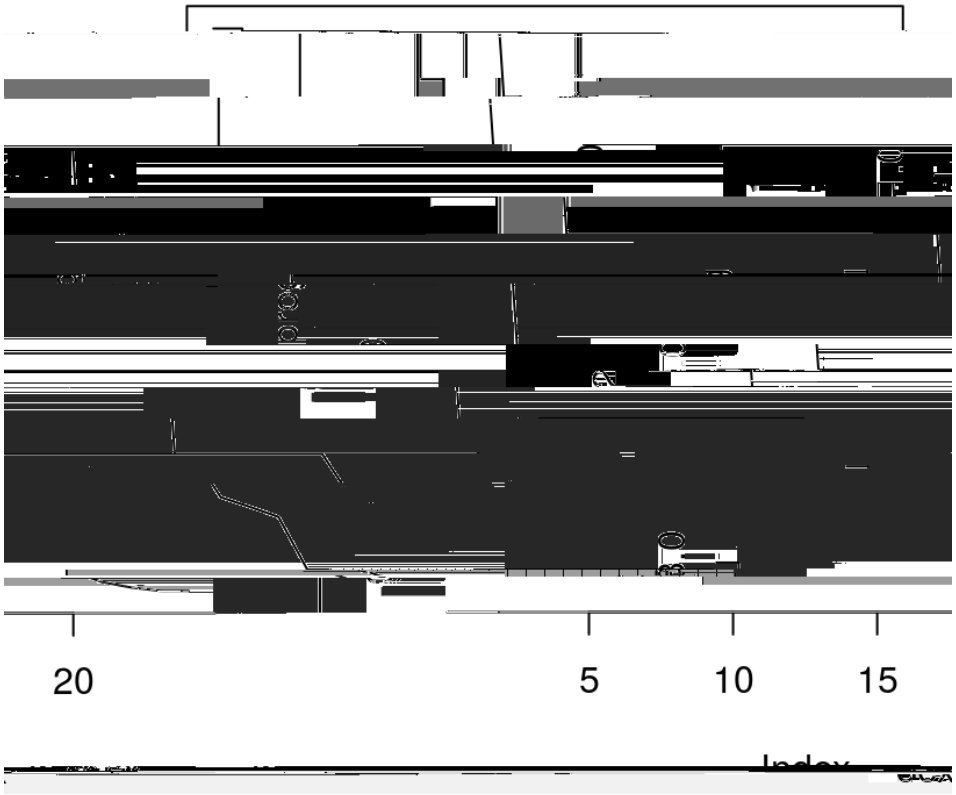
One-step of search

Here's what that looks like in the input domain.

```
plot(X[1:ninit,], xlab="x1", ylab="x2", xlim=c(0,1), ylim=c(0,1))
arrows(X[m,1], X[m,2], opt$par[1], opt$par[2], length=0.1)
```

Progress

```
prog <- y
for(i in
```

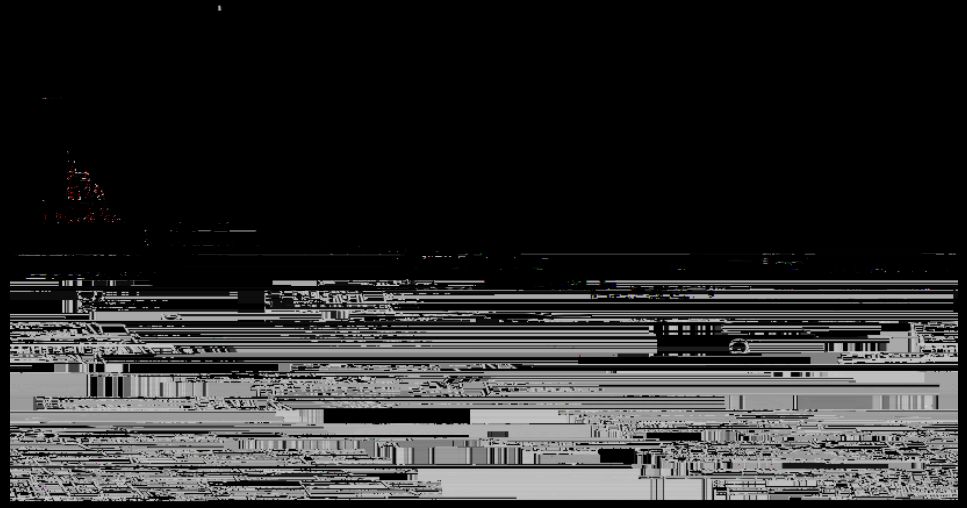


Encapsulating f

Visualizing average progress

Clearly this is not a global optimization tool.

```
matplot(t(prog), type="l", lty=1)
```



Improvmpmm R Y Y Y R S

Analytic expression

The cool thing is that if $f(\quad$ is Gaussian,

the EI has a convenient closed form expression.

Notice how it organically balances

Our o

Multi-start scheme

Although EI has a "maximizing variance" aspect, which could cause the EI surface to be multi-modal

- just like ALM and ALC,
- it will not be as pathologically so.

The number of EI modes will fluctuate as the algorithm runs,

- but eventually it will resemble the actual (number of) modes in .

Therefore a sensible multi-start scheme might include

As

Encapsulation

Ave

Consumo

Known constraints

First, lets keep it simple and assvssvssvss

Multiple real-valued con

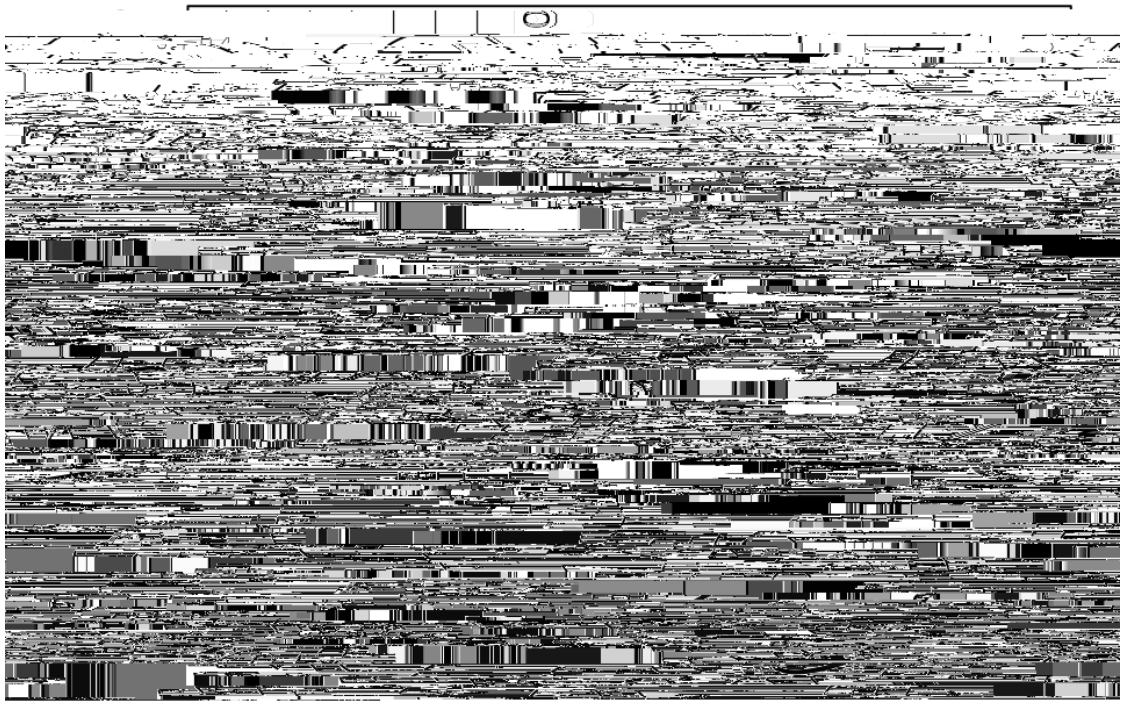
Conve

Visualizing progress

```
plot(efi$prog, type="l", ylim=c(0.6, 1.6), ylab="progress", xlab="evaluations")
lines(ey$prog, col=2, lty=2); lines(ei.mc$prog, col=3, lty=3)
lines(ei.sl$prog, col=4, lty=4); lines(ei.slopt$prog, col=5, lty=5)
legend("topright", c("EFI", "EY", "EI.mc", "EI.sl", "EI.slopt"), col=1:5, lty=1:5)
```

Visualizing

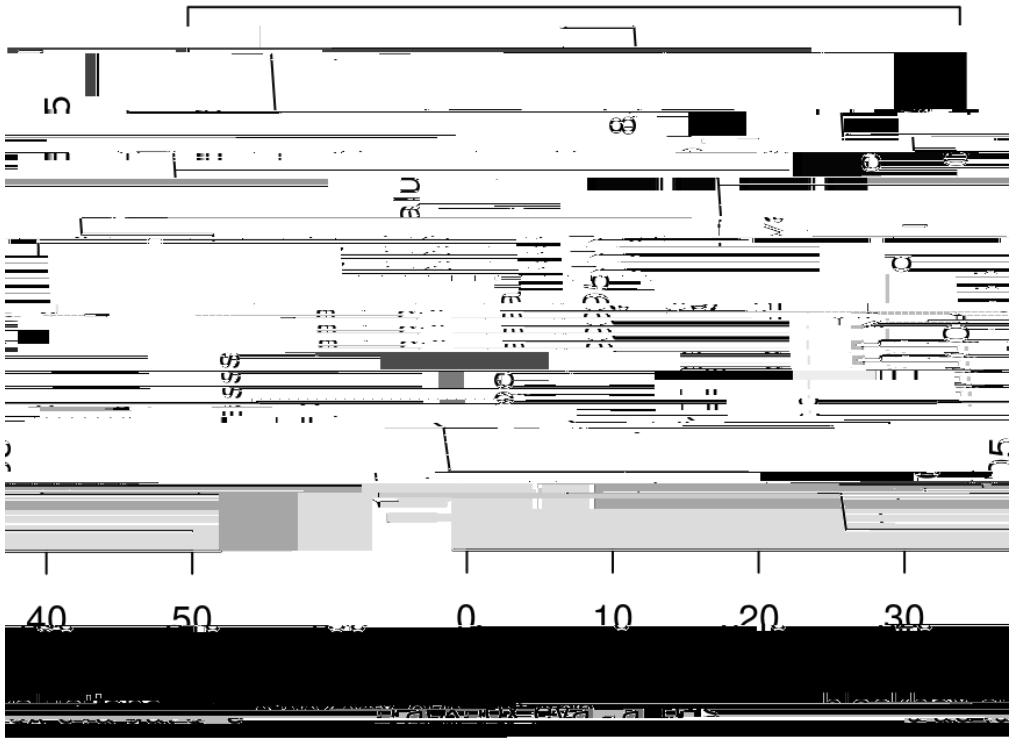
```
plot.aimprob2()
```



Optimizing via EI with AL

Pretty fast progress.

```
out2 <-
```



`demo("GSBP")` Ć such a crazy surface!