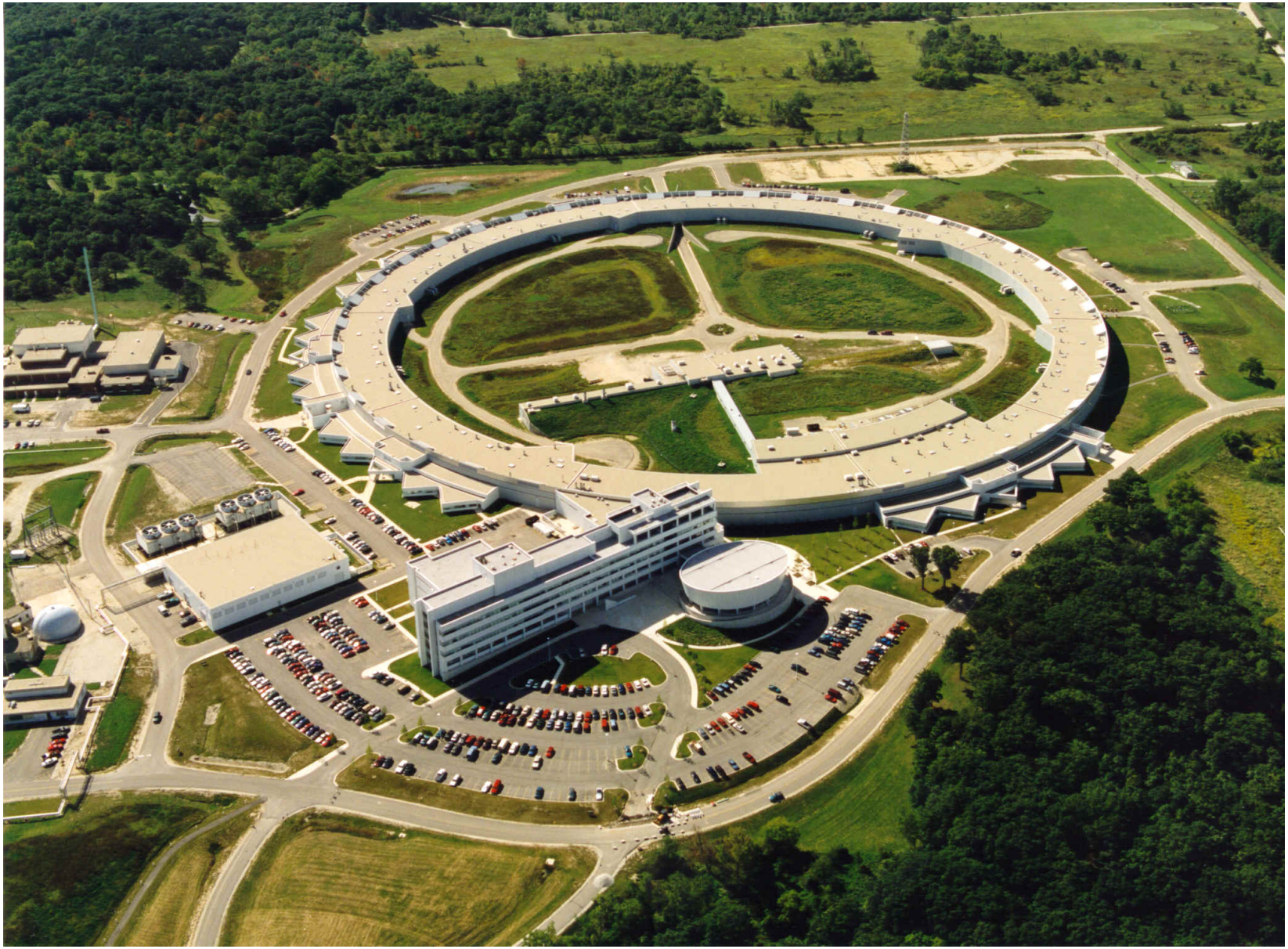
# Multi-start, Model-based Derivative-free Optimization

Jeffrey Larson  Stefan Wild

Argonne National Laboratory

September 10, 2014

# Motivation



$467M to build; tens of millions to operate.

# Motivation

- We want to identify distinct local minimizers of the problem

$$\begin{aligned} \text{minimize}\, f(x) \\ l \leq x \leq u \\ x \in \mathbb{R}^n \end{aligned}$$

- The simulation $f$ is already using parallel resources, it may not scale to the entire machine.

- We do not expect hundreds of meaningful local minima.

# Multi-Level Single Linkage

A multistart method with some local optimization routine $\mathcal{L}$:

**Algorithm 1:** MLSL

**for** $k = 1, 2, \ldots$ **do**

Evaluate $f$ at $N$ random points drawn uniformly from $\mathcal{D}$.

Start $\mathcal{L}$ at any previously evaluated point $x$:

- that is not a local minima
- $\nexists x_i : \|x - x_i\| \leq r_k$ and $f(x_i) < f(x)$

**end**

# Multi-Level Single Linkage

If the simulation $f$, and local optimization method $\mathcal{L}$, and $r_k$ satisfy some assumptions, then MLSL has nice theoretical properties.

- $f$
  - Twice continuously differentiable
  - All local minima are interior points
  - Some positive distance between all minima

# Multi-Level Single Linkage

If the simulation $f$, and local optimization method $\mathcal{L}$, and $r_k$ satisfy some assumptions, then MLSL has nice theoretical properties.

- $f$
  - Twice continuously differentiable
  - All local minima are interior points
  - Some positive distance between all minima
- $\mathcal{L}$
  - Strictly descent
  - Converges to nearby minimum (not stationary point)

# Multi-Level Single Linkage

If the simulation $f$, and local optimization method $\mathcal{L}$, and $r_k$ satisfy some assumptions, then MLSL has nice theoretical properties.

- $f$
  - Twice continuously differentiable
  - All local minima are interior points
  - Some positive distance between all minima
- $\mathcal{L}$
  - Strictly descent
  - Converges to nearby minimum (not stationary point)
- $r_k$

$$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\Gamma\left(1 + \frac{n}{2}\right) \operatorname{vol}(\mathcal{D}) \frac{\sigma \log kN}{kN}} \tag{1}$$

# Multi-Level Single Linkage

**Theorem 8[1]**

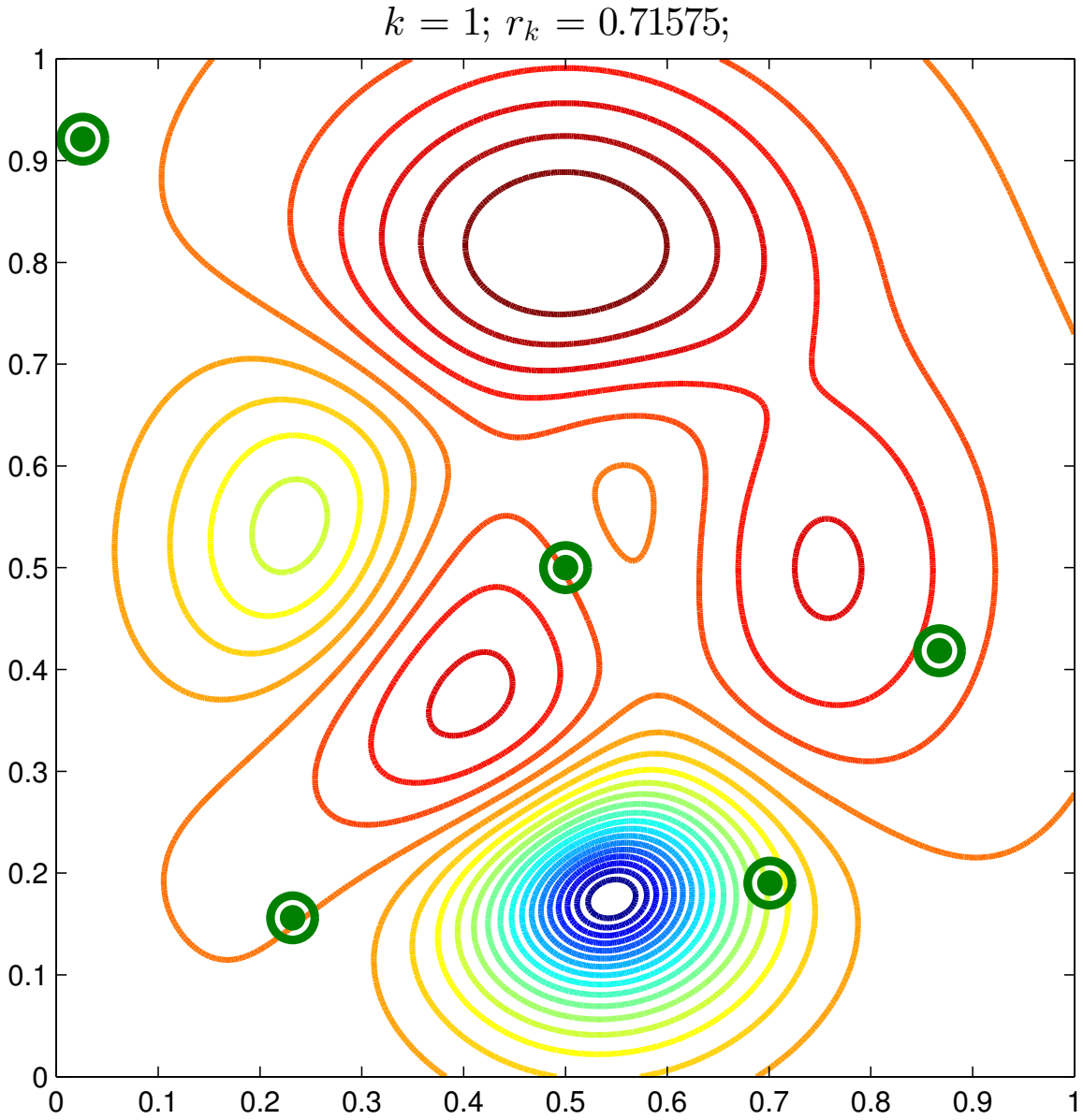
If $r_k$ is defined by (1) and $\sigma > 2$, then the probability that $\mathcal{L}$ is started at iteration $k$ tends to $0$ with increasing $k$.
If $\sigma > 4$, then, even if the sampling continues forever, the total number of local searches ever started is finite with probability 1.

**Theorem 12[1]**

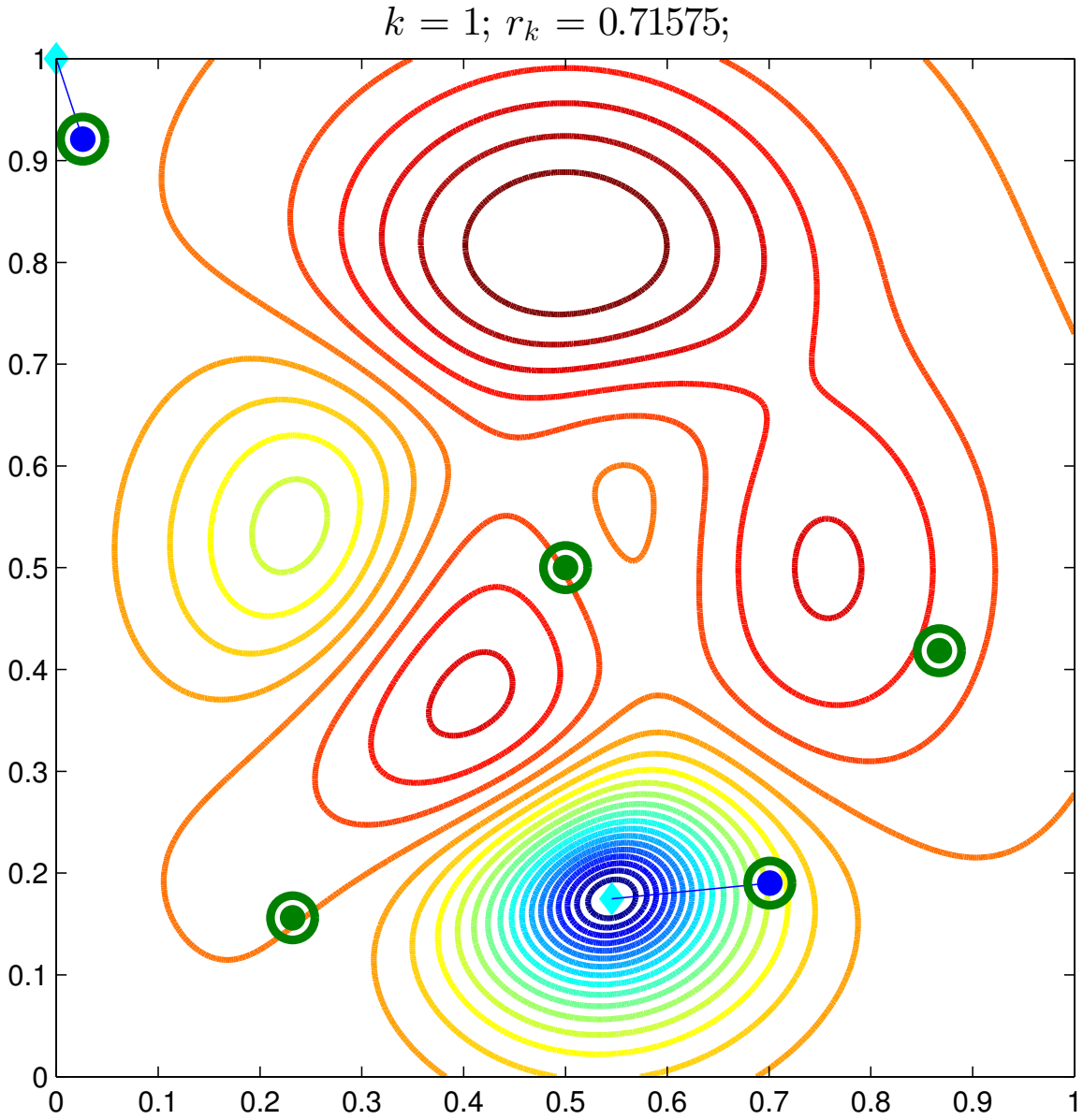
If $r_k \to 0$, all local minima will be found.

[1] A. H. G. Rinnooy Kan and G. T. Timmer. "Stochastic Global Optimization Methods Part II: Multi Level Methods." *Mathematical Programming*, 39(1):57–78, Sept. 1987.

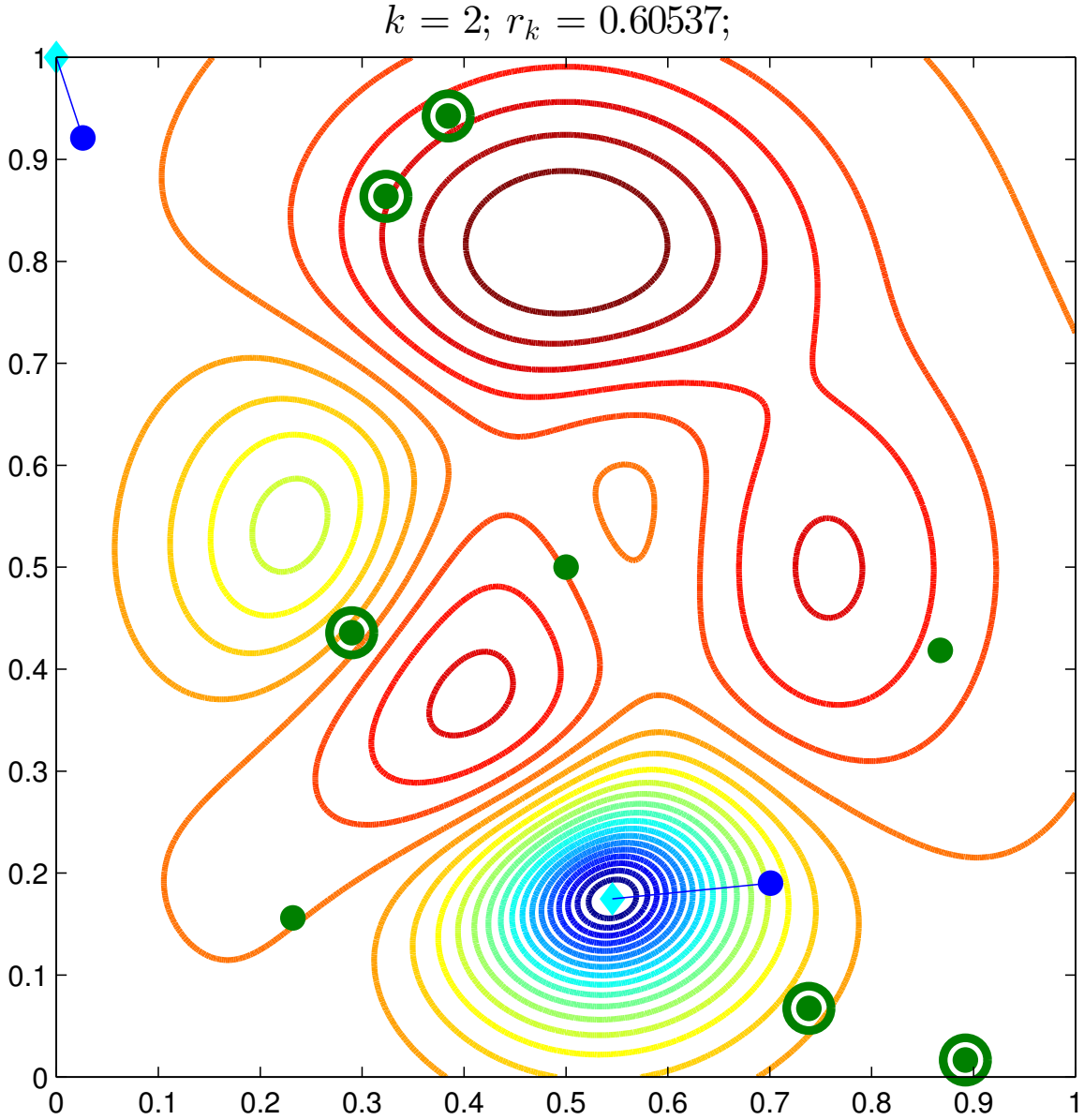
# Multi-Level Single Linkage



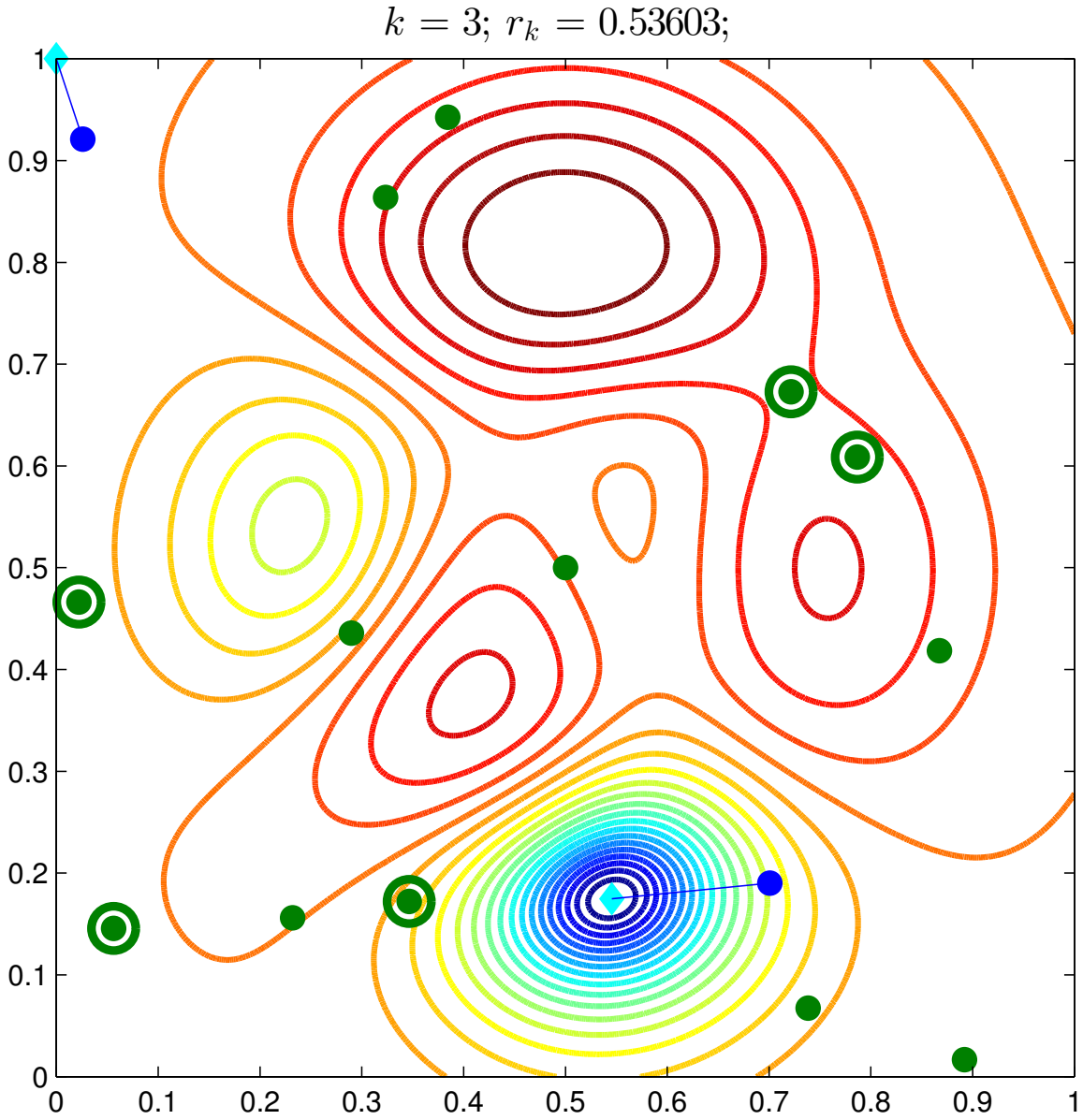
# Multi-Level Single Linkage



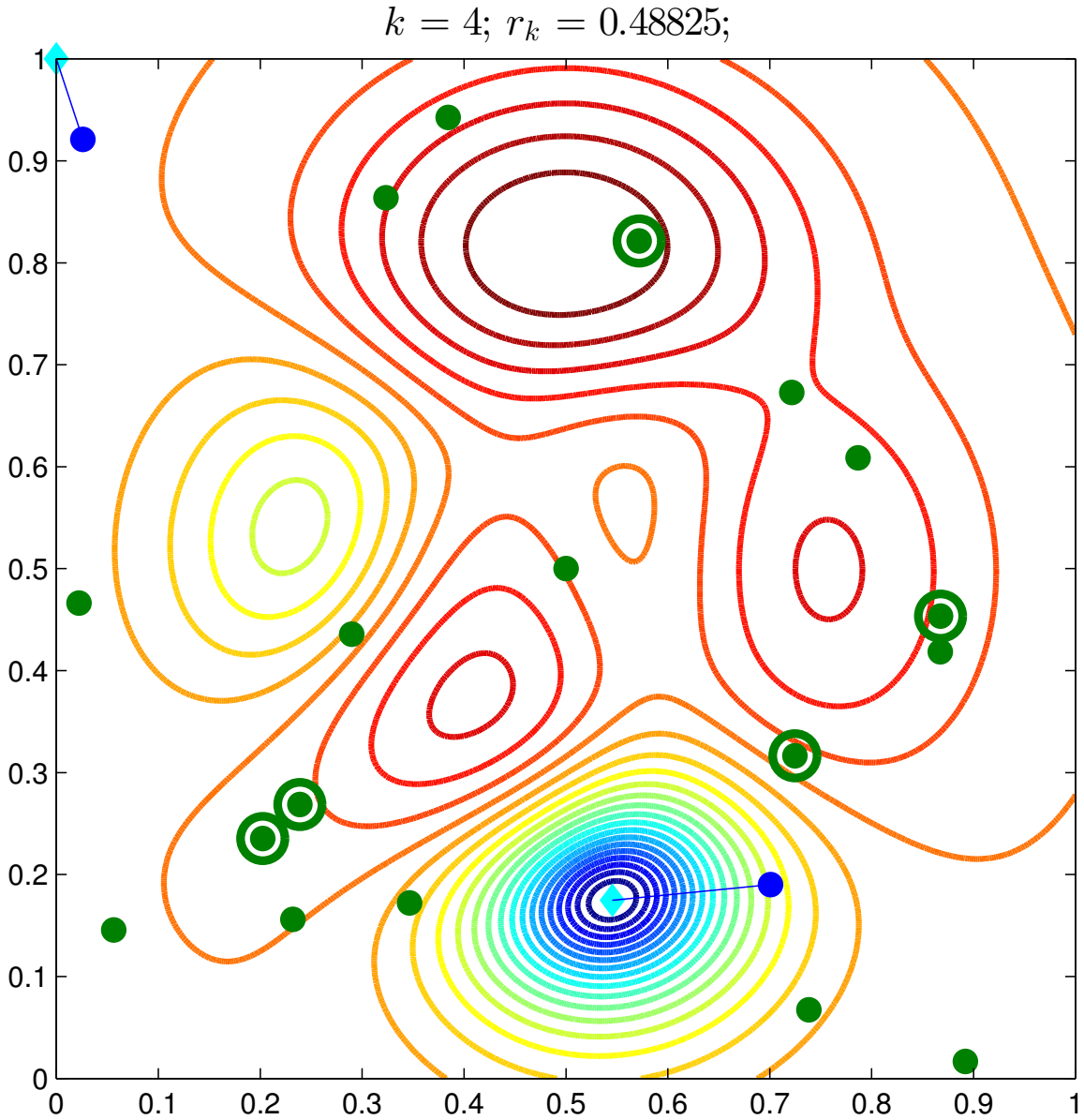
# Multi-Level Single Linkage



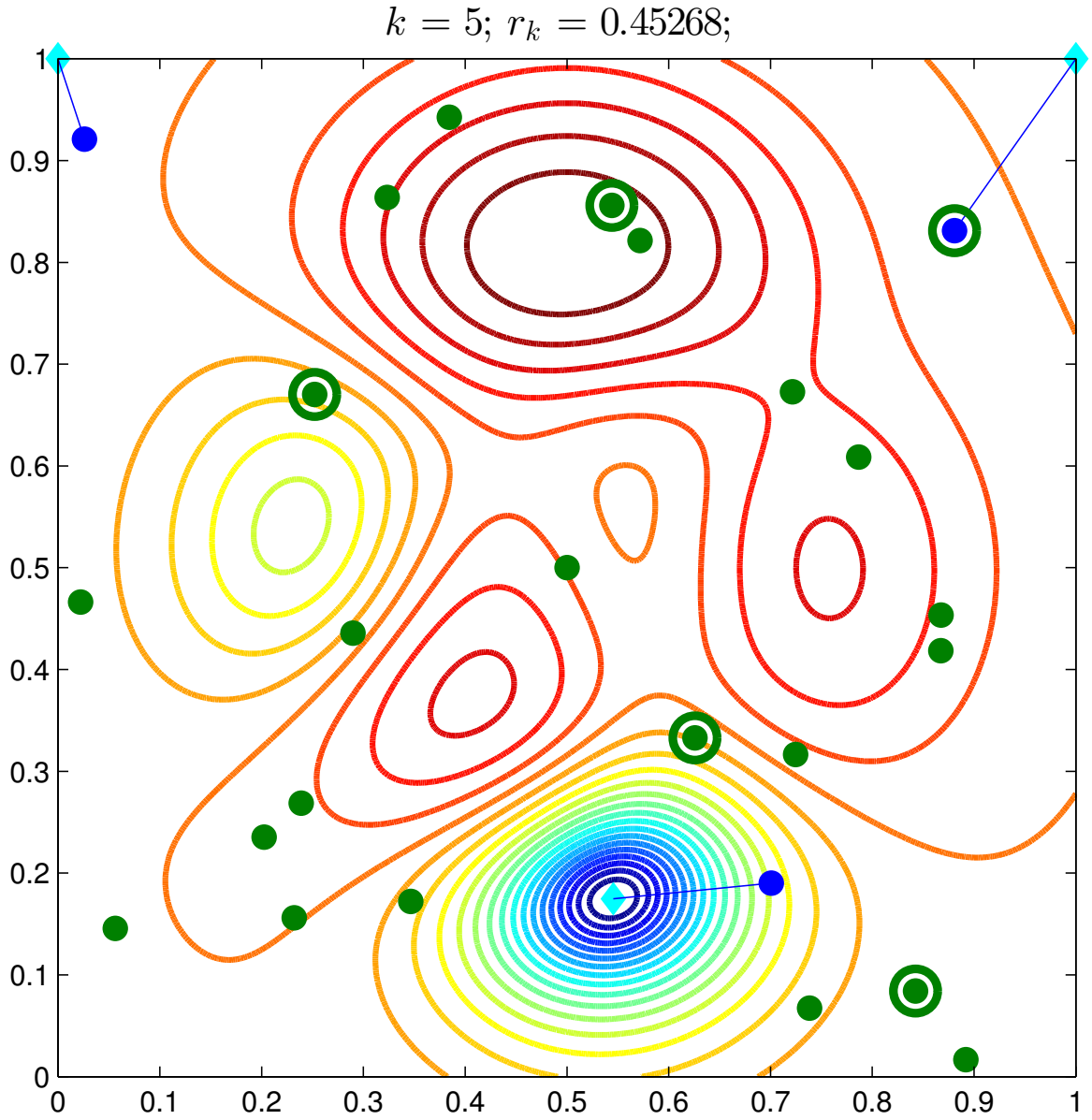
# Multi-Level Single Linkage



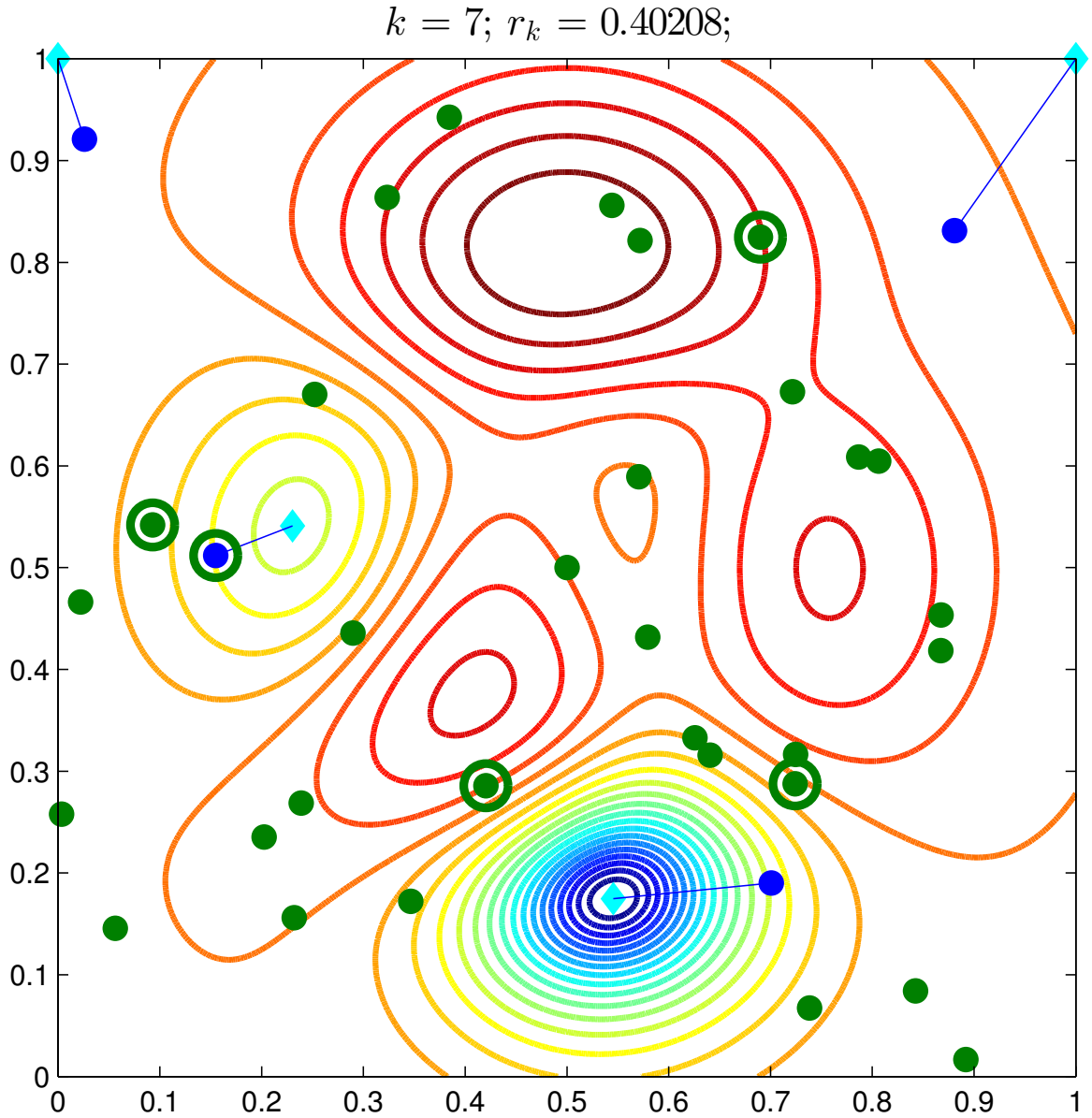
# Multi-Level Single Linkage



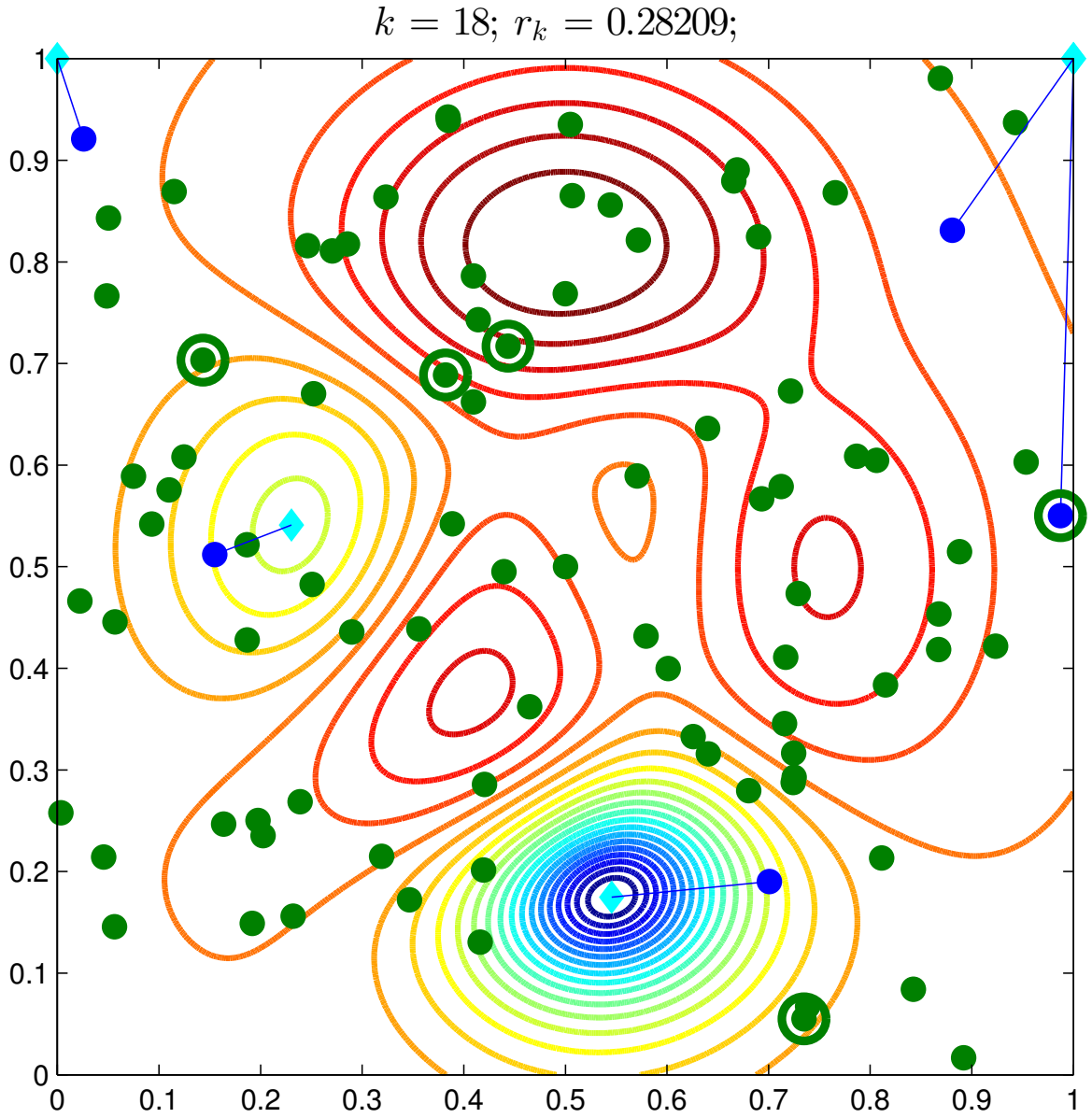
# Multi-Level Single Linkage



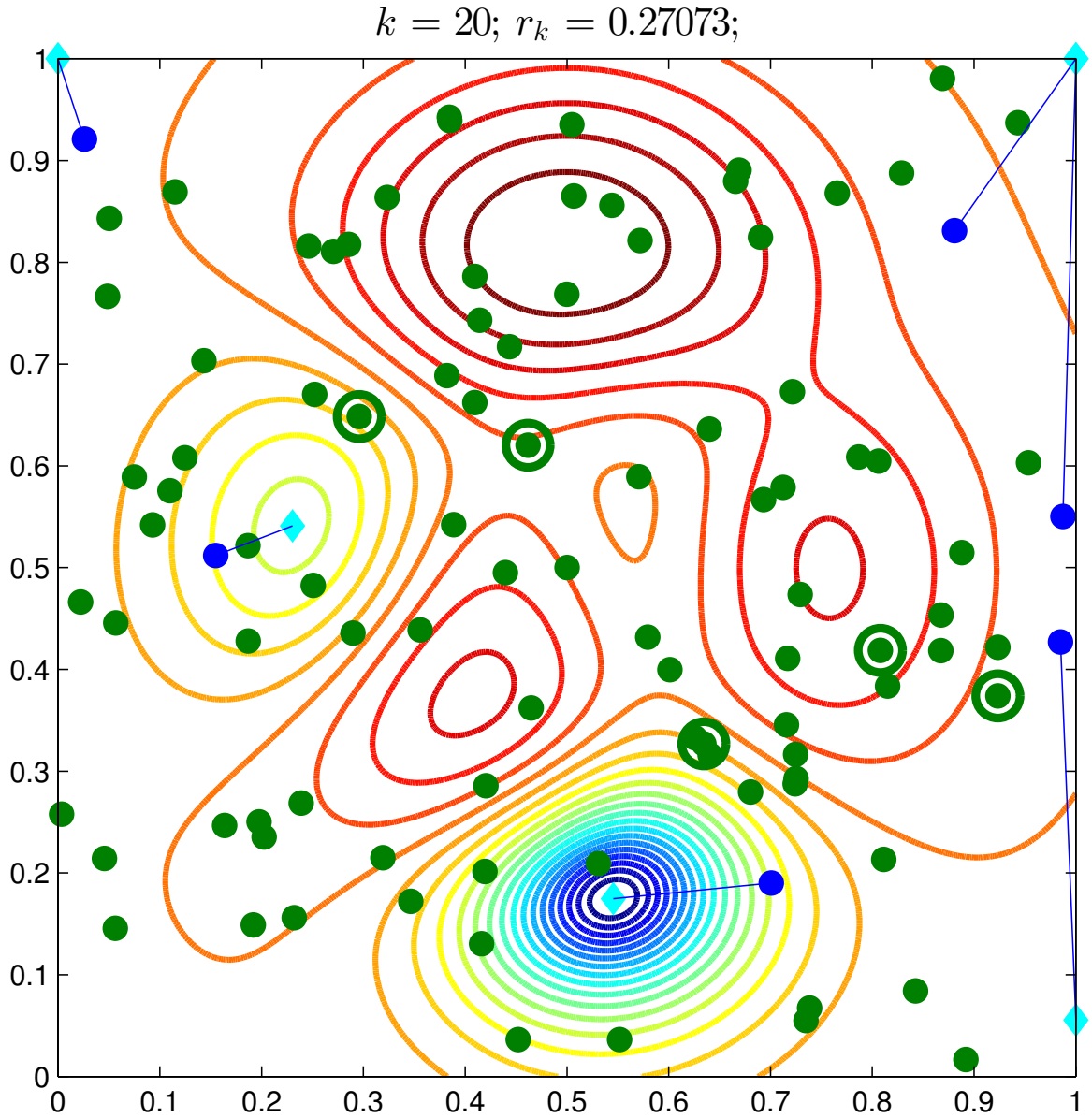
# Multi-Level Single Linkage



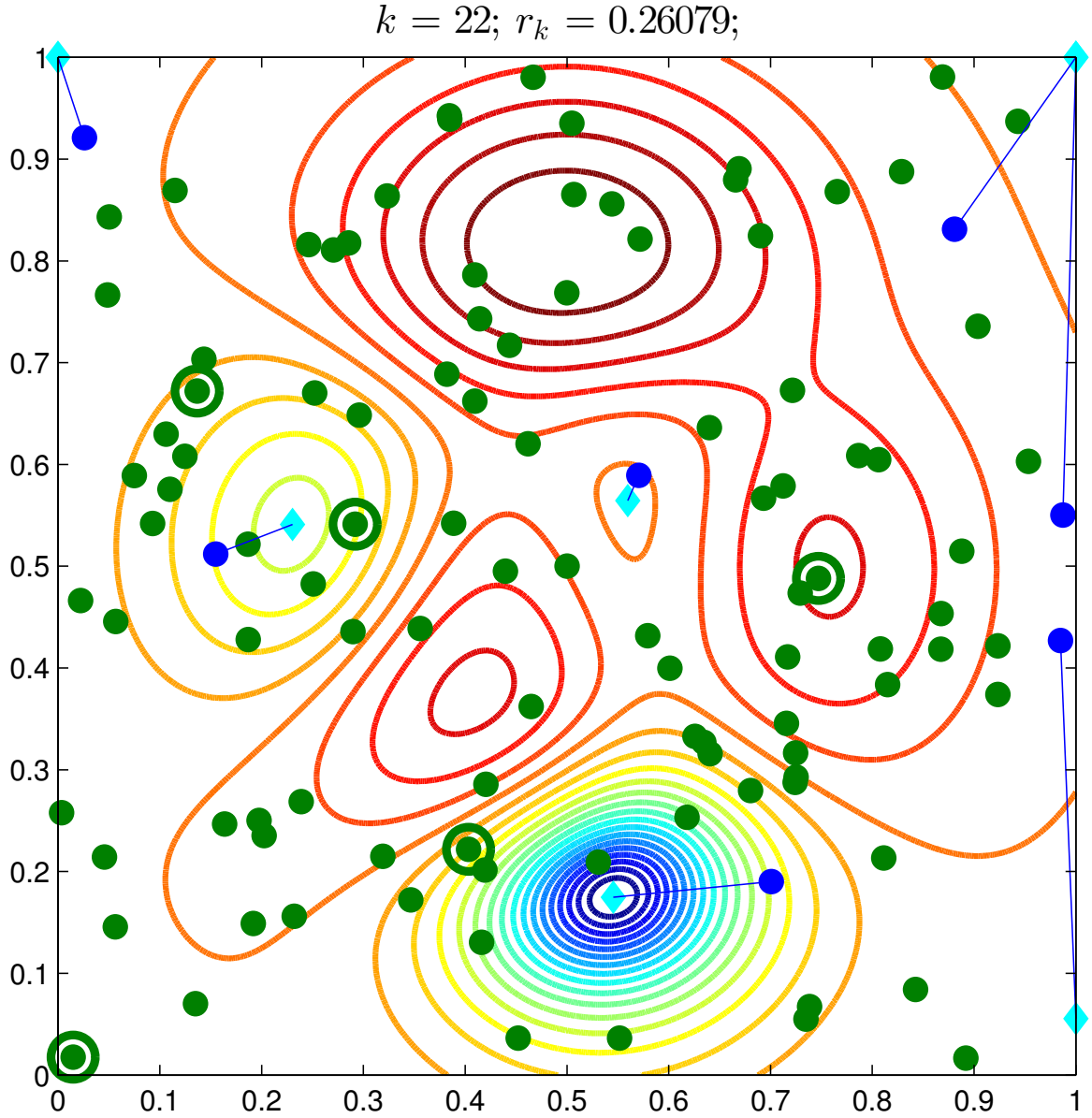
# Multi-Level Single Linkage



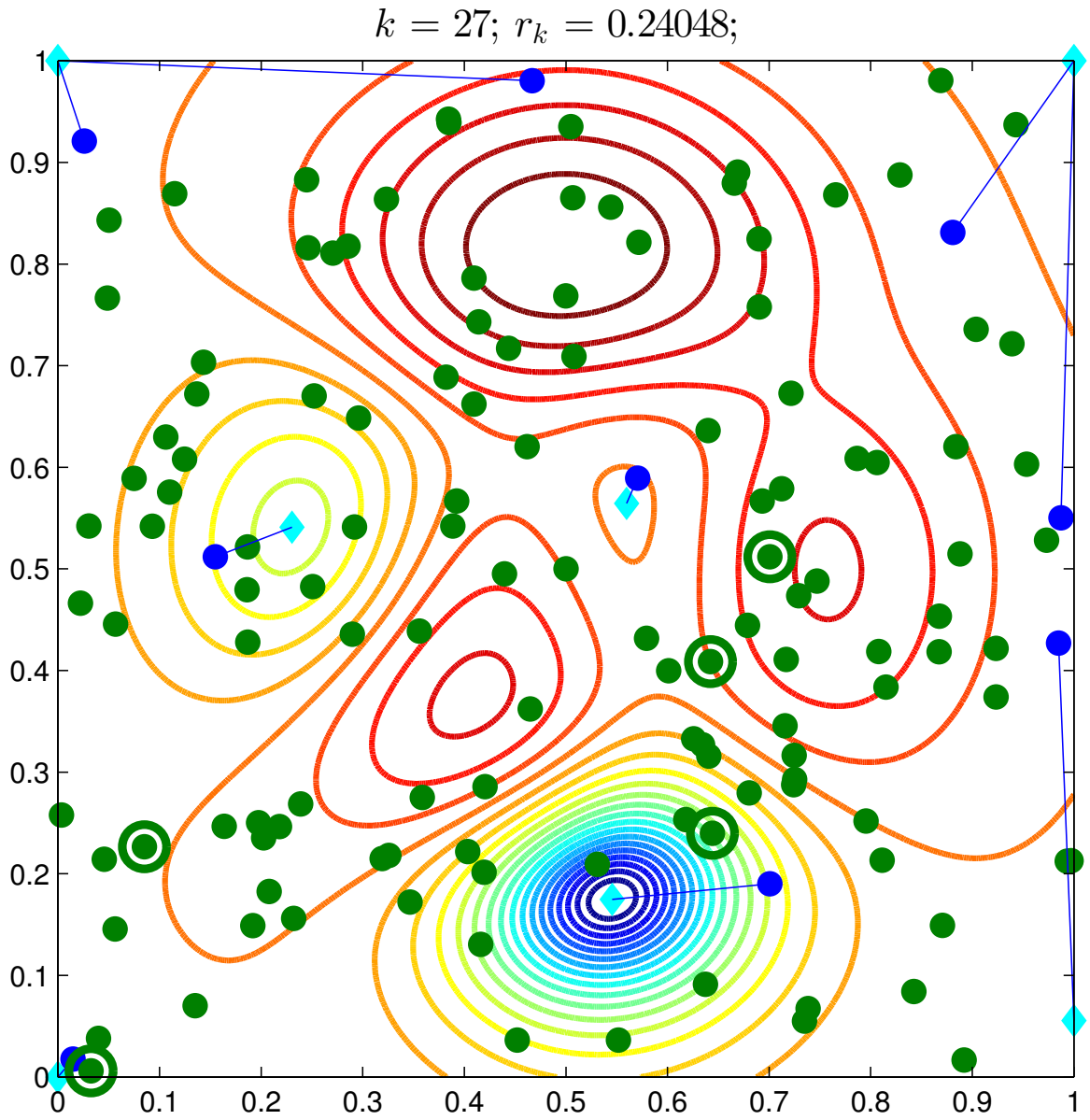
# Multi-Level Single Linkage



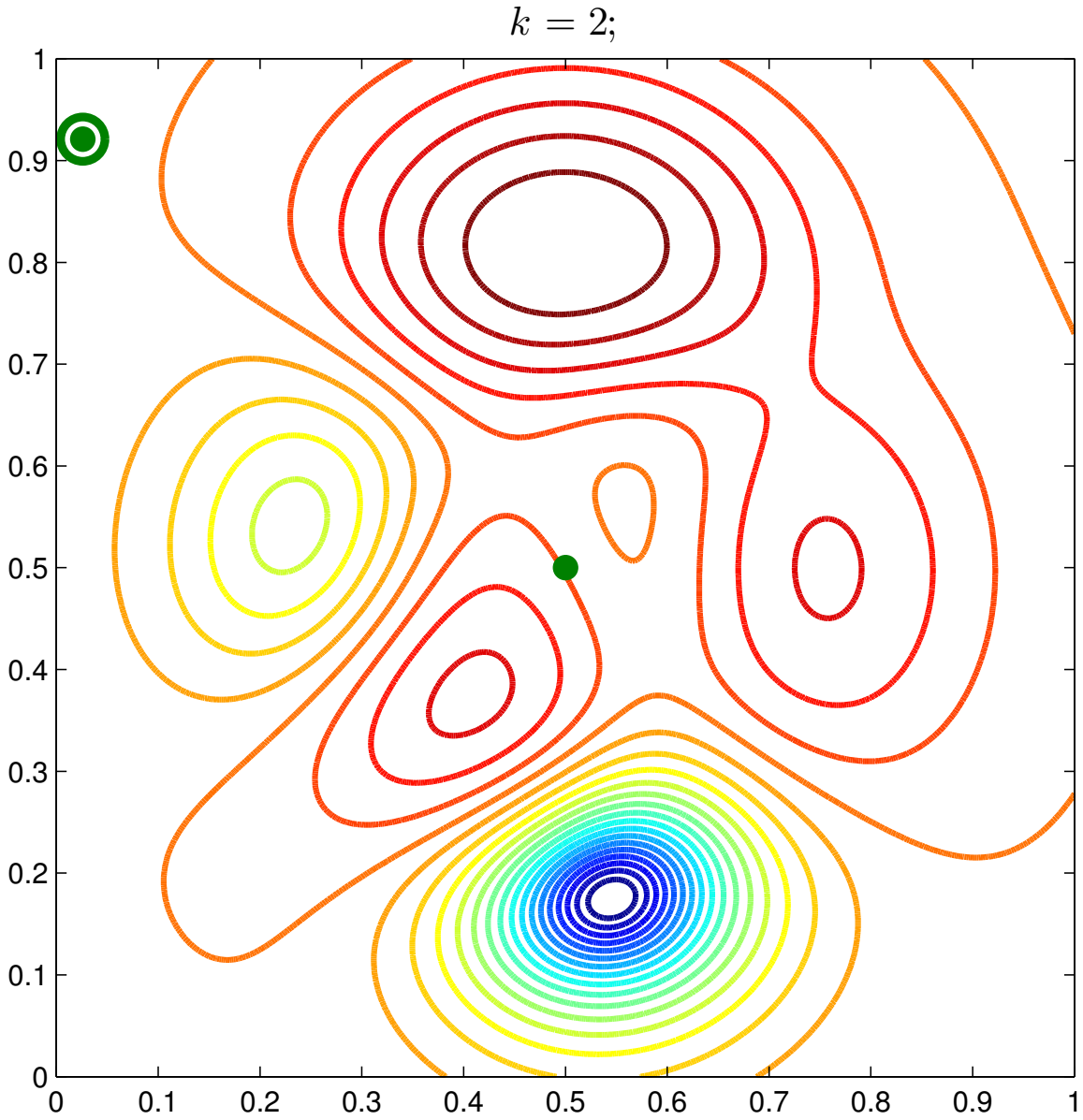
# Multi-Level Single Linkage



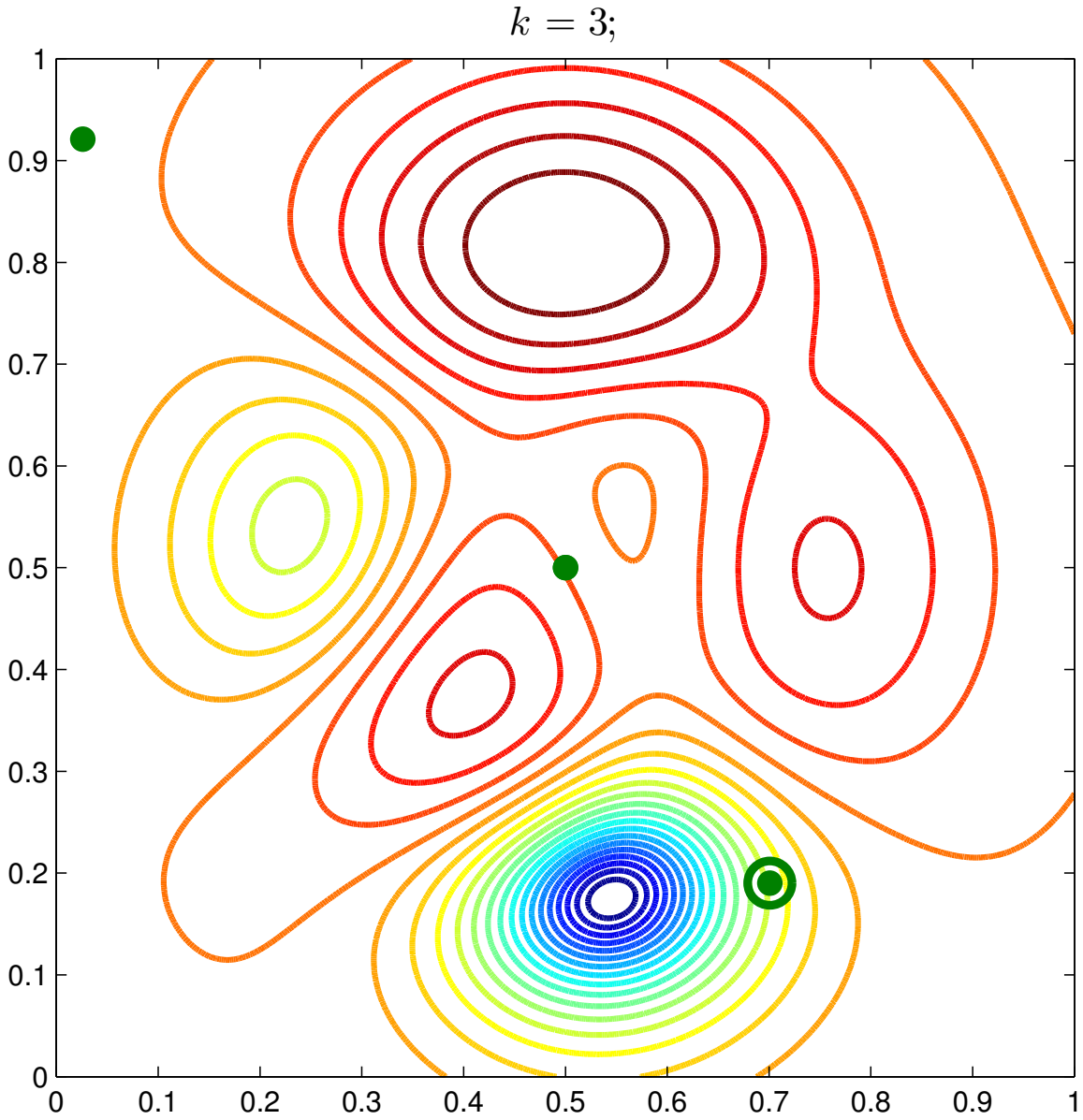
# Multi-Level Single Linkage



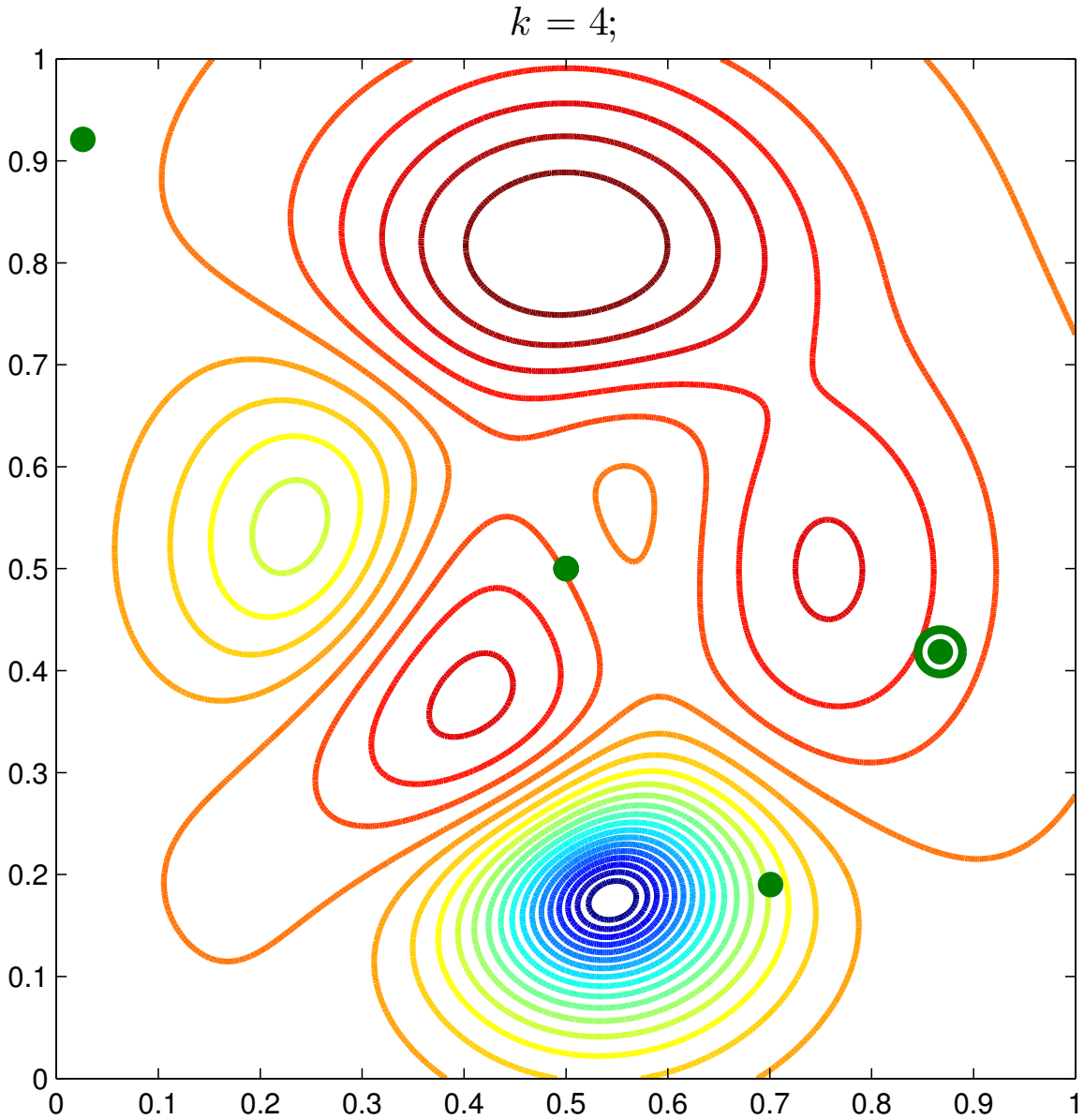
# Derivative-free MLSL



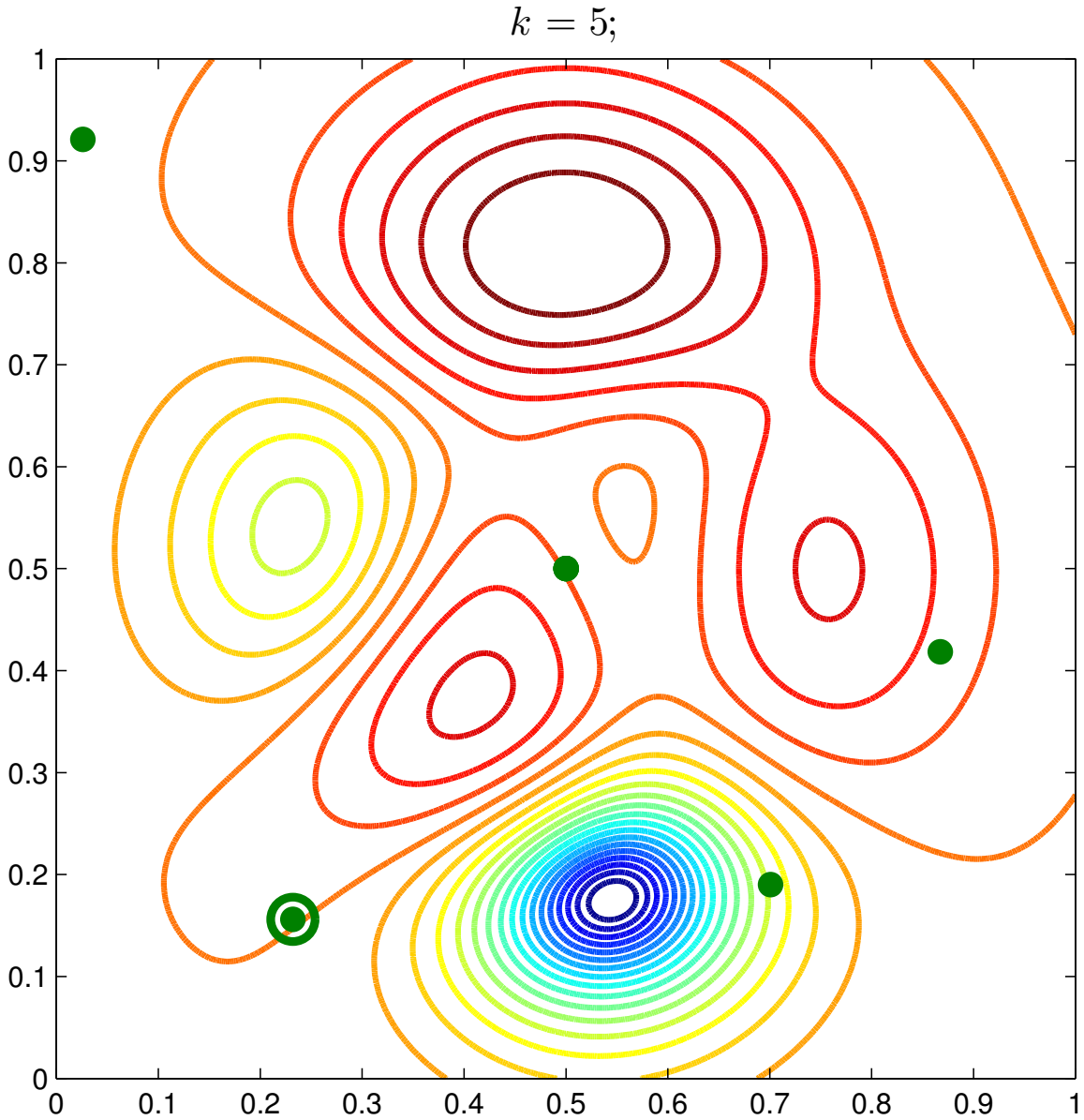
# Derivative-free MLSL



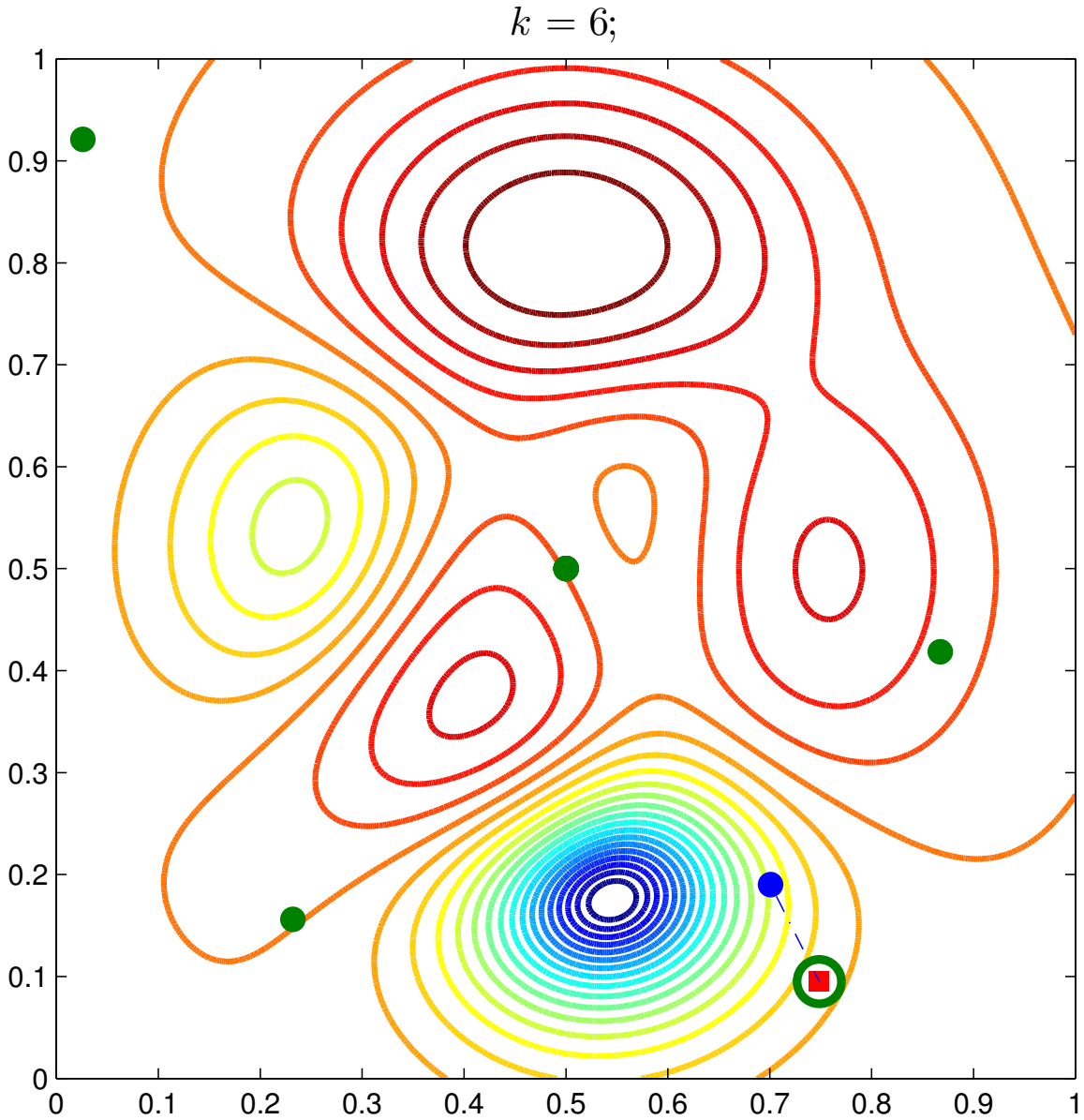
# Derivative-free MLSL



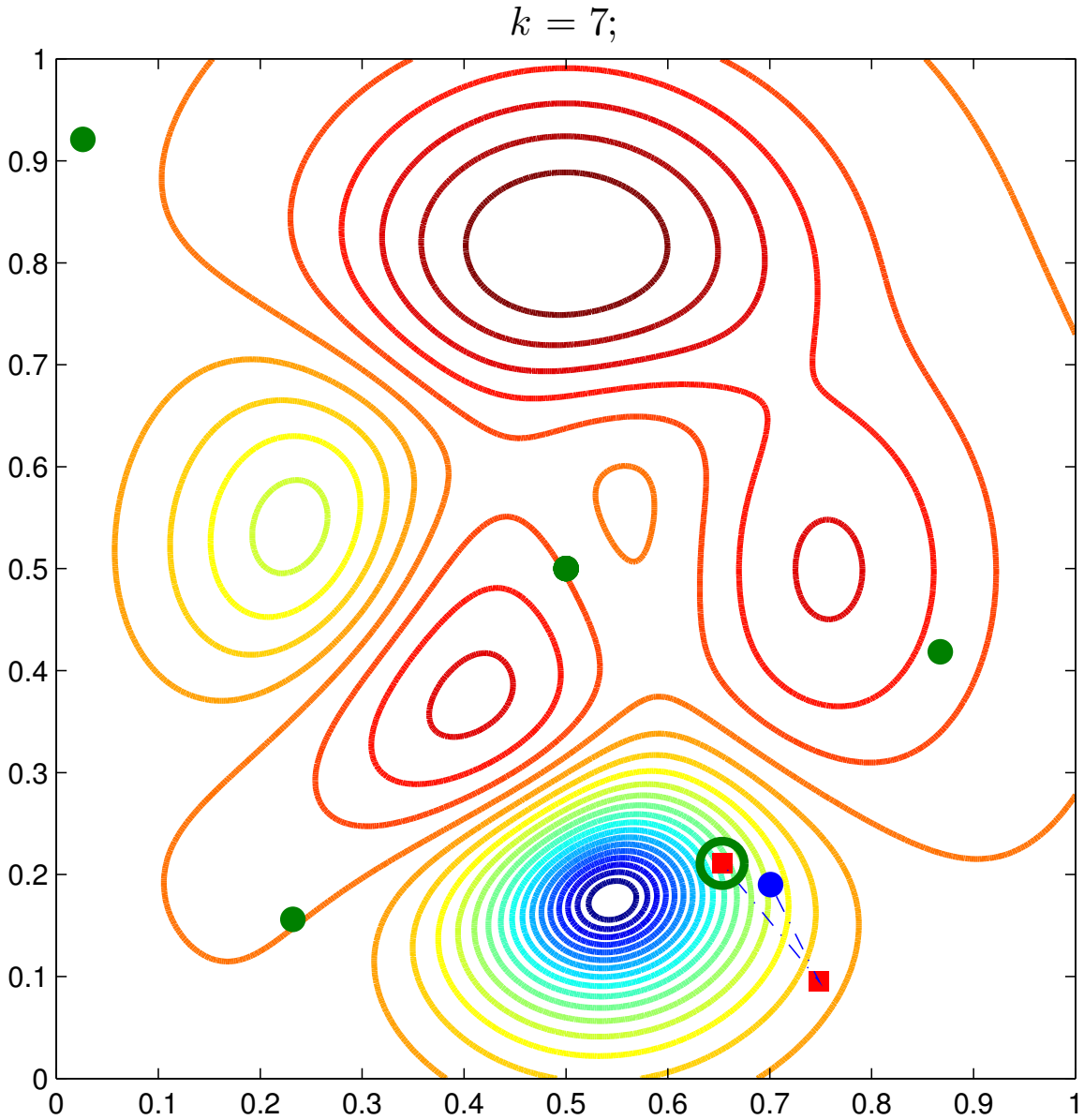
# Derivative-free MLSL



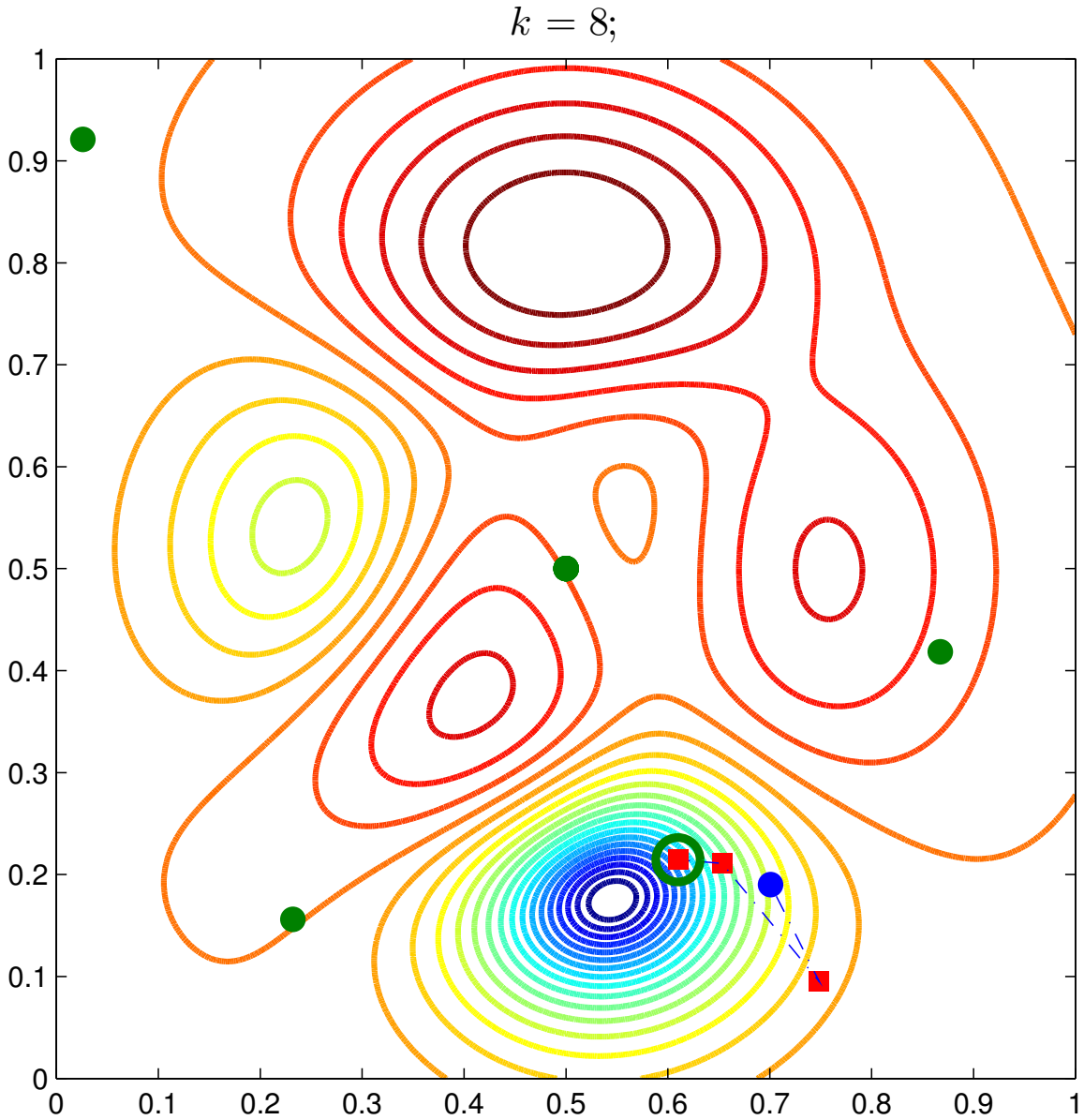
# Derivative-free MLSL



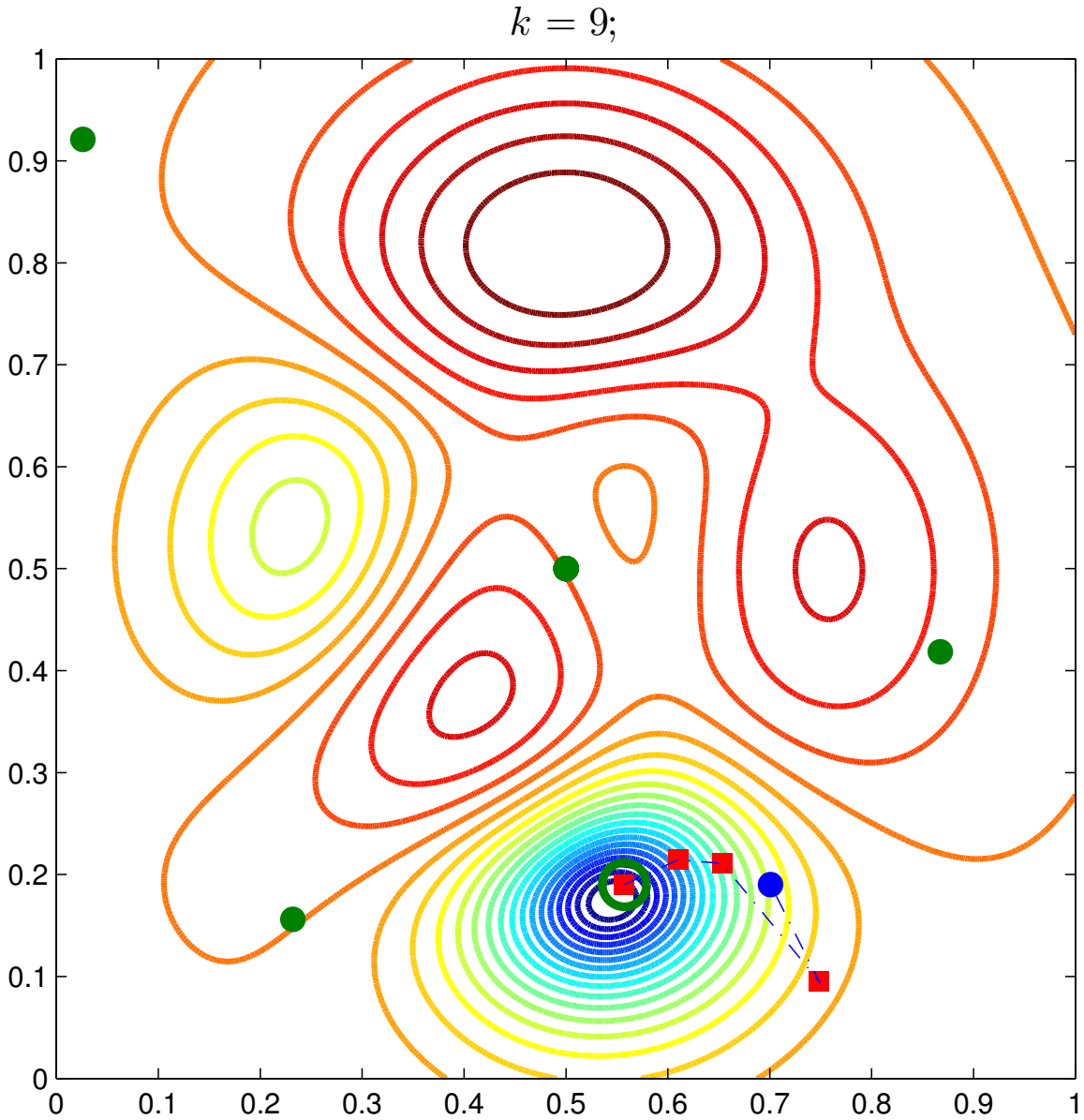
# Derivative-free MLSL



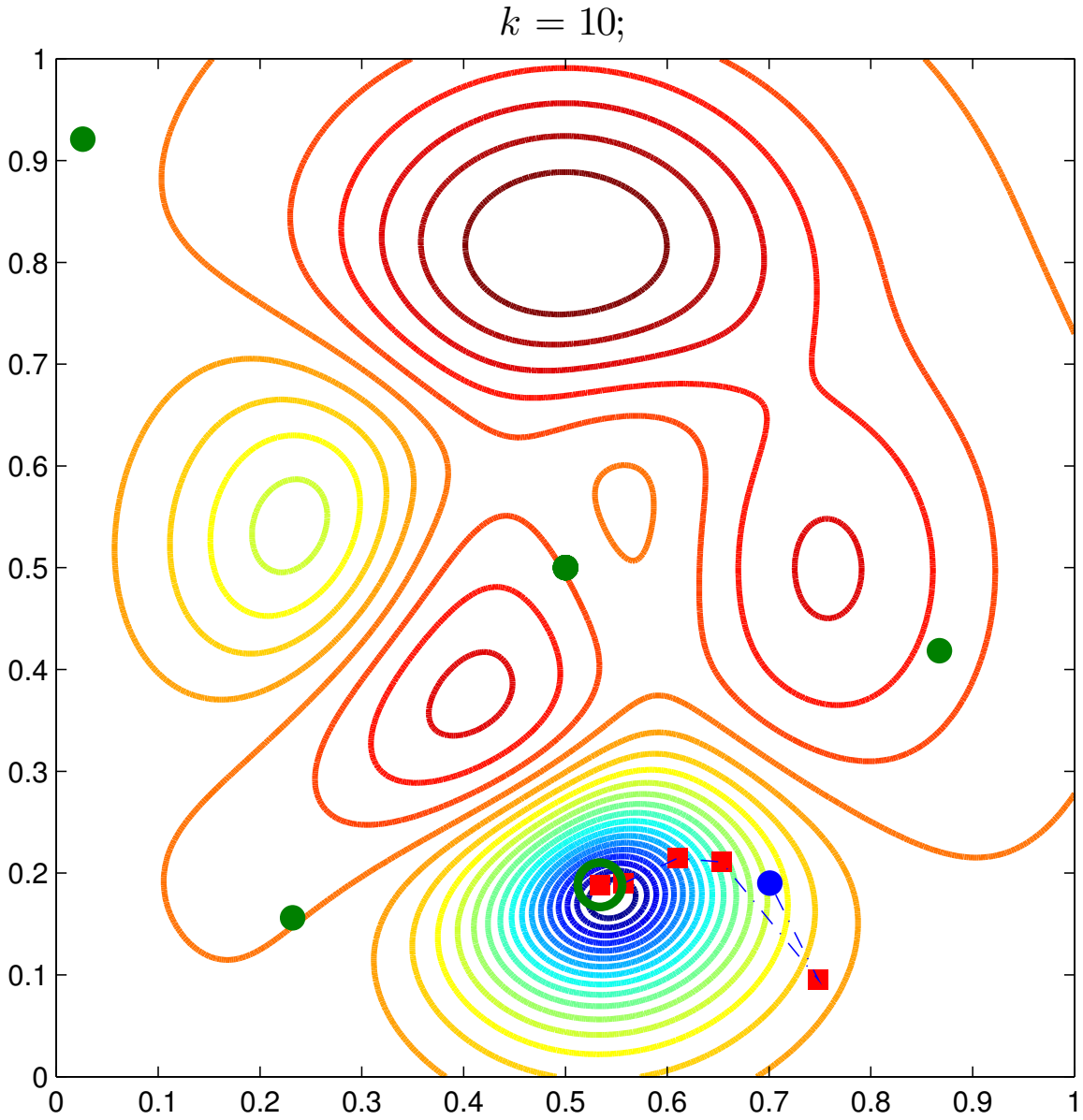
# Derivative-free MLSL



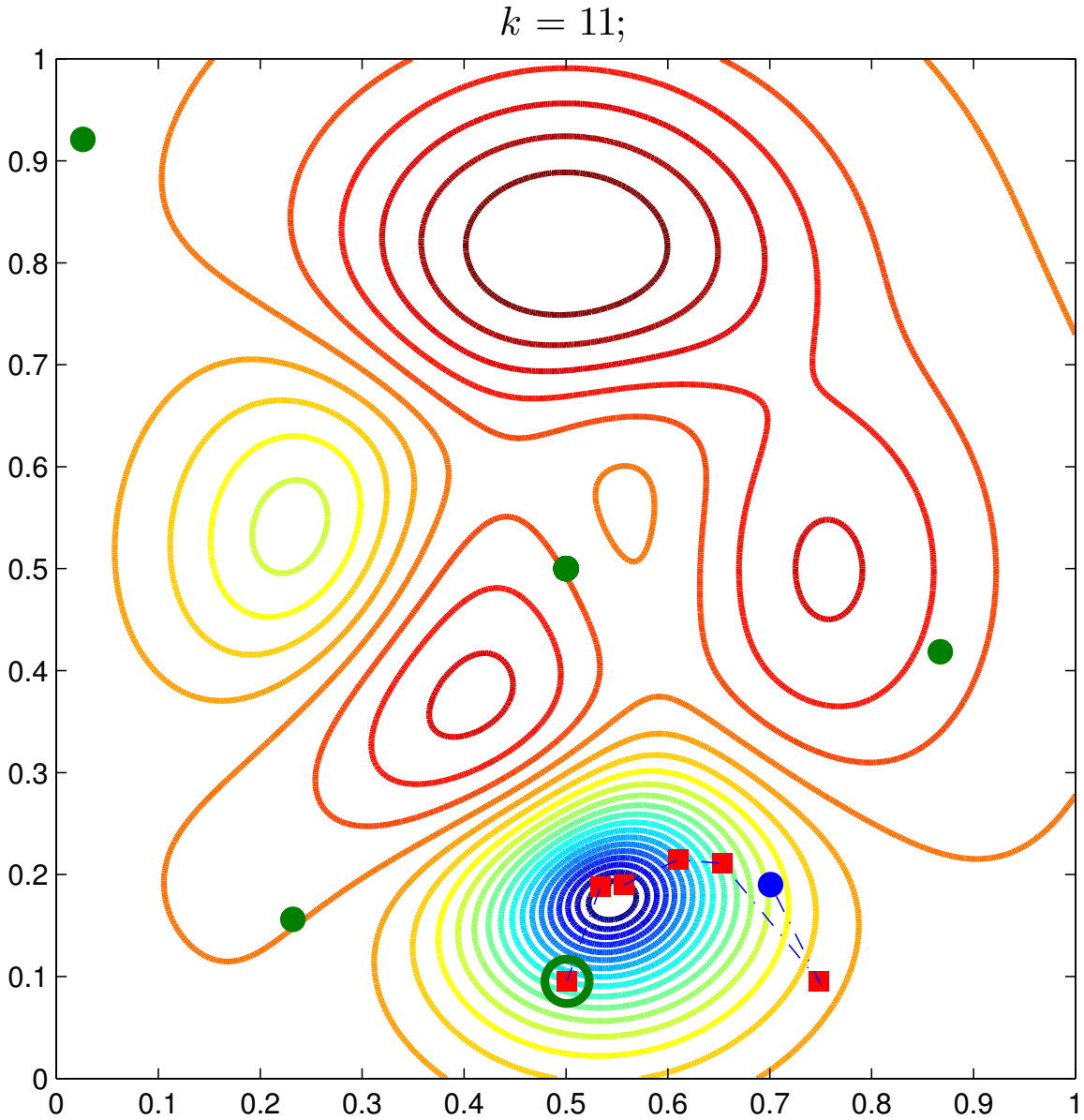
# Derivative-free MLSL



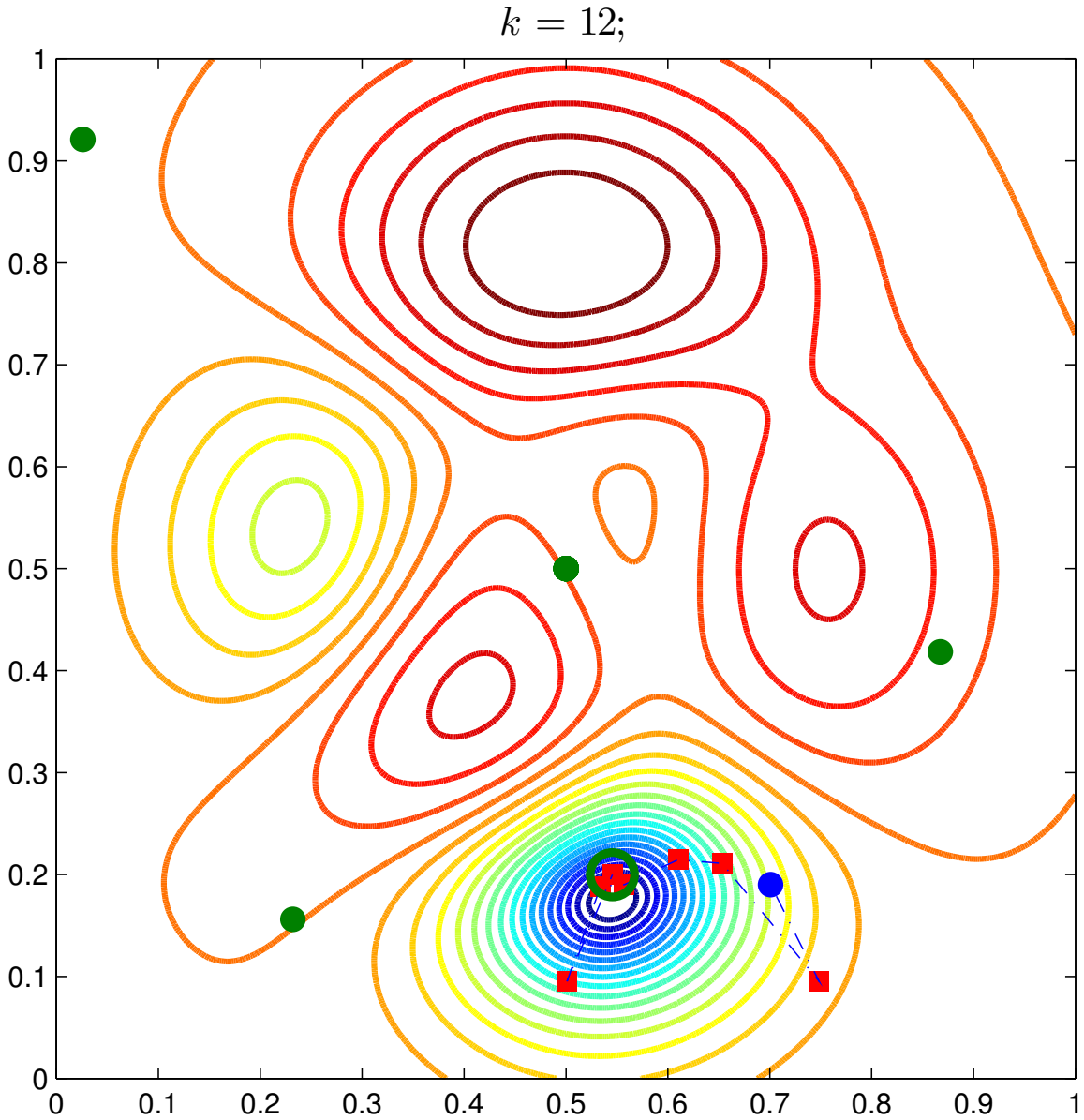
# Derivative-free MLSL



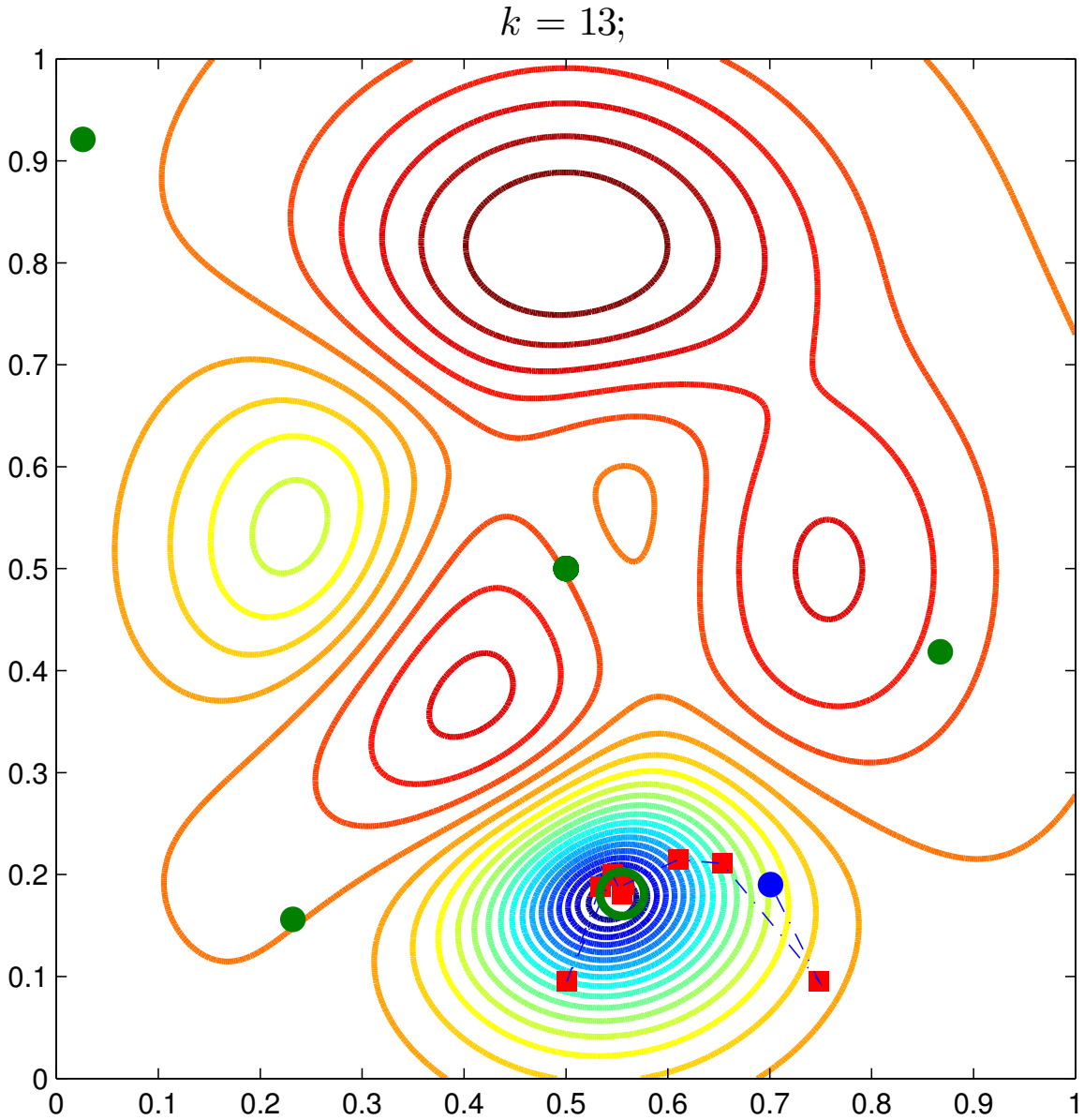
# Derivative-free MLSL



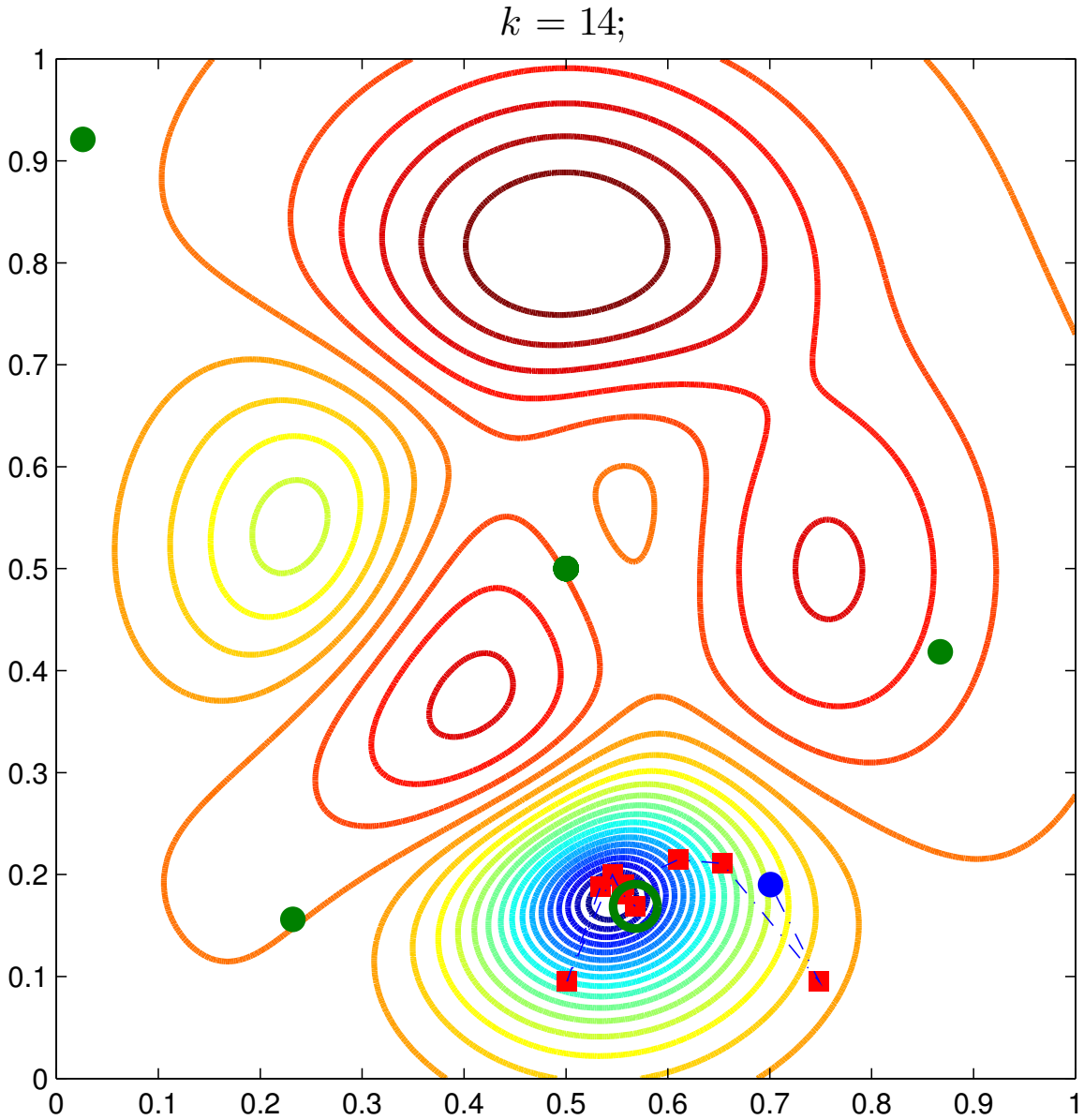
# Derivative-free MLSL



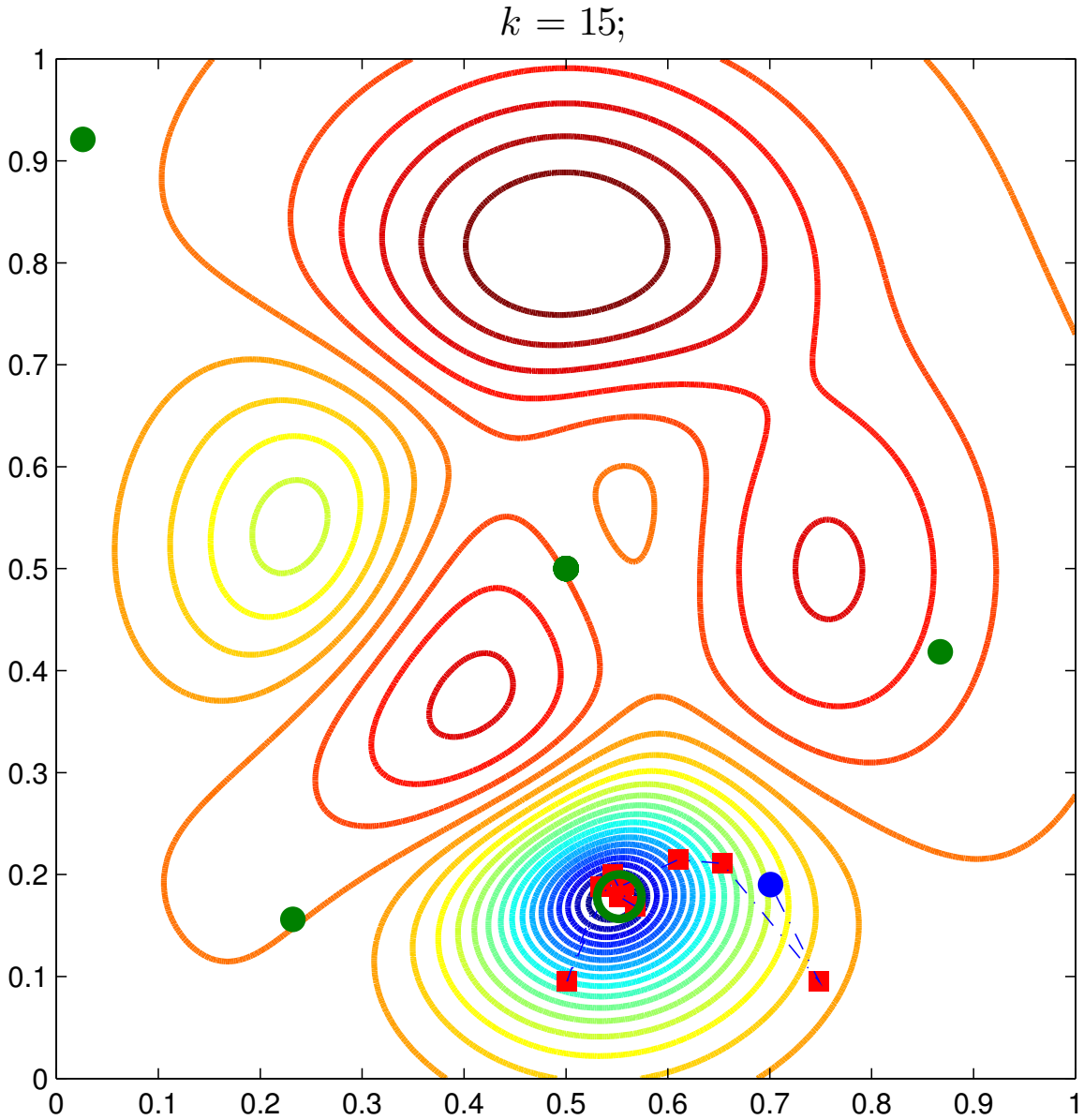
# Derivative-free MLSL



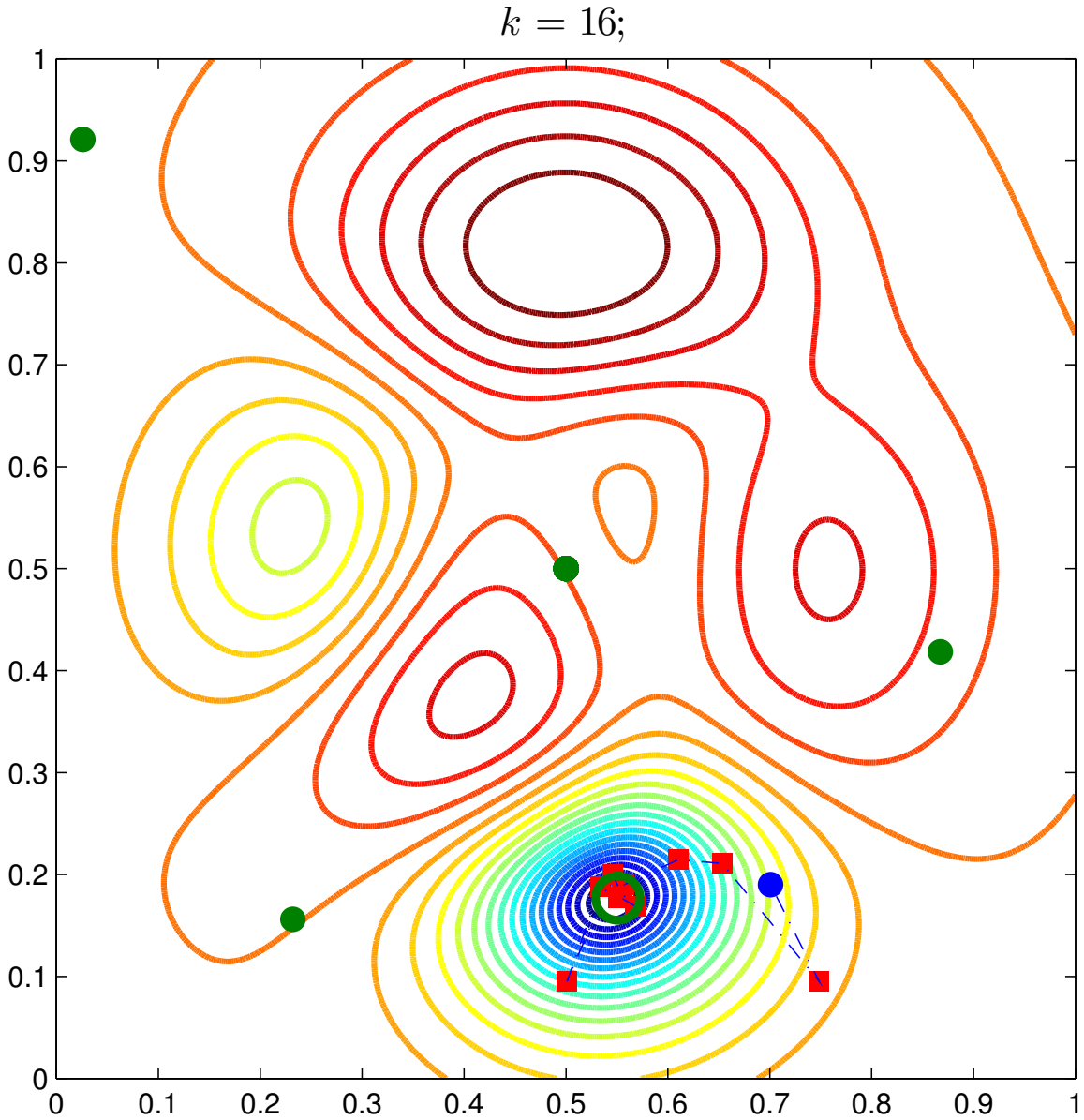
# Derivative-free MLSL



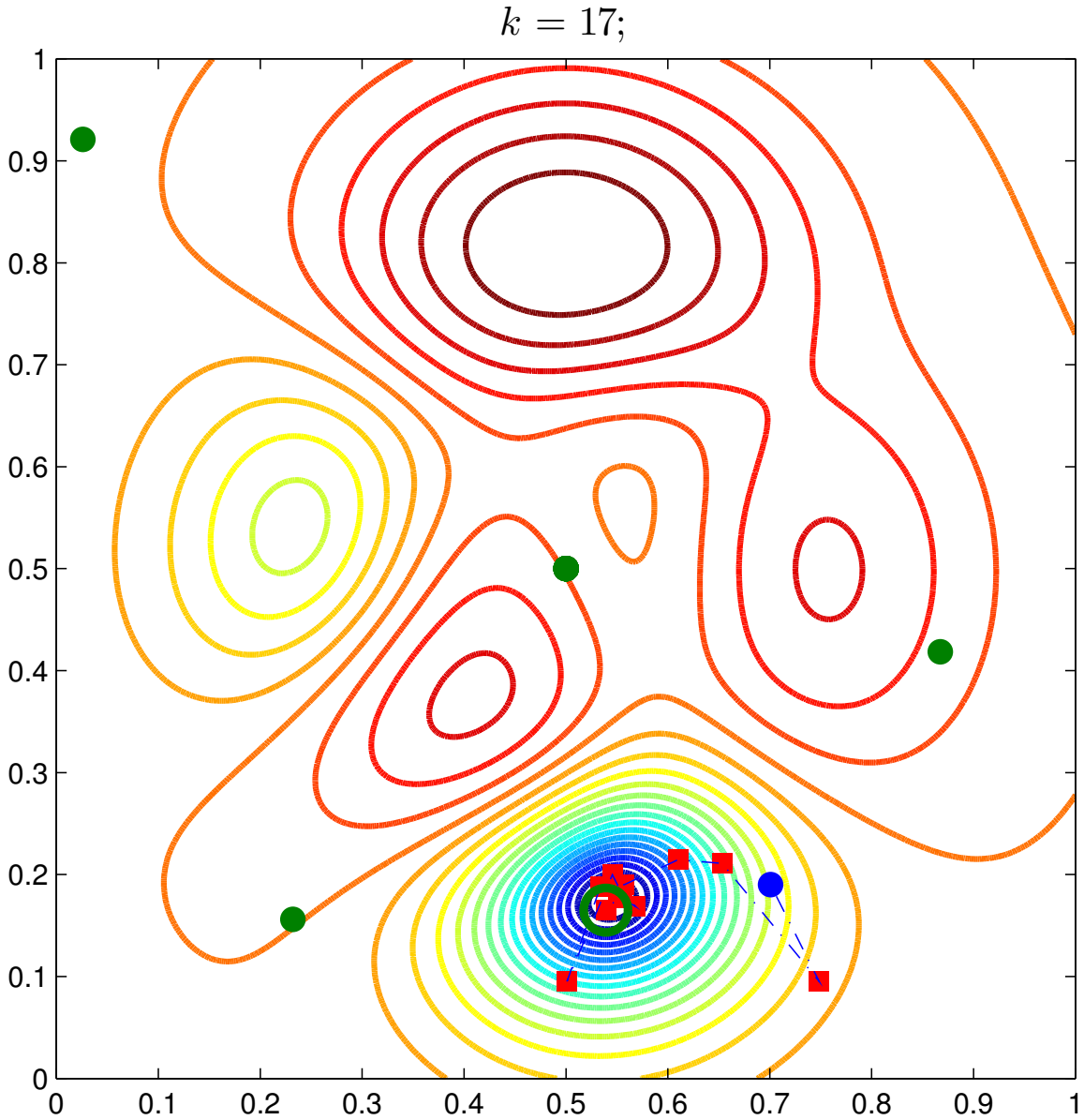
# Derivative-free MLSL



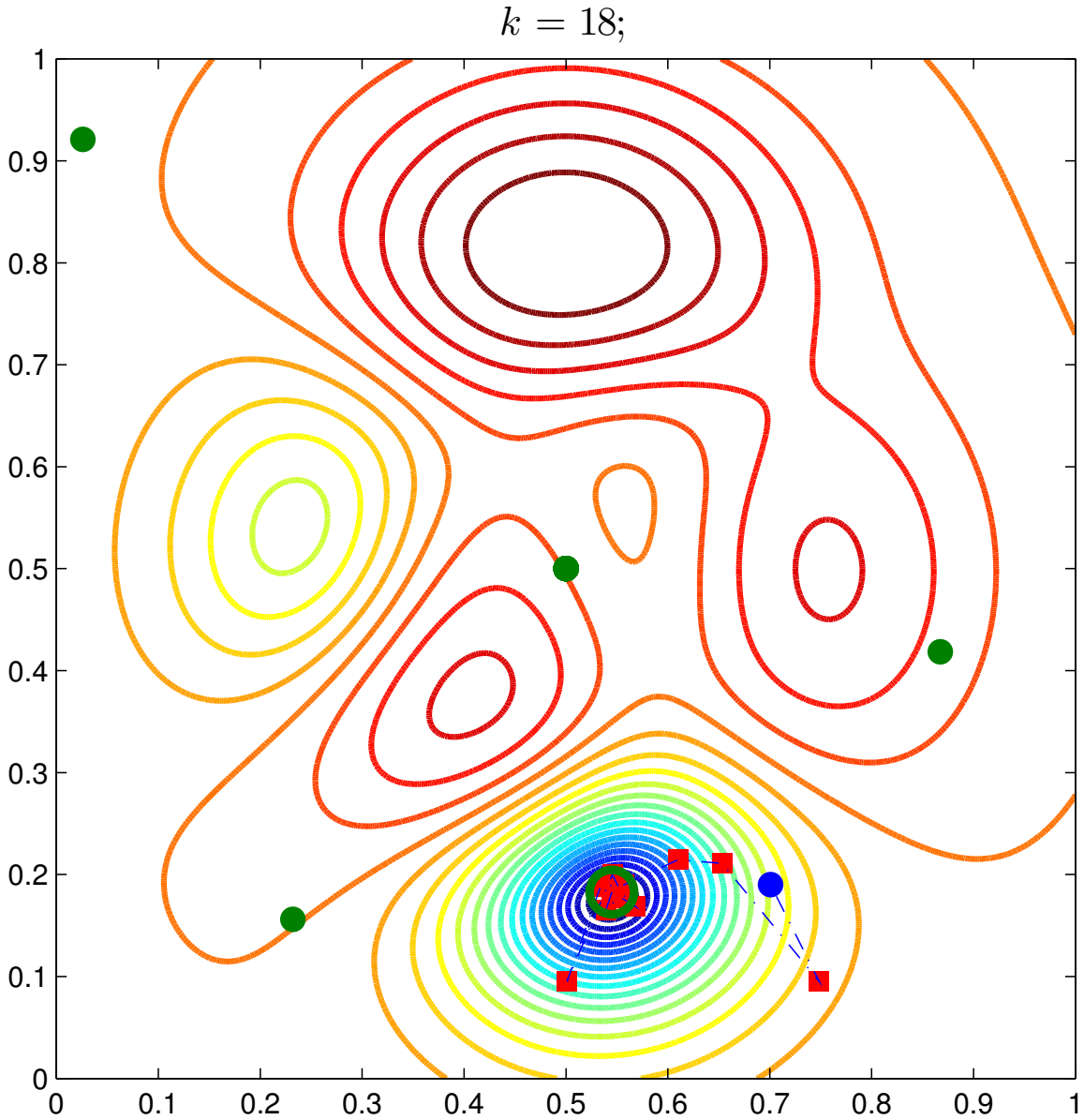
# Derivative-free MLSL



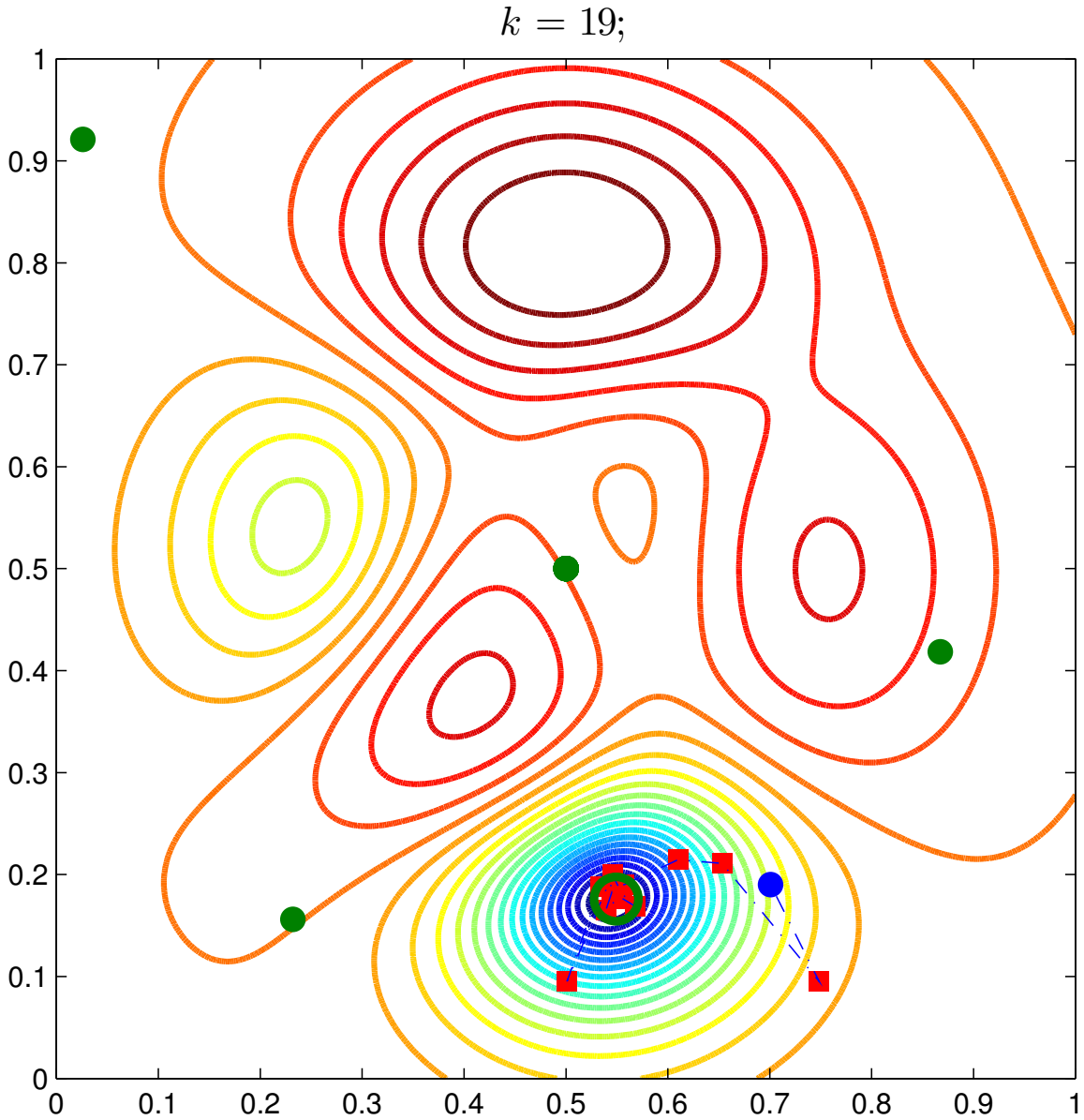
# Derivative-free MLSL



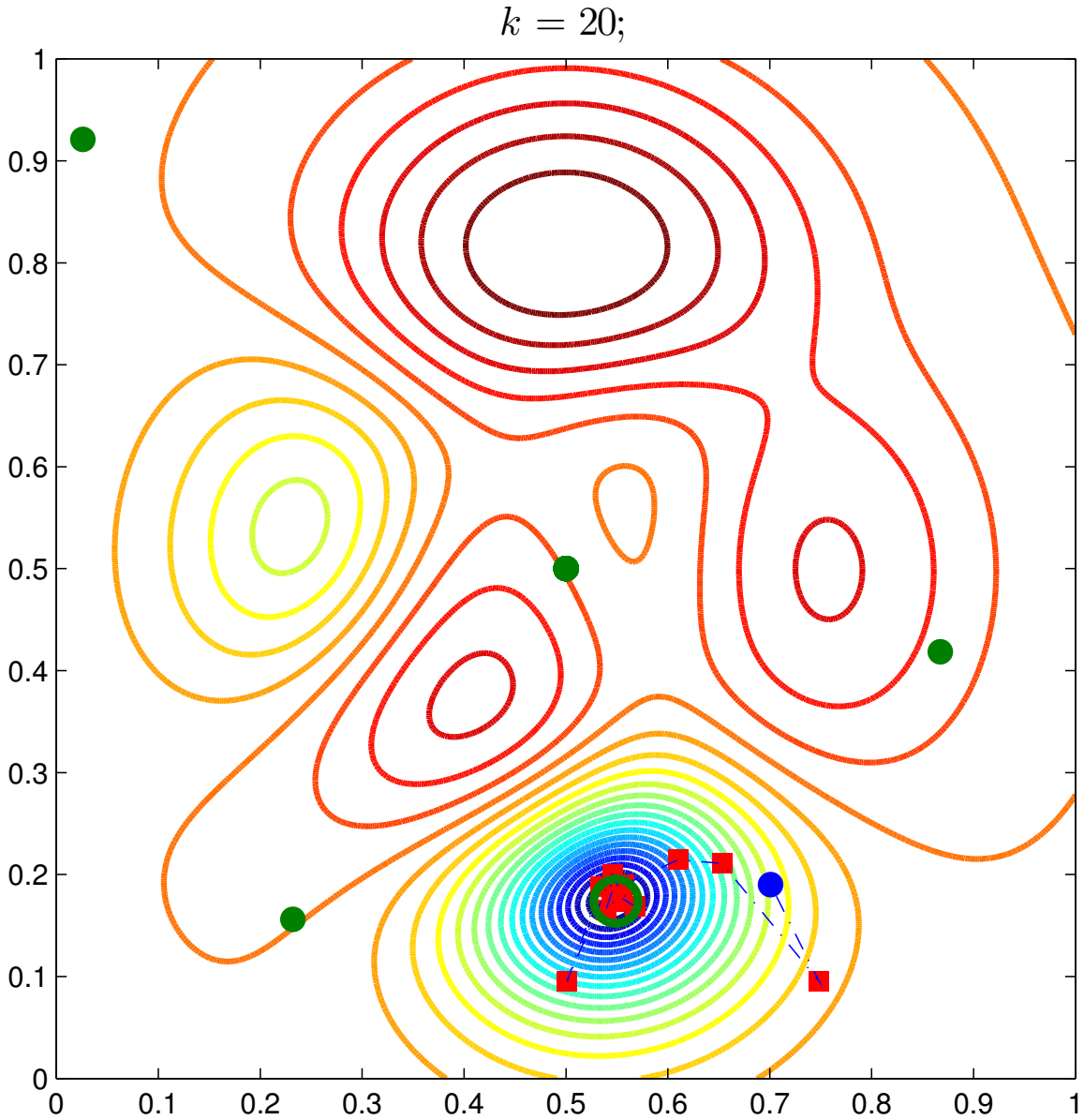
# Derivative-free MLSL



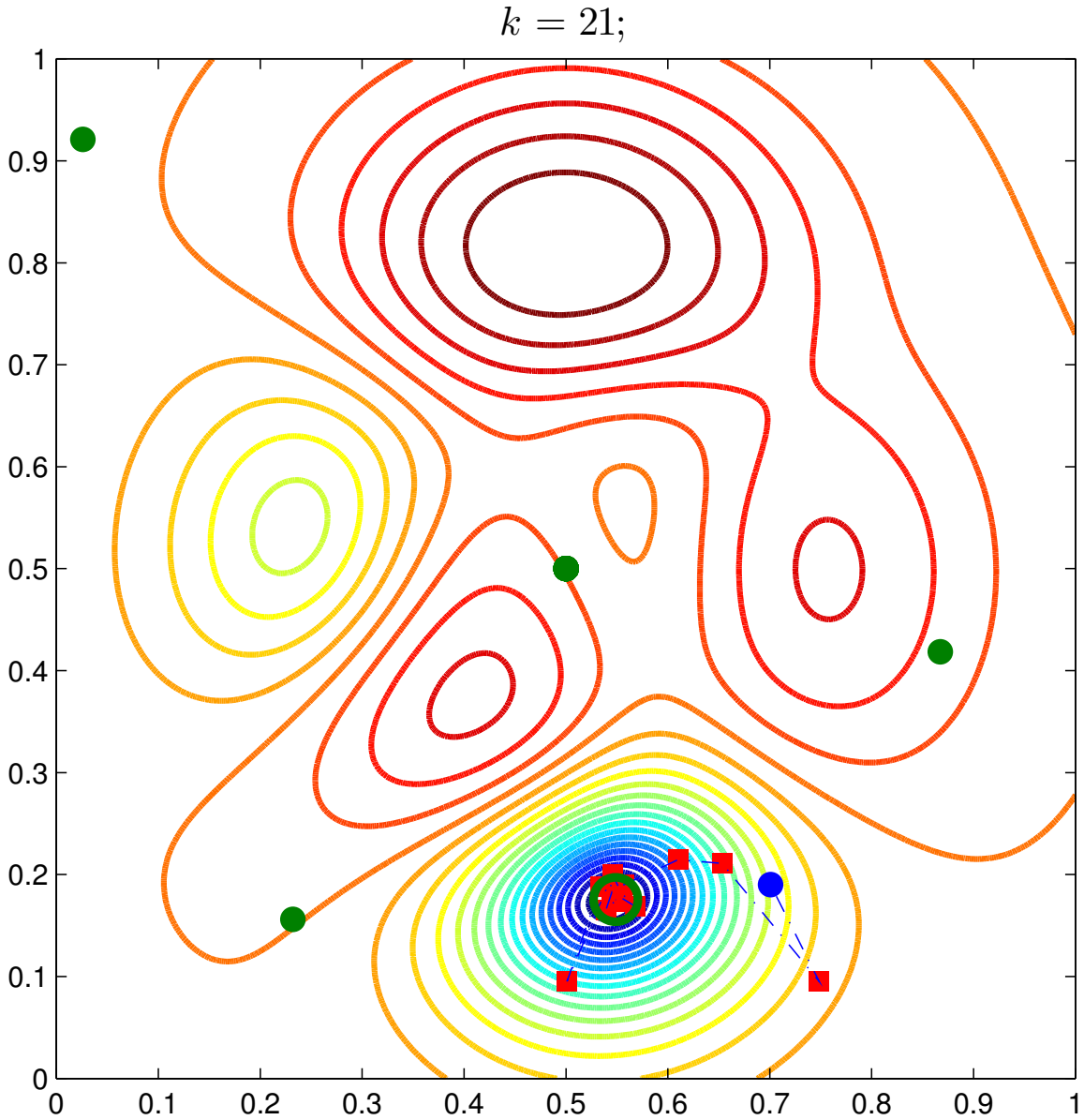
# Derivative-free MLSL



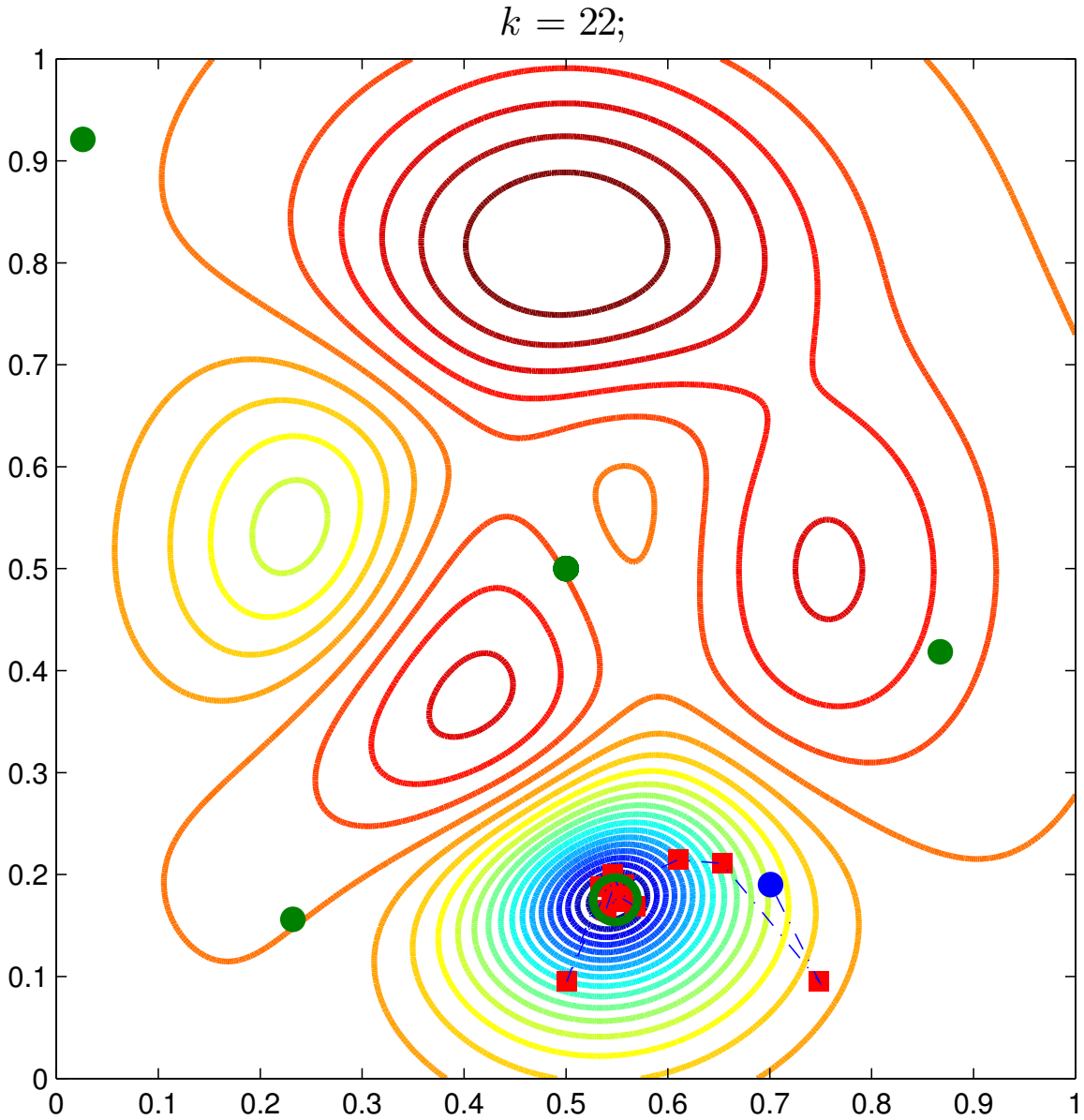
# Derivative-free MLSL



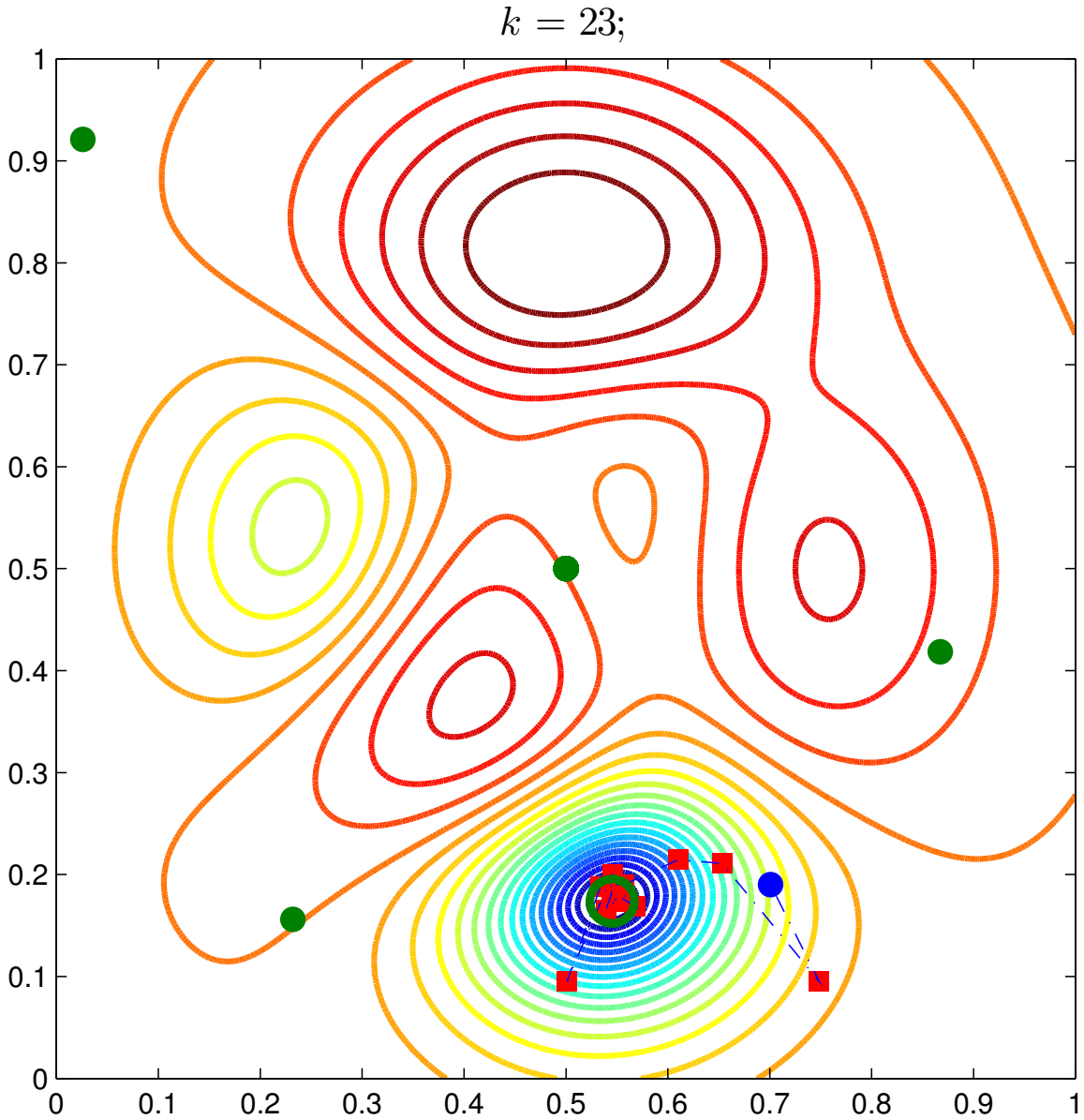
# Derivative-free MLSL



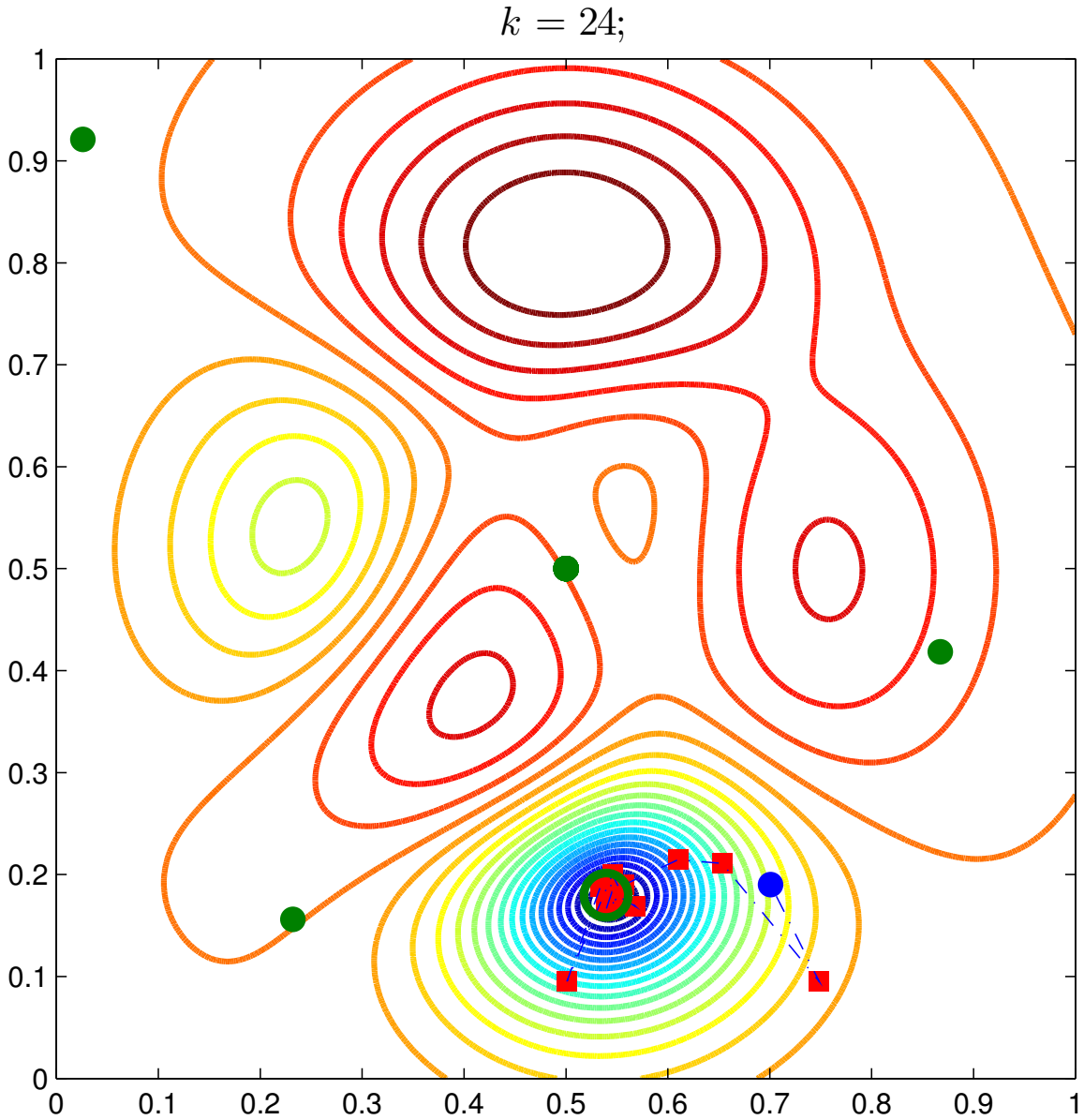
# Derivative-free MLSL



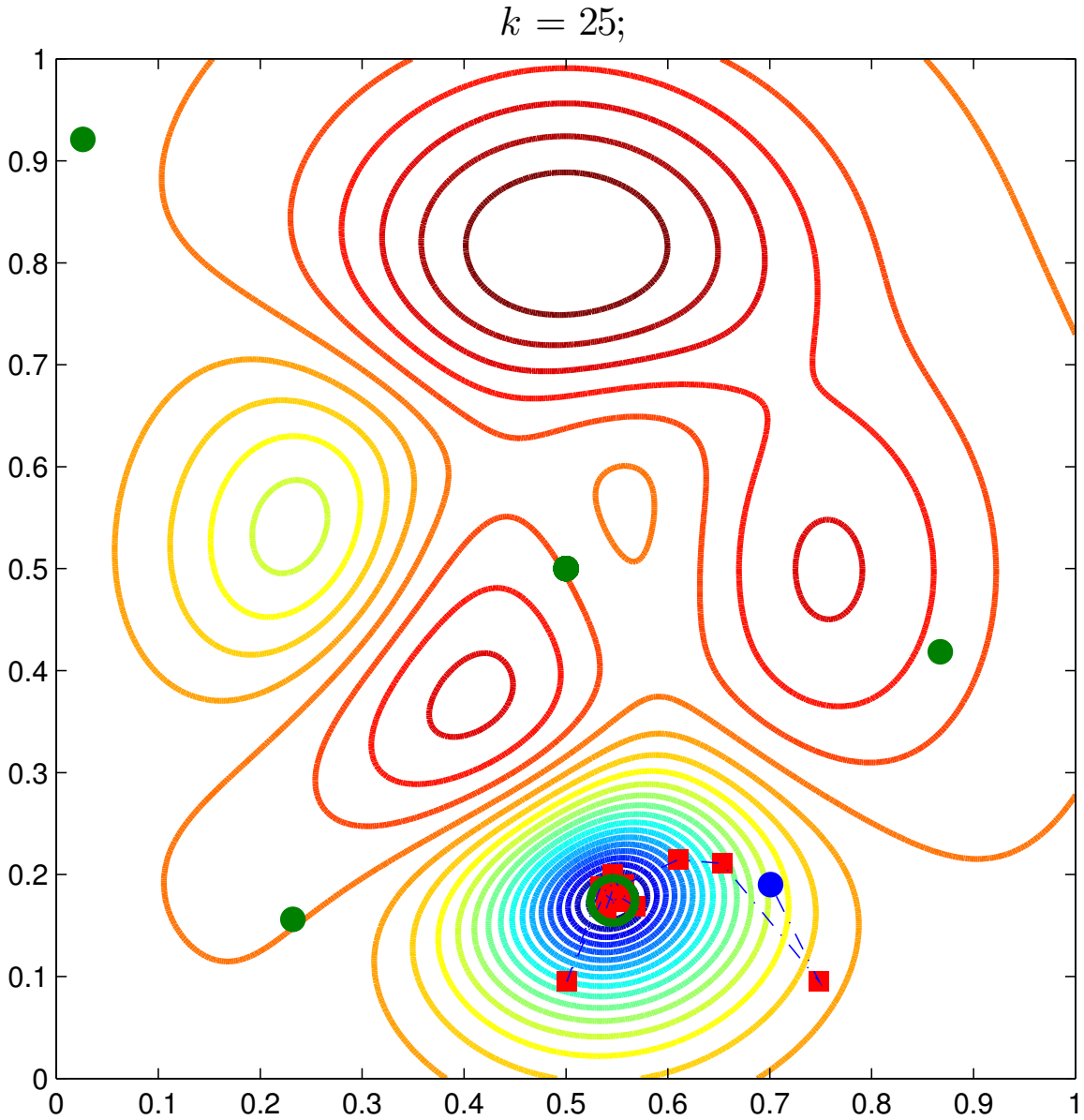
# Derivative-free MLSL



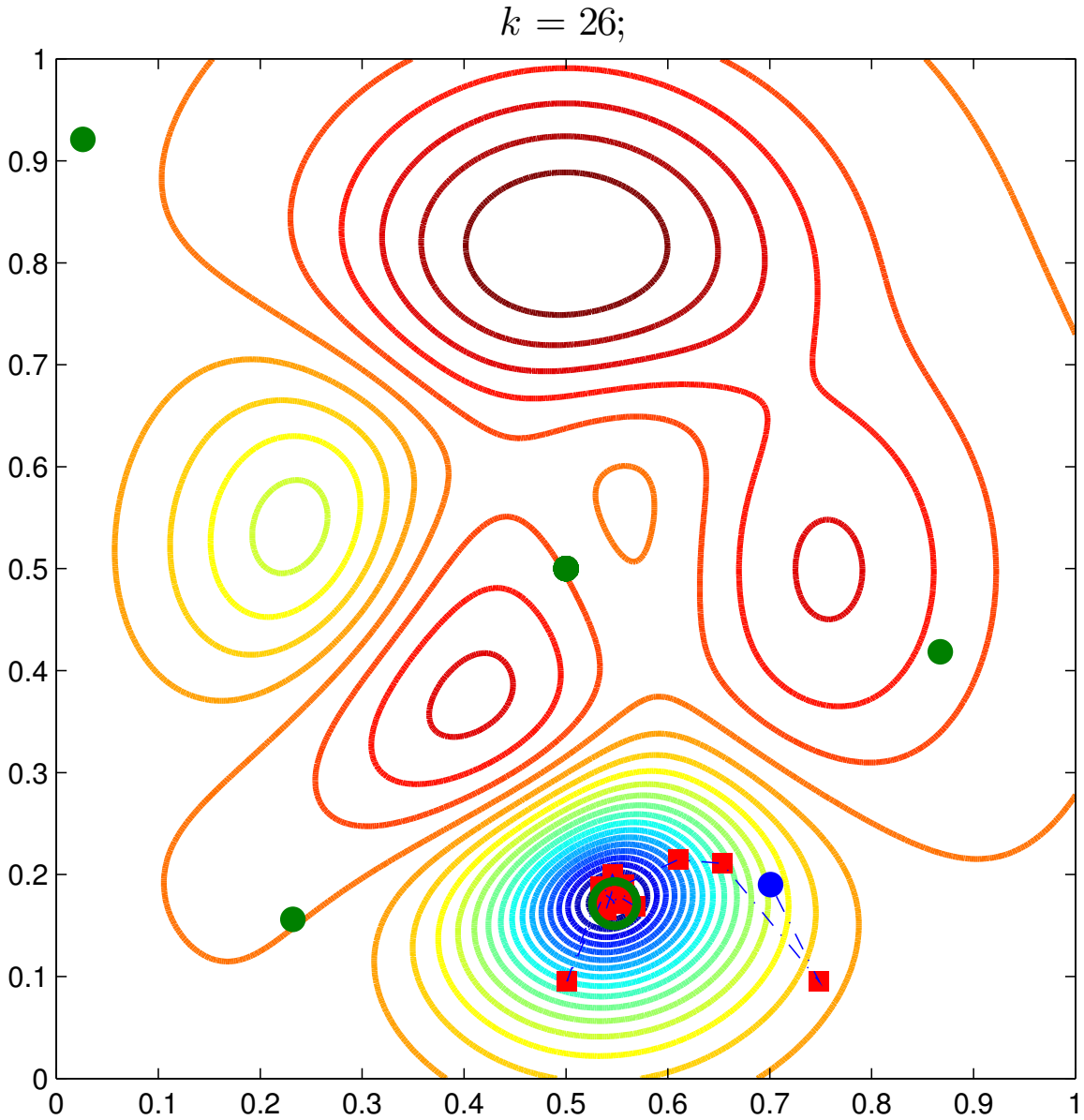
# Derivative-free MLSL



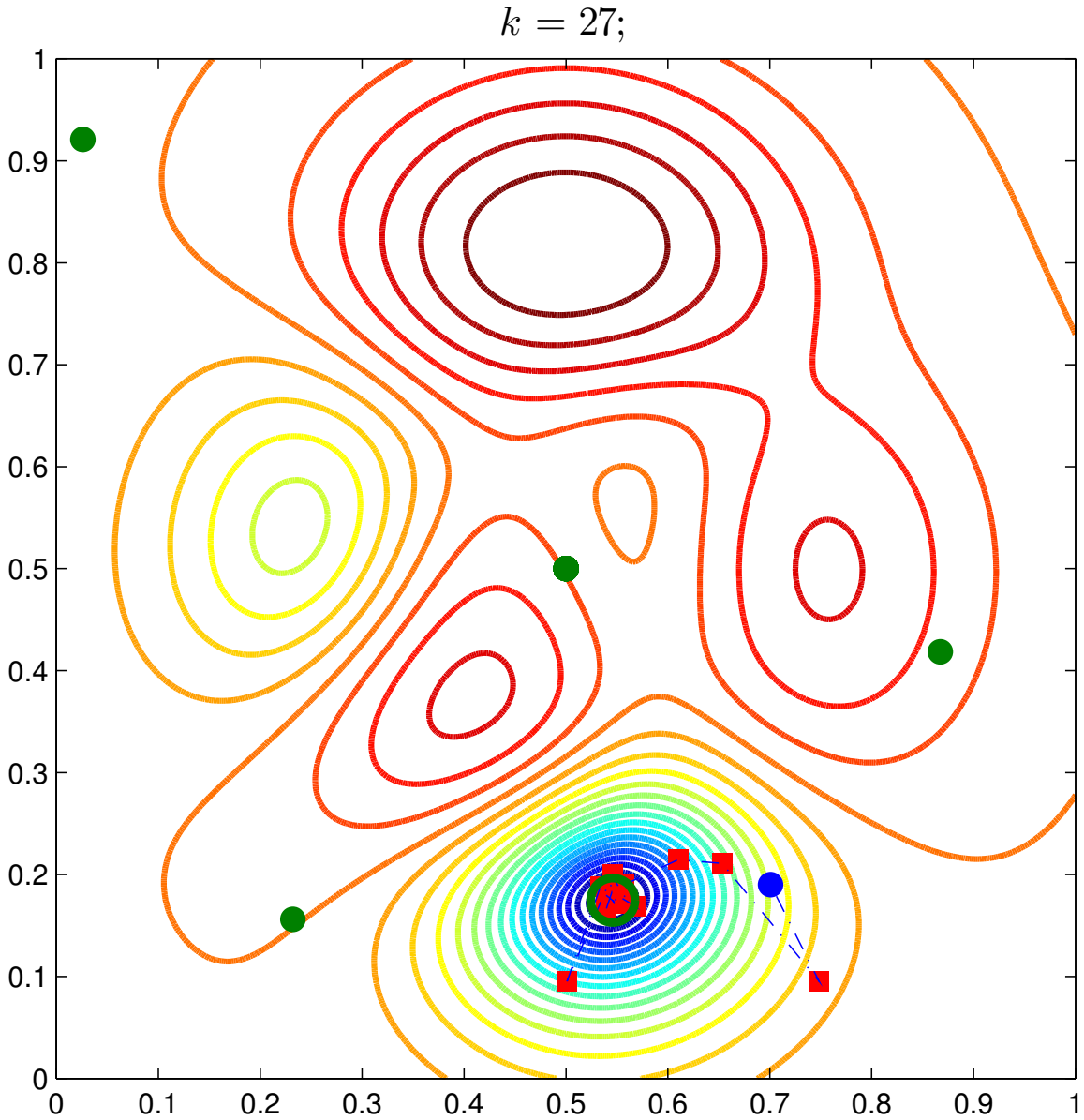
# Derivative-free MLSL



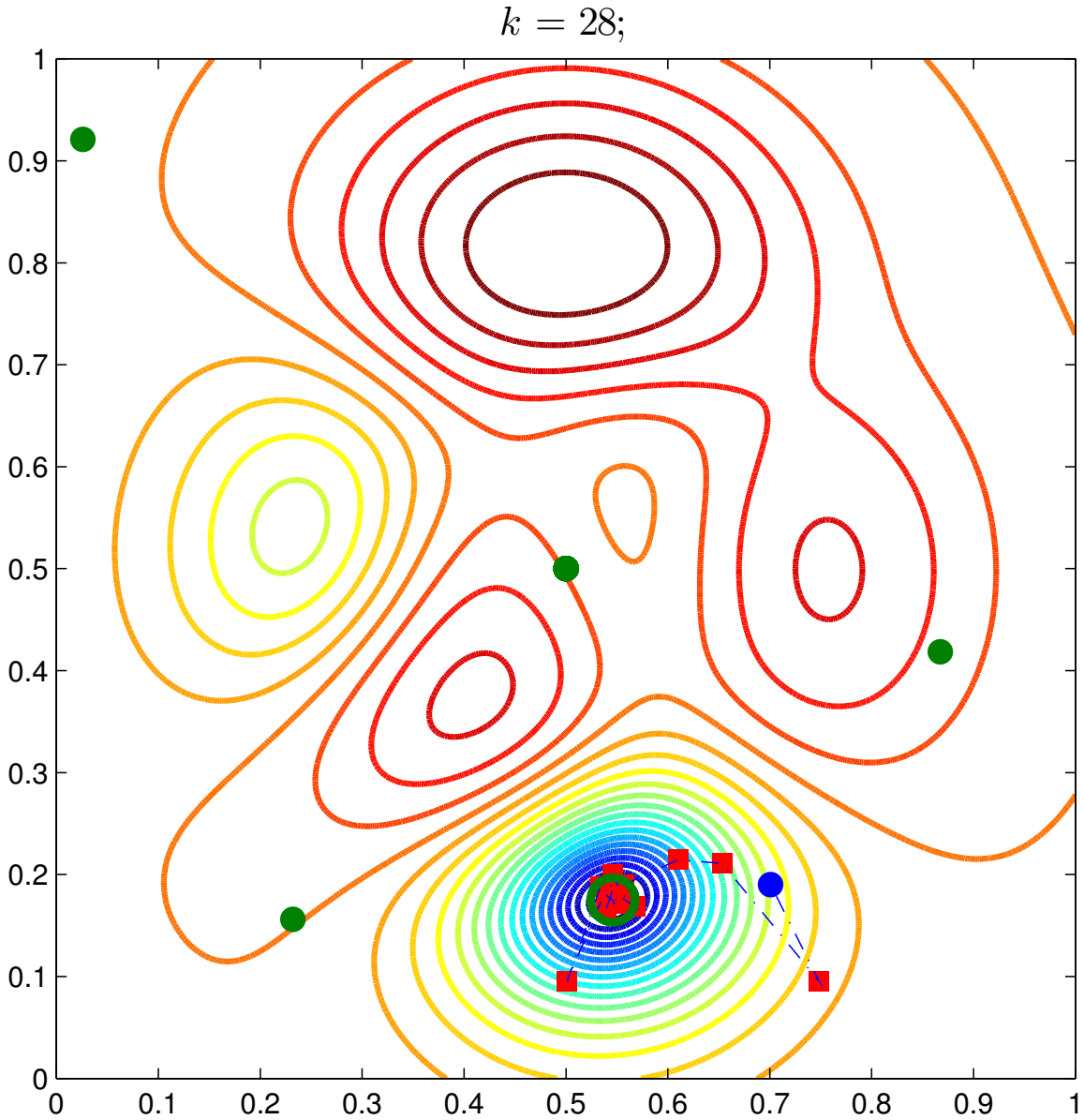
# Derivative-free MLSL



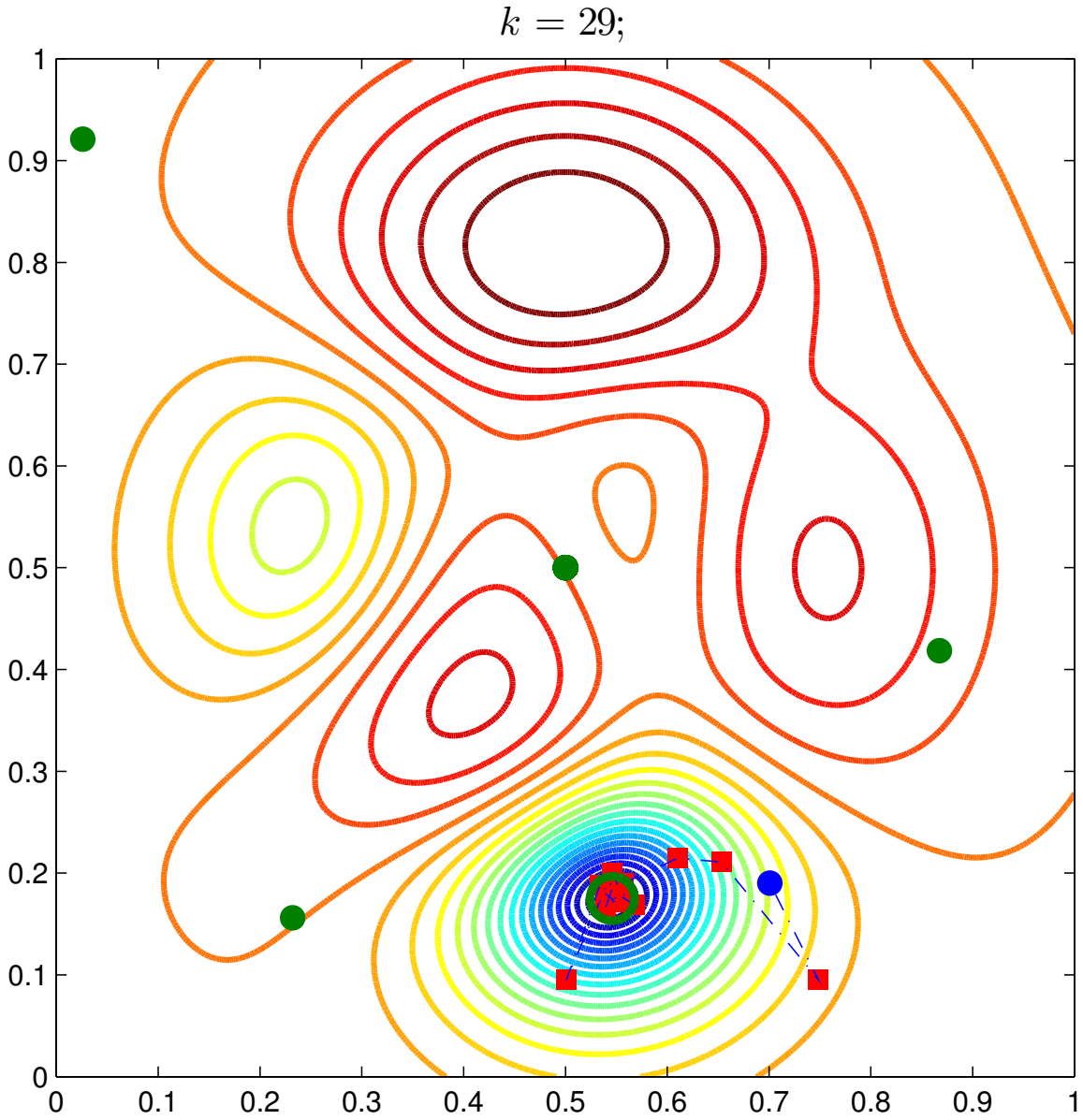
# Derivative-free MLSL



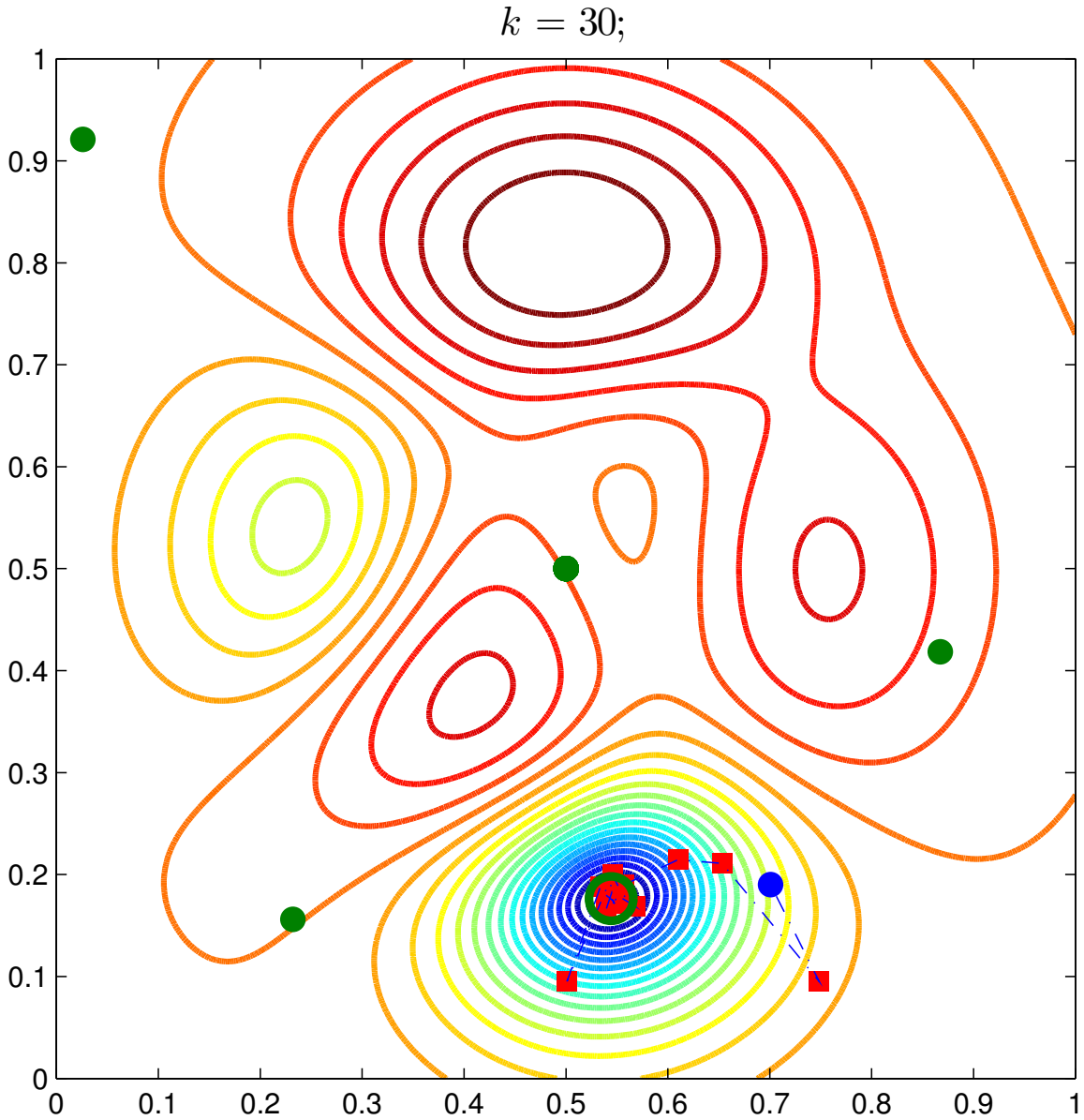
# Derivative-free MLSL



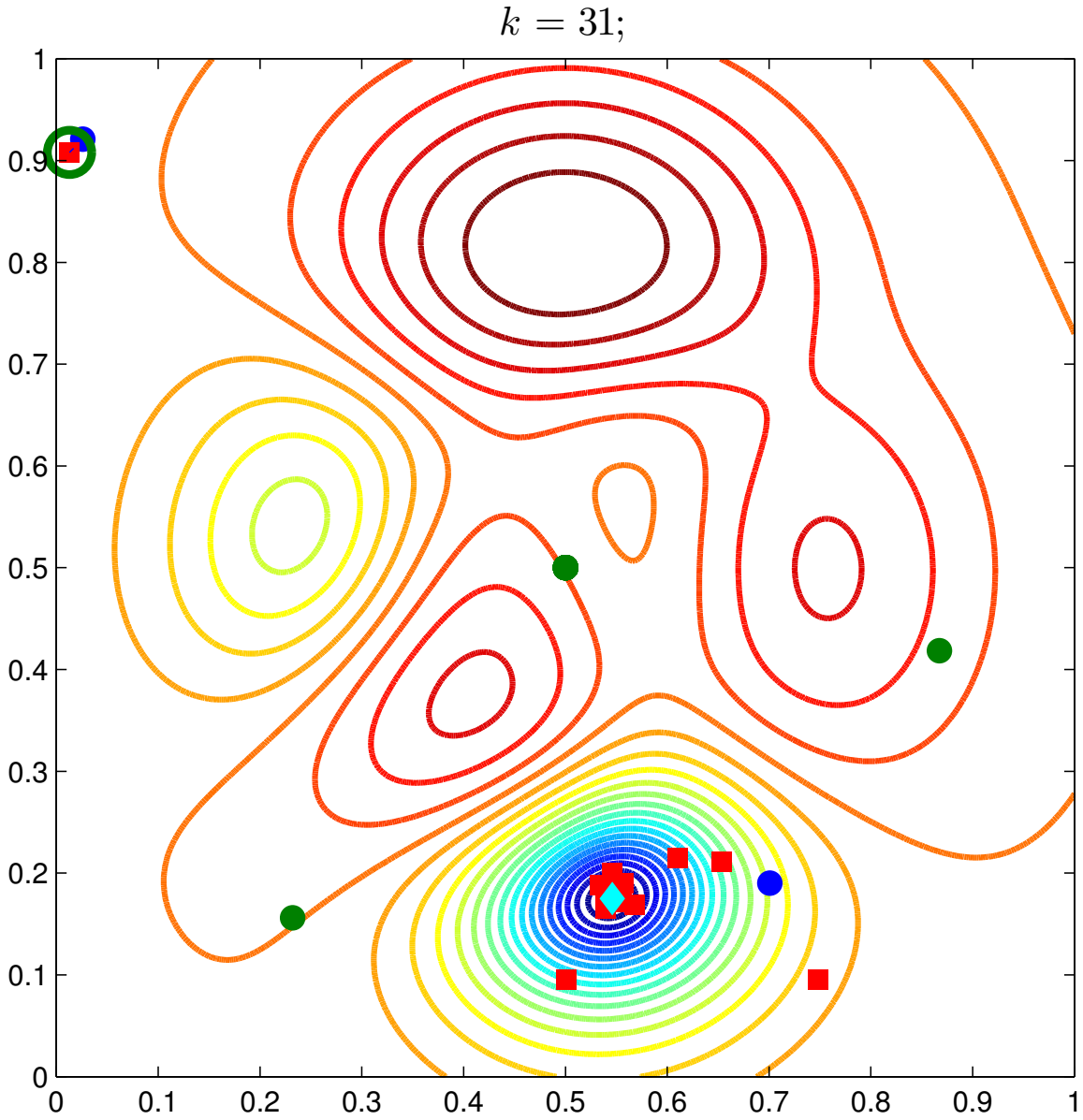
# Derivative-free MLSL



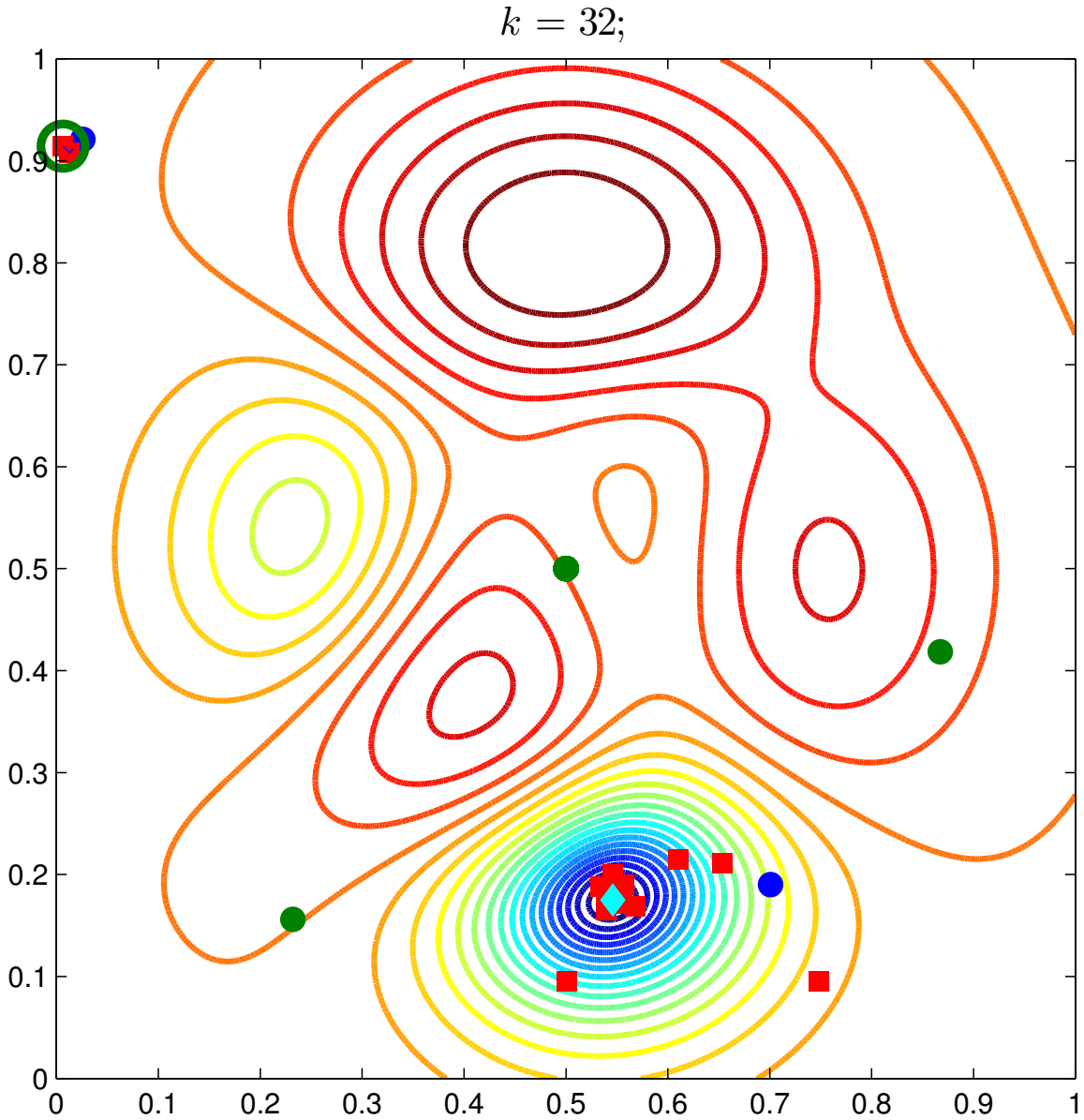
# Derivative-free MLSL



# Derivative-free MLSL



# Derivative-free MLSL

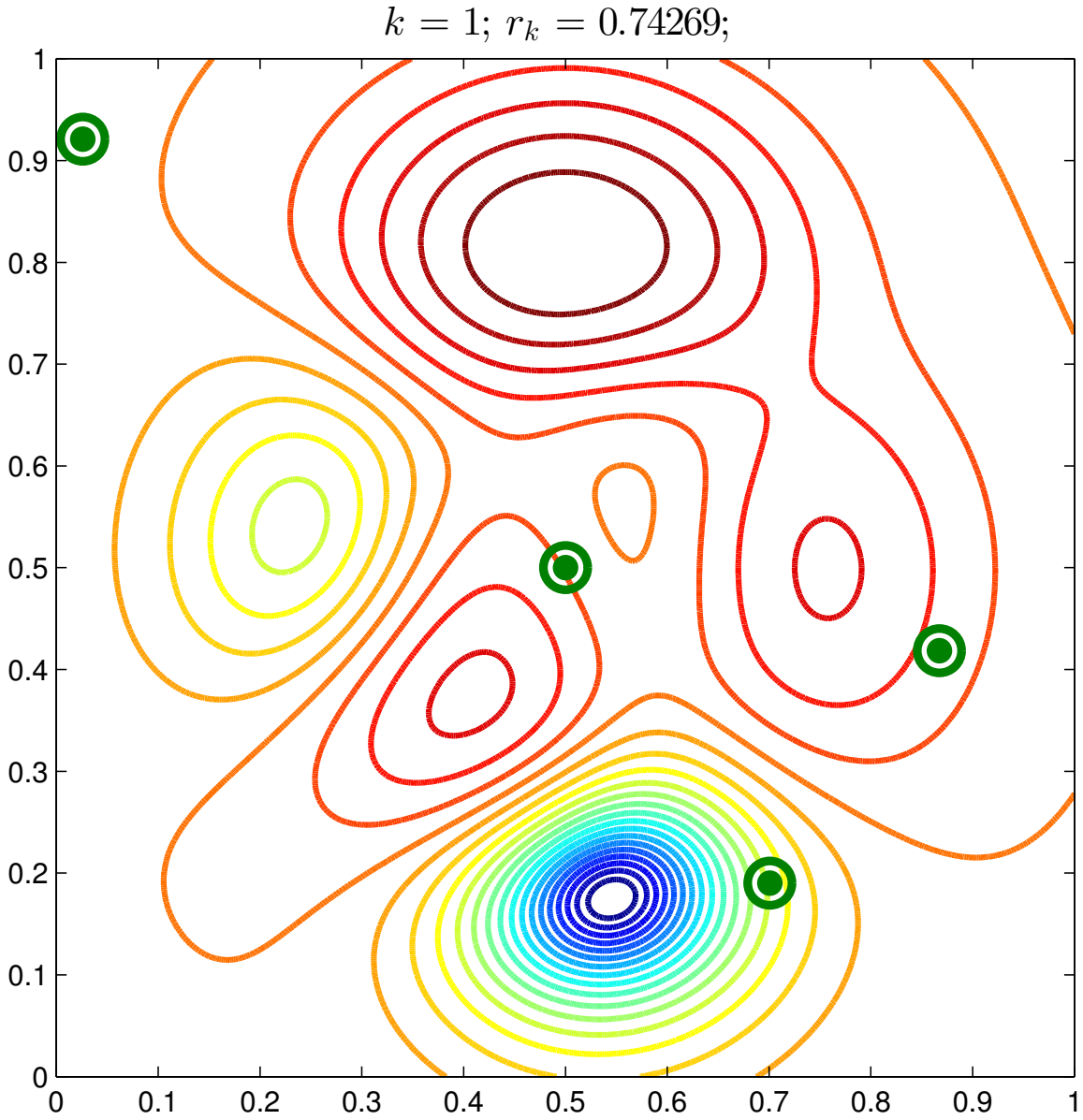


# Derivative-free MLSL

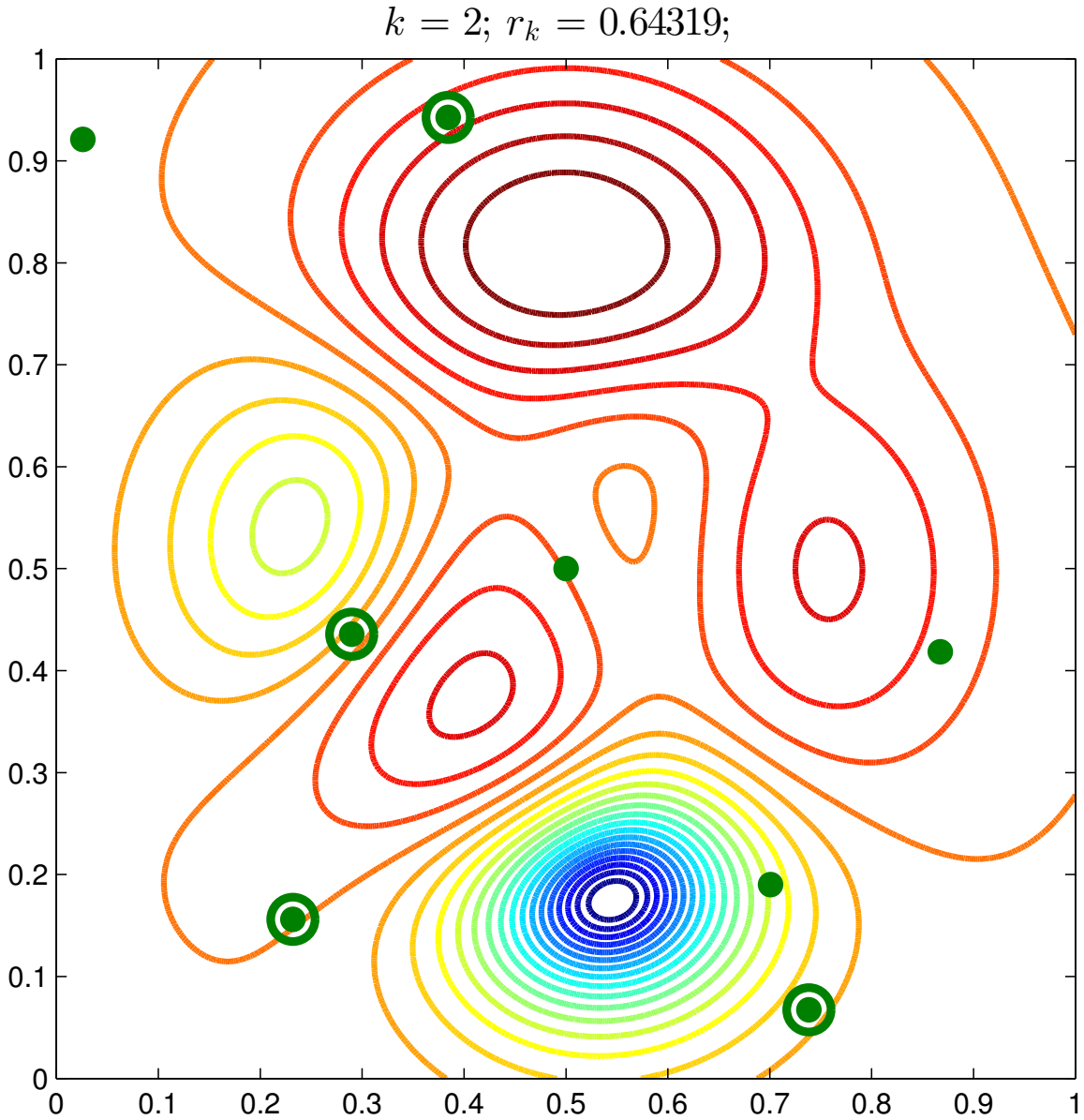
- If we had the ability to concurrently evaluate points, what might be the best course?

- Is it better to do e,g., finite differences computations or separate localopt runs?

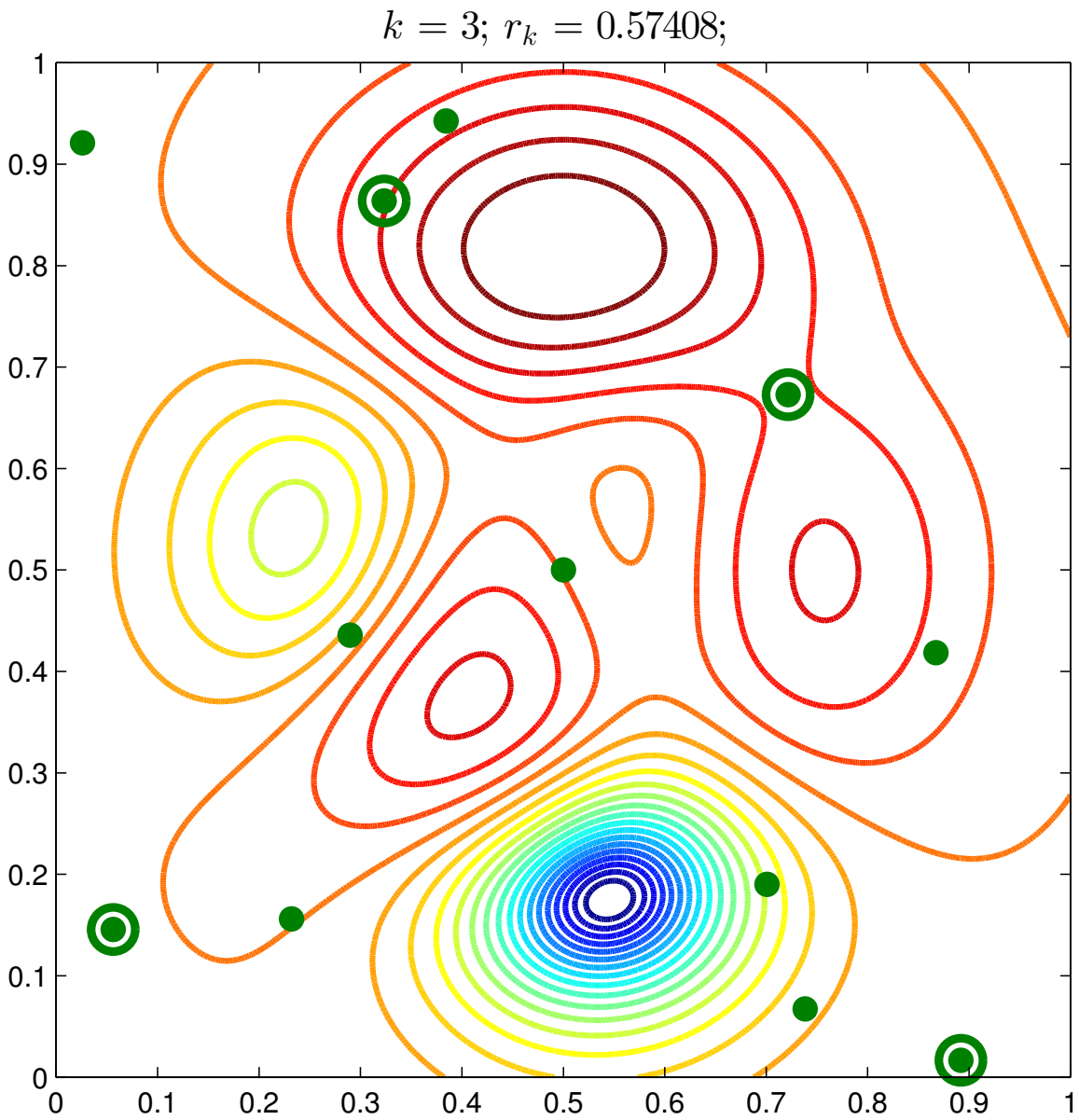
# Our Method



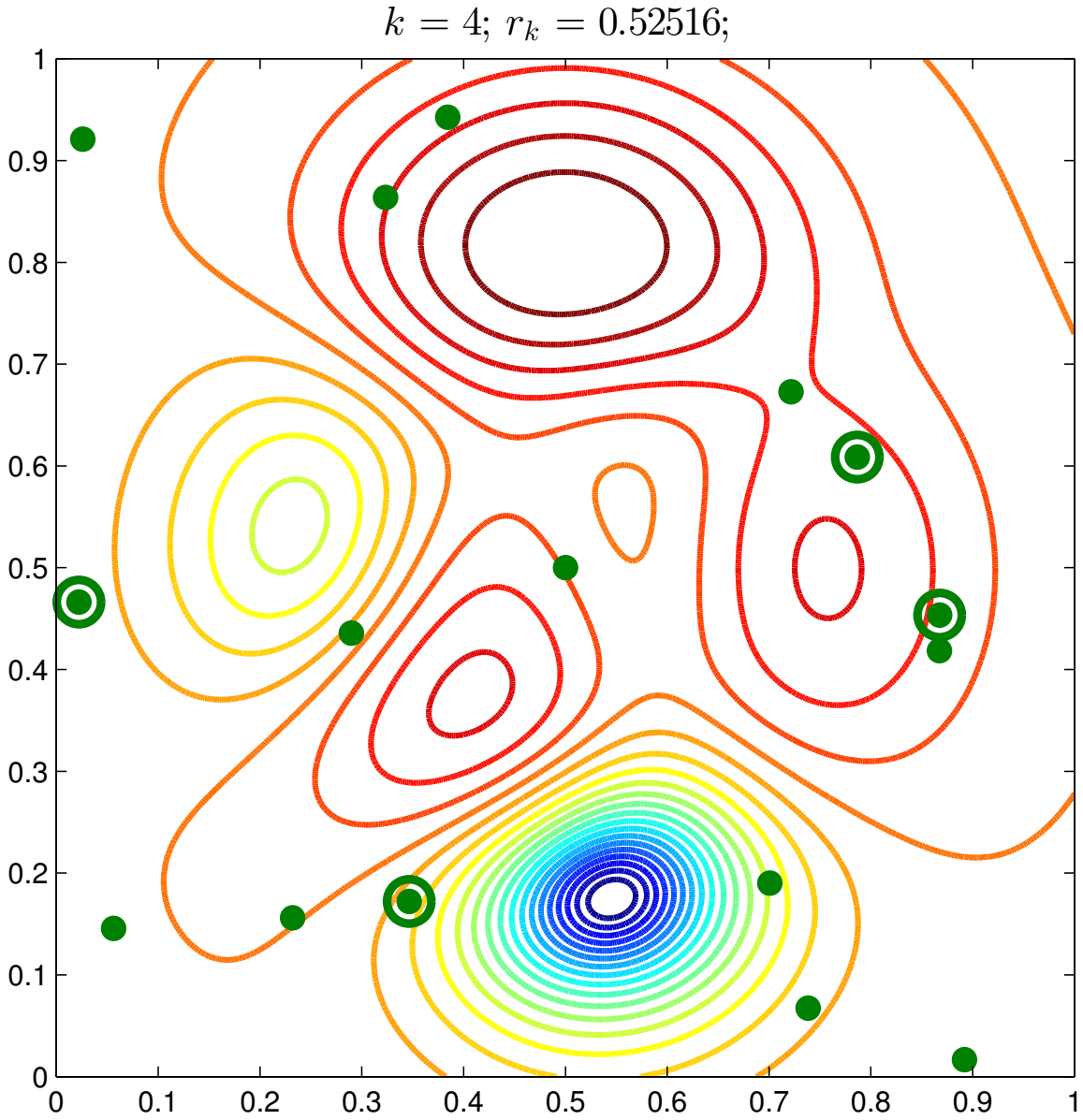
# Our Method



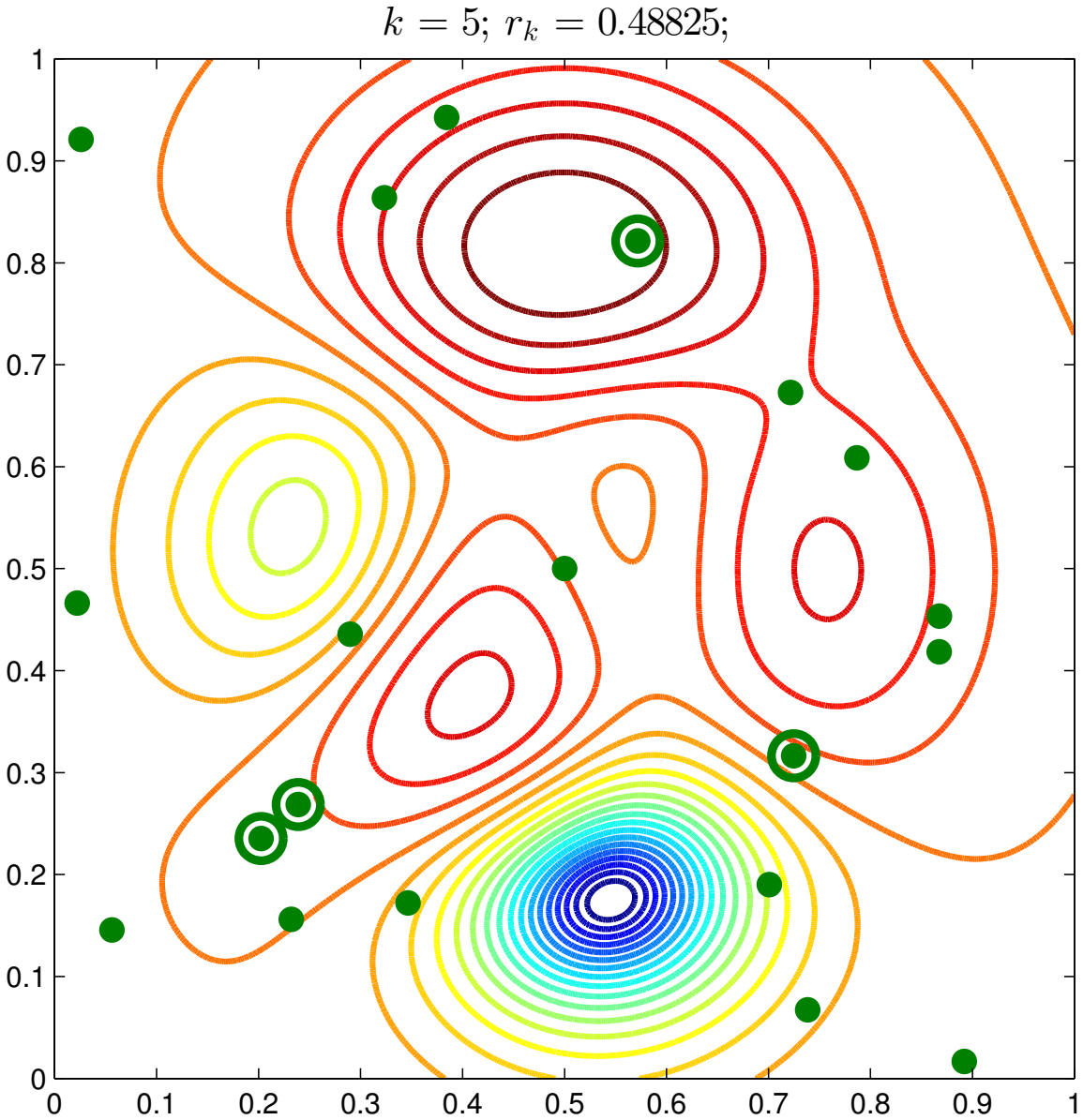
# Our Method



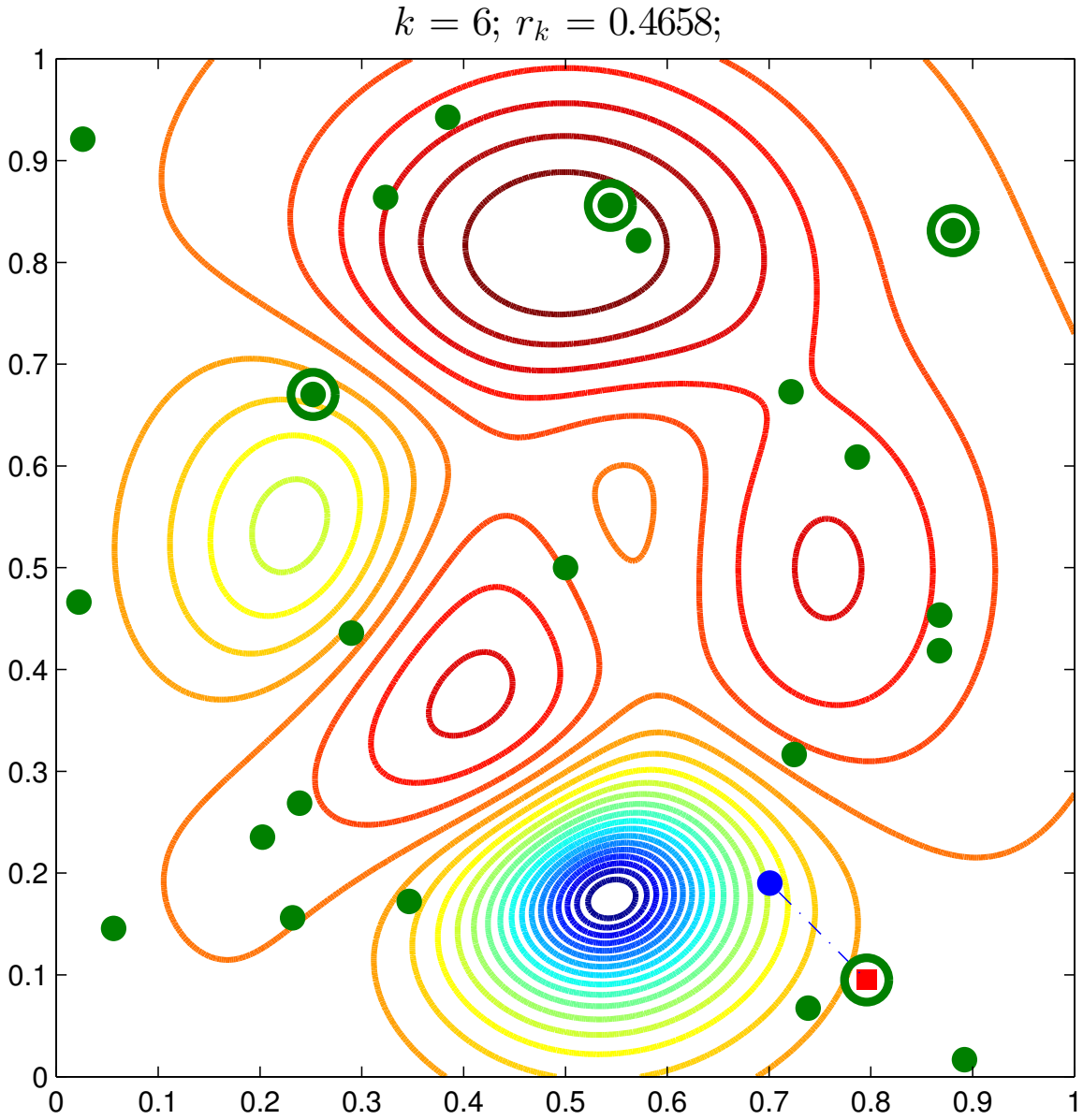
# Our Method



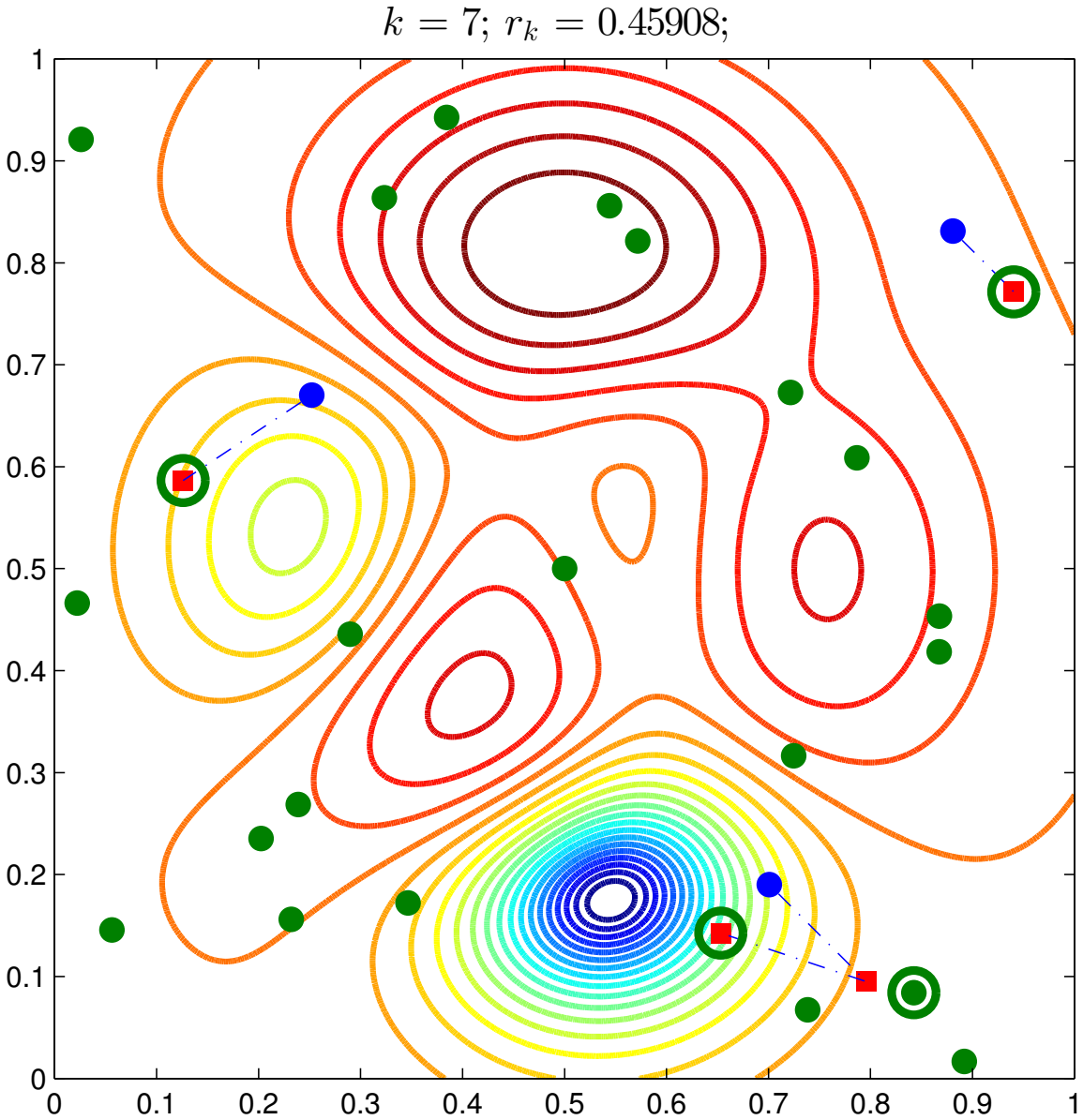
# Our Method



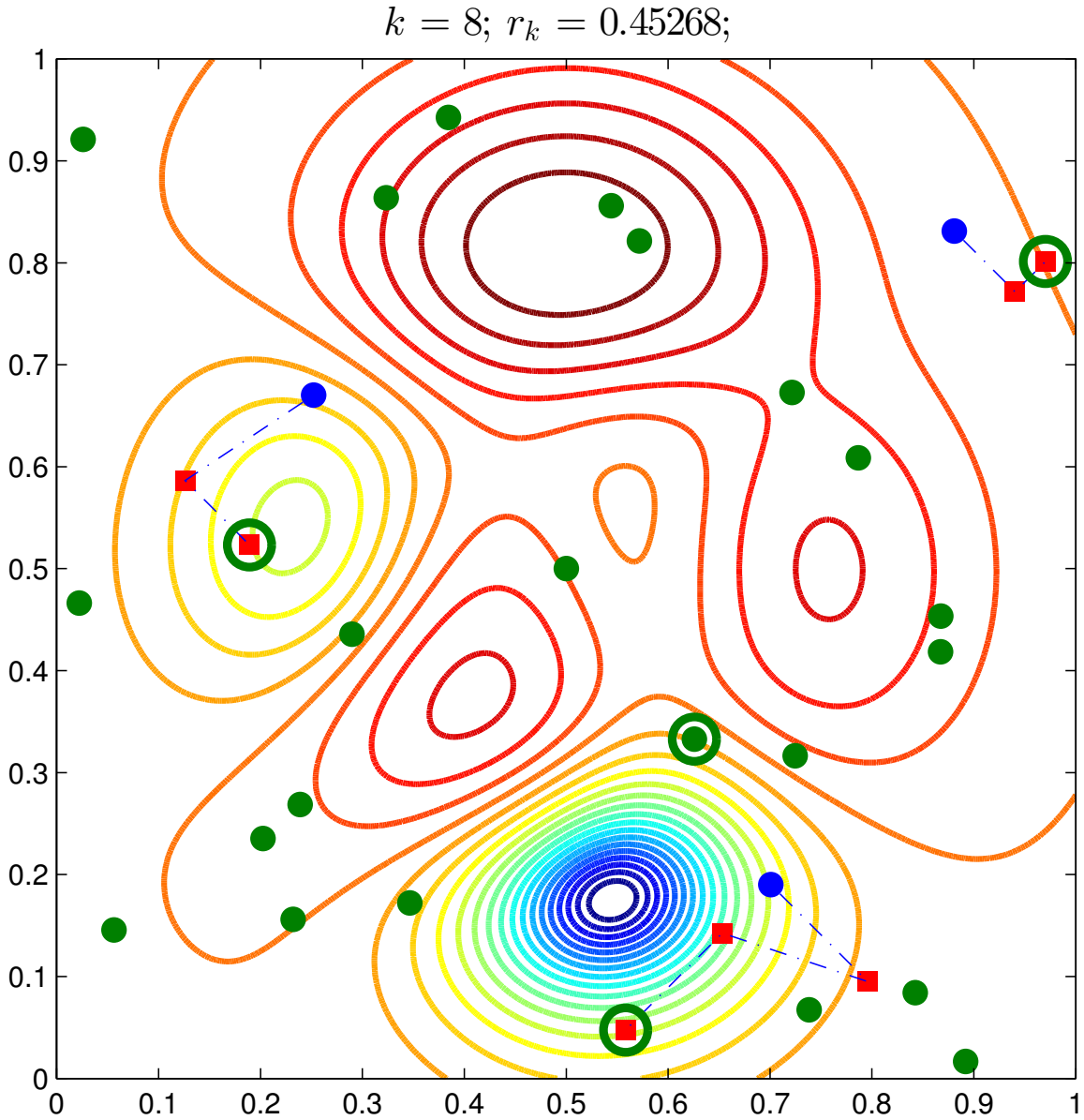
# Our Method



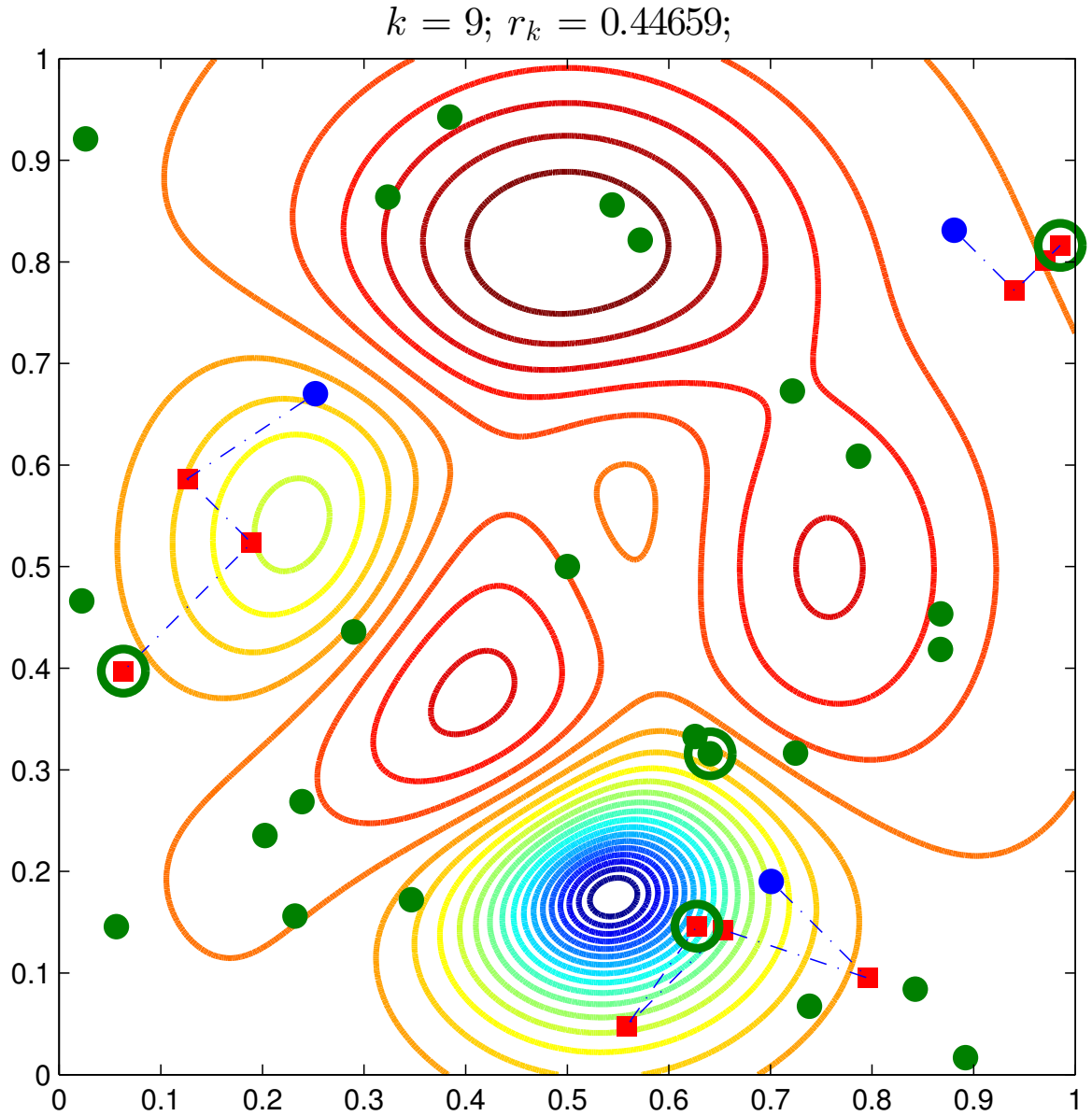
# Our Method



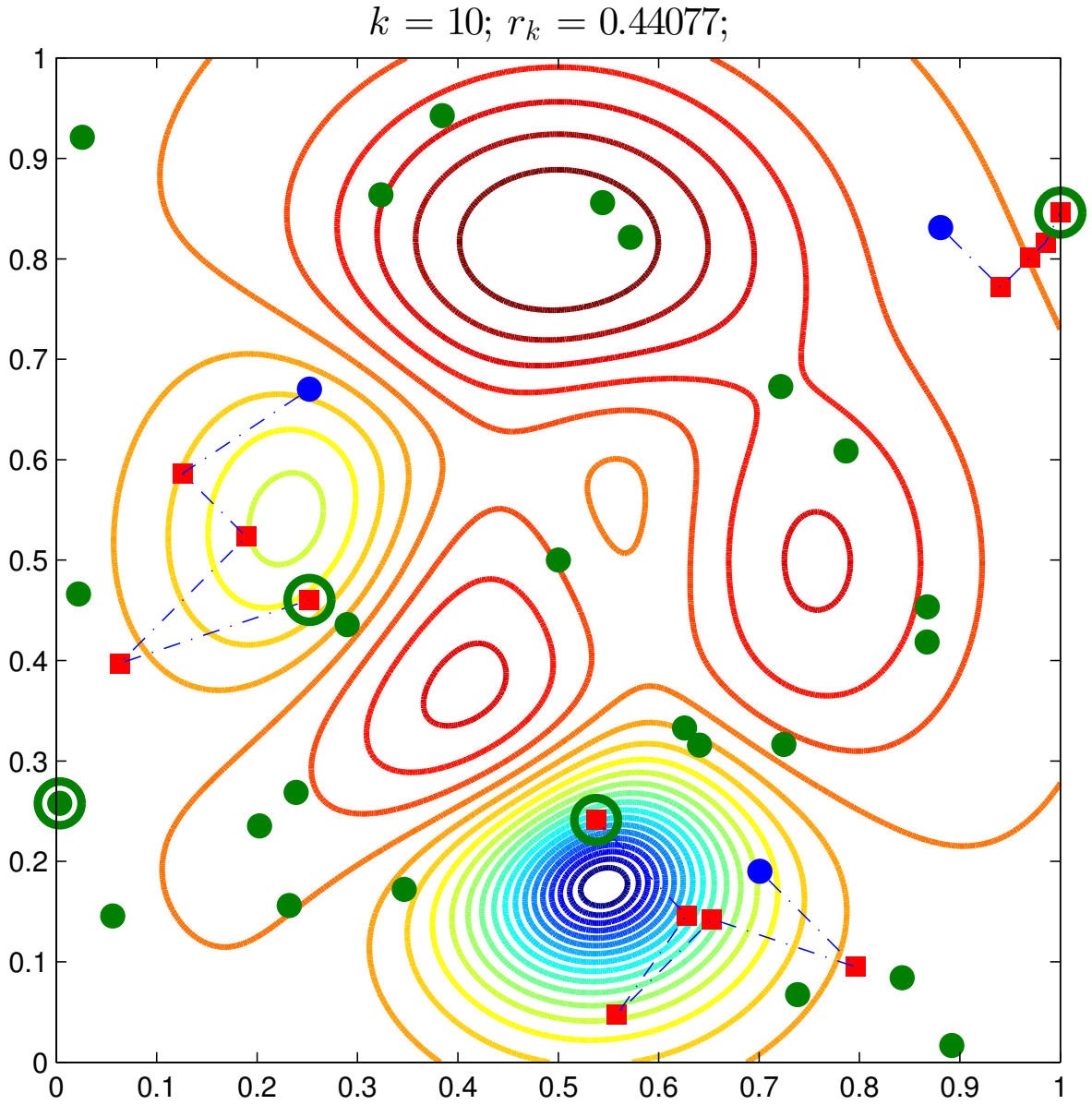
# Our Method



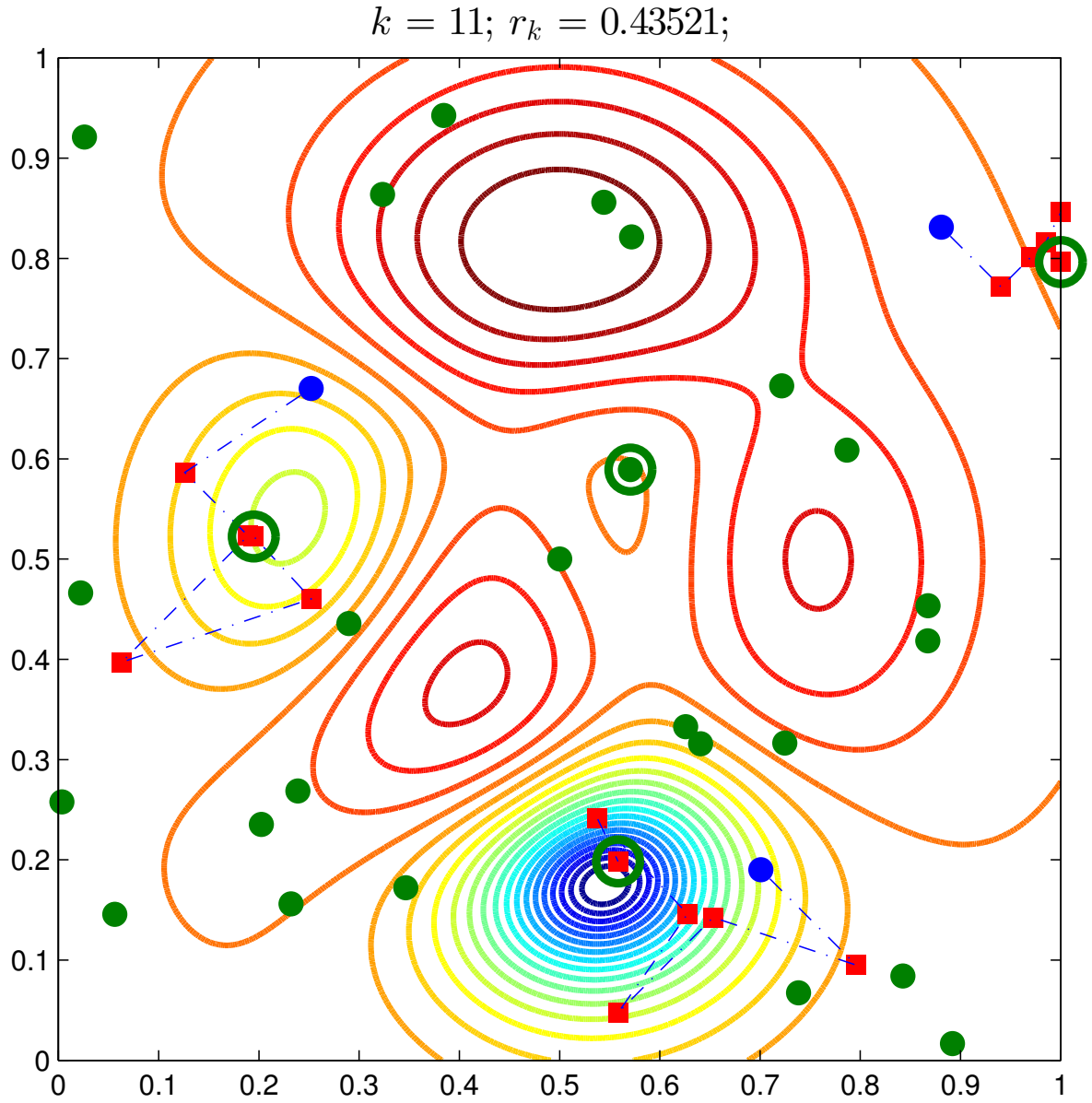
# Our Method



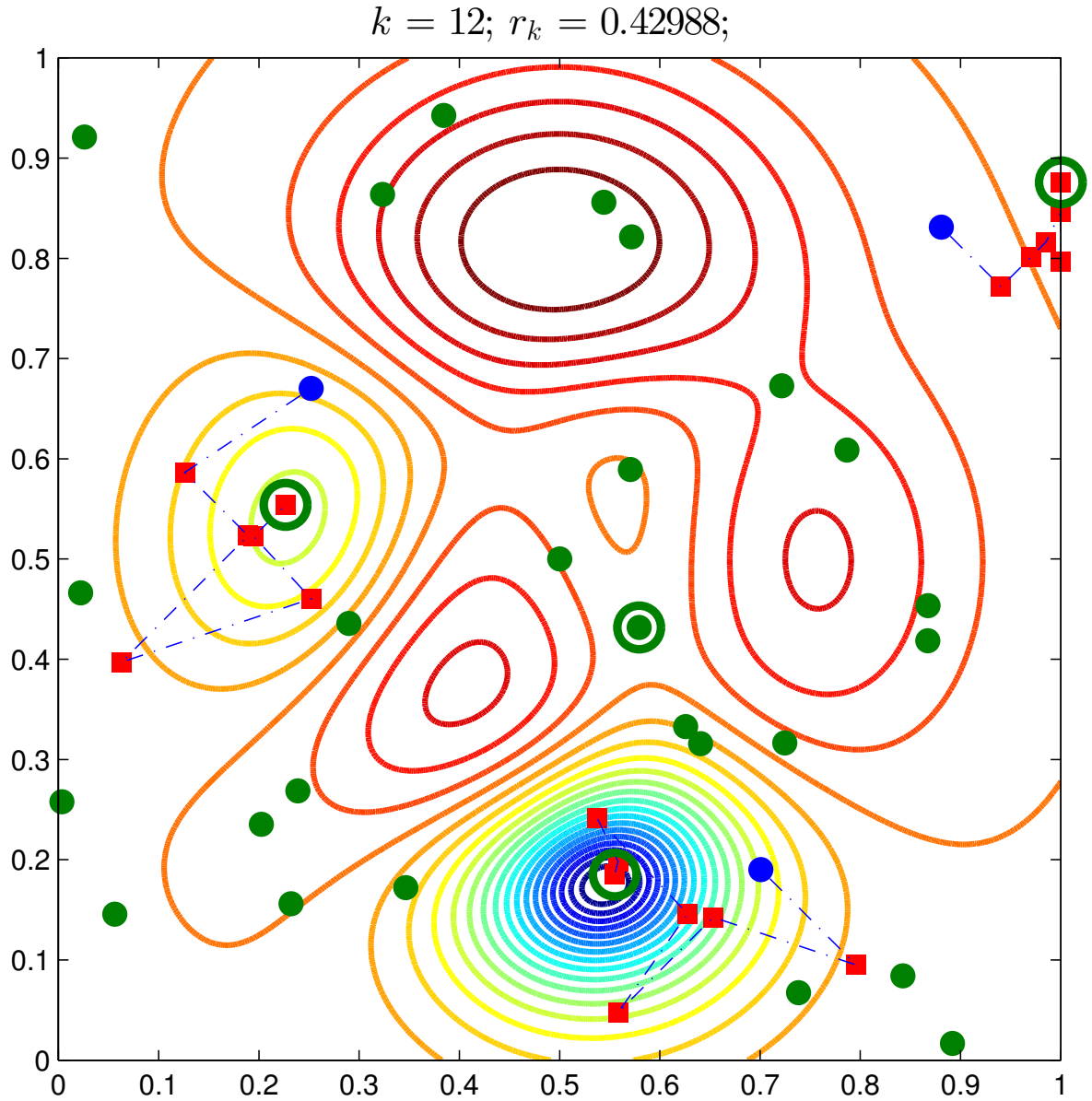
# Our Method



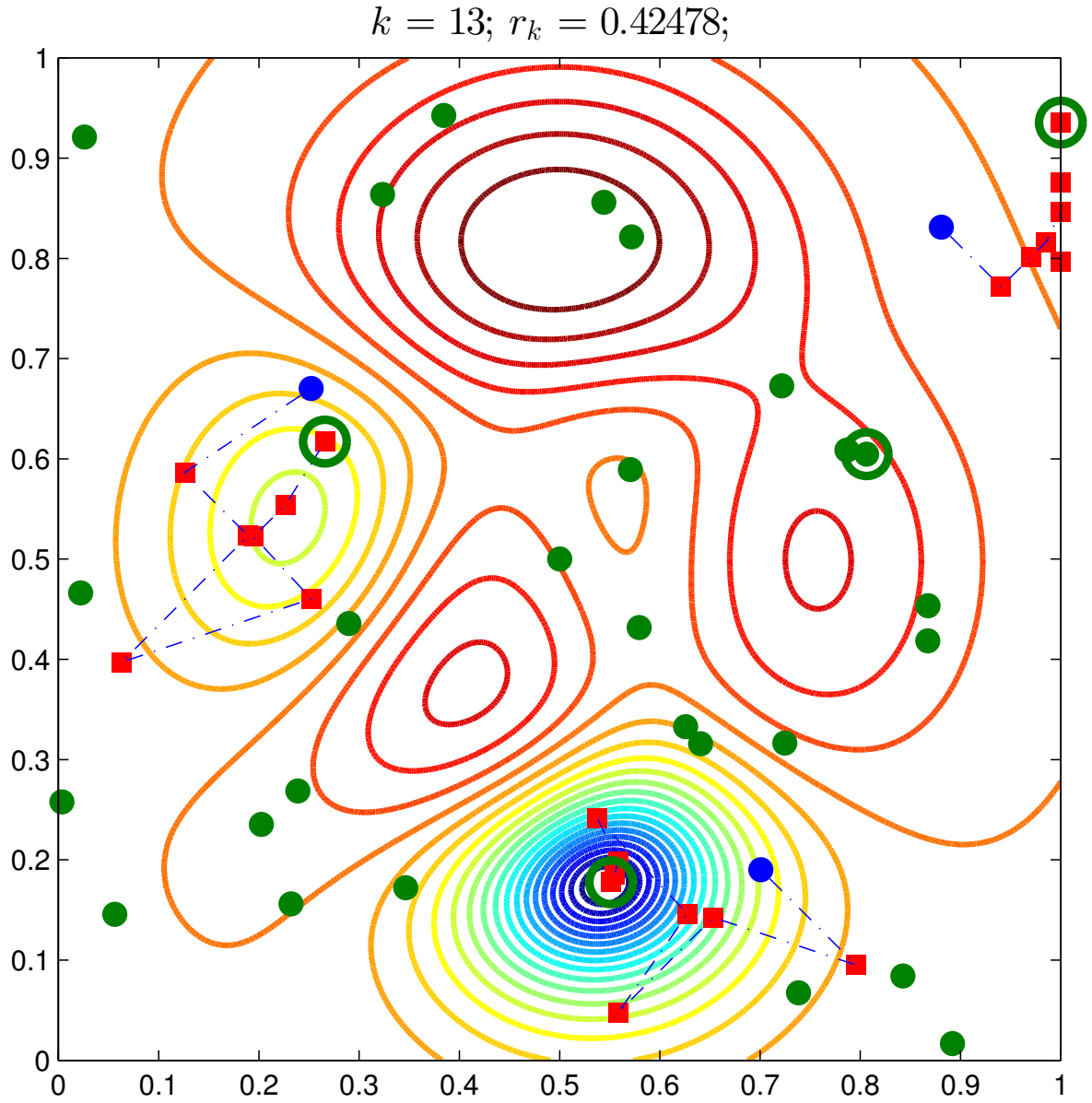
# Our Method



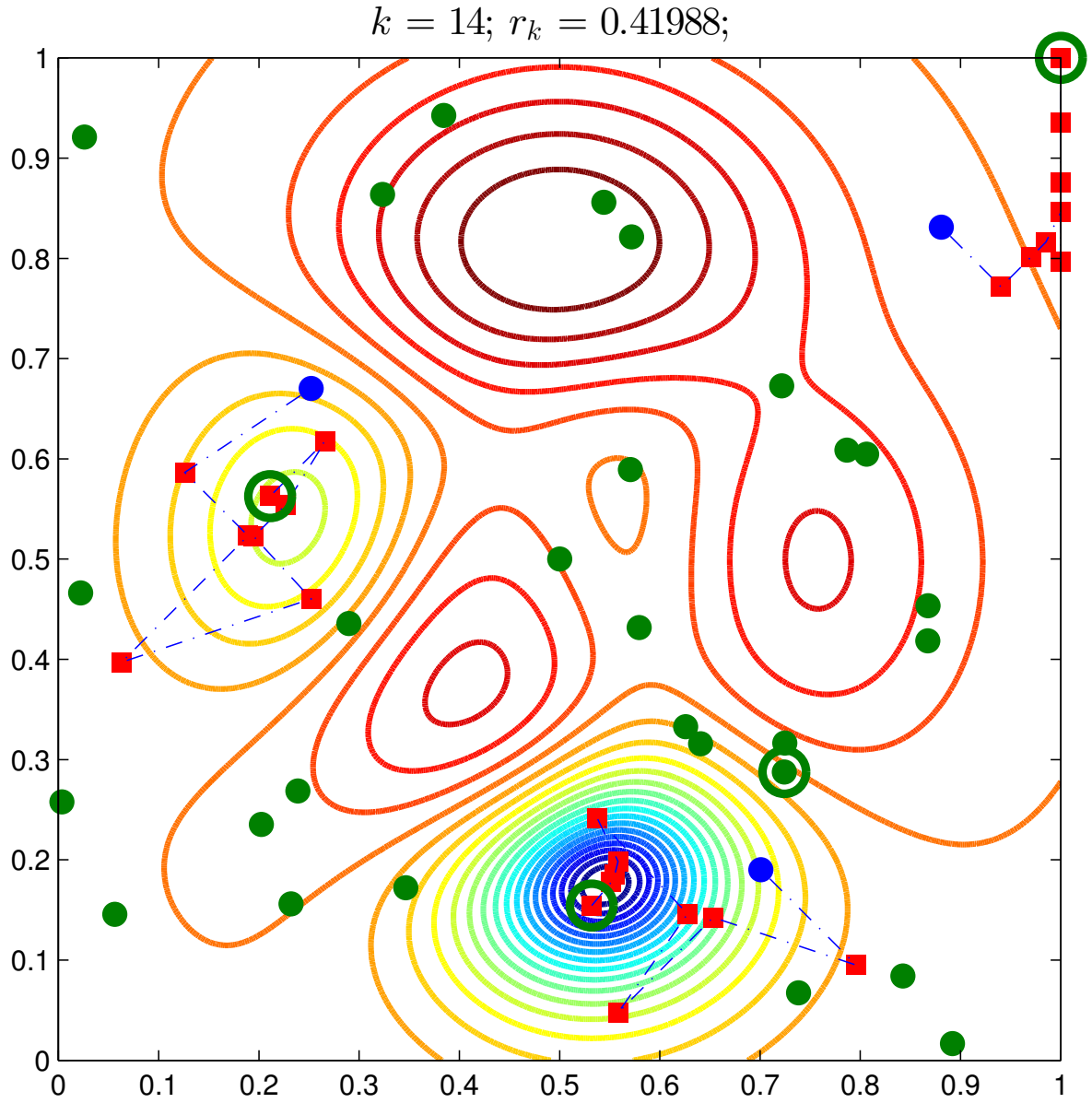
# Our Method



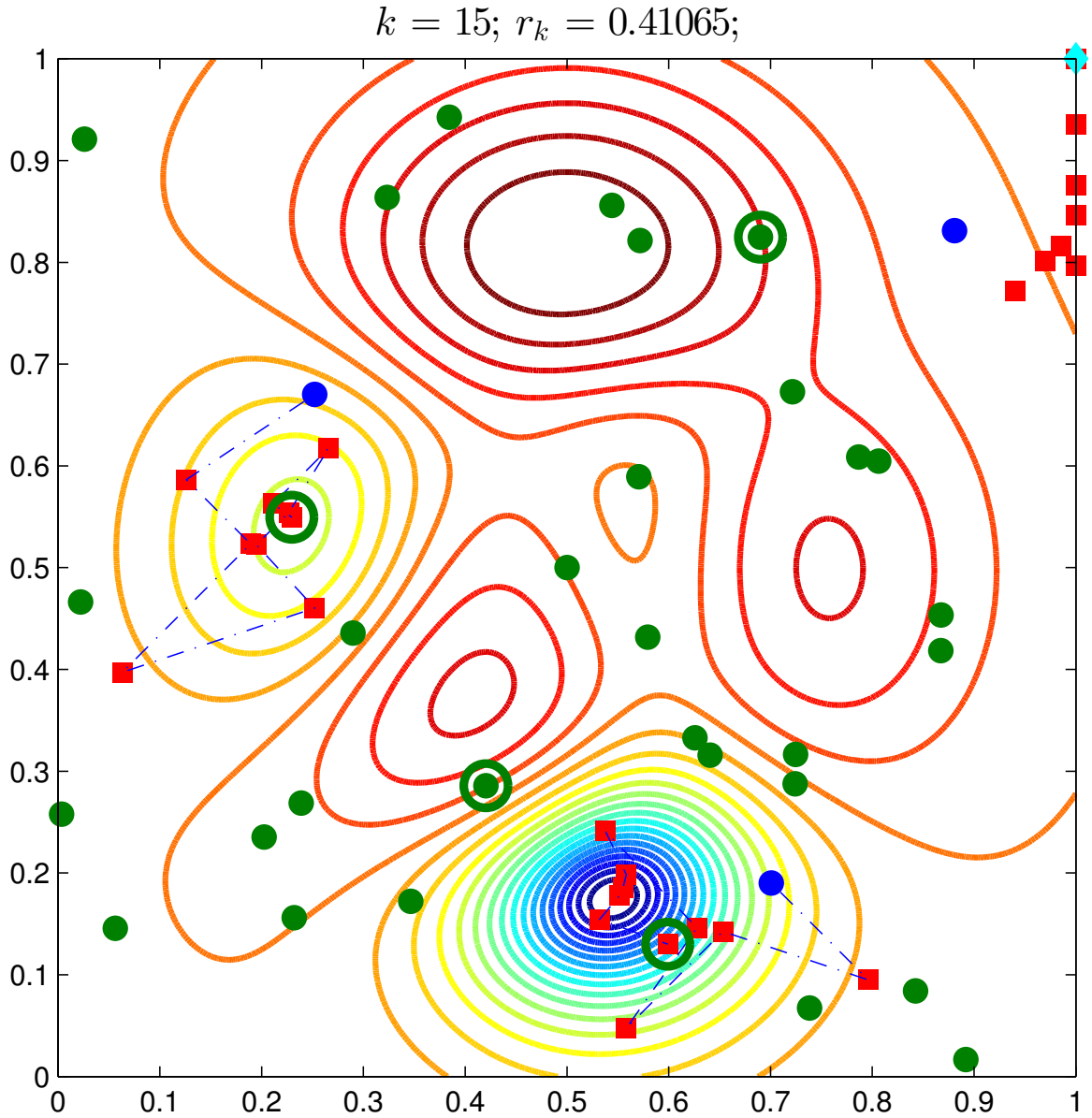
# Our Method



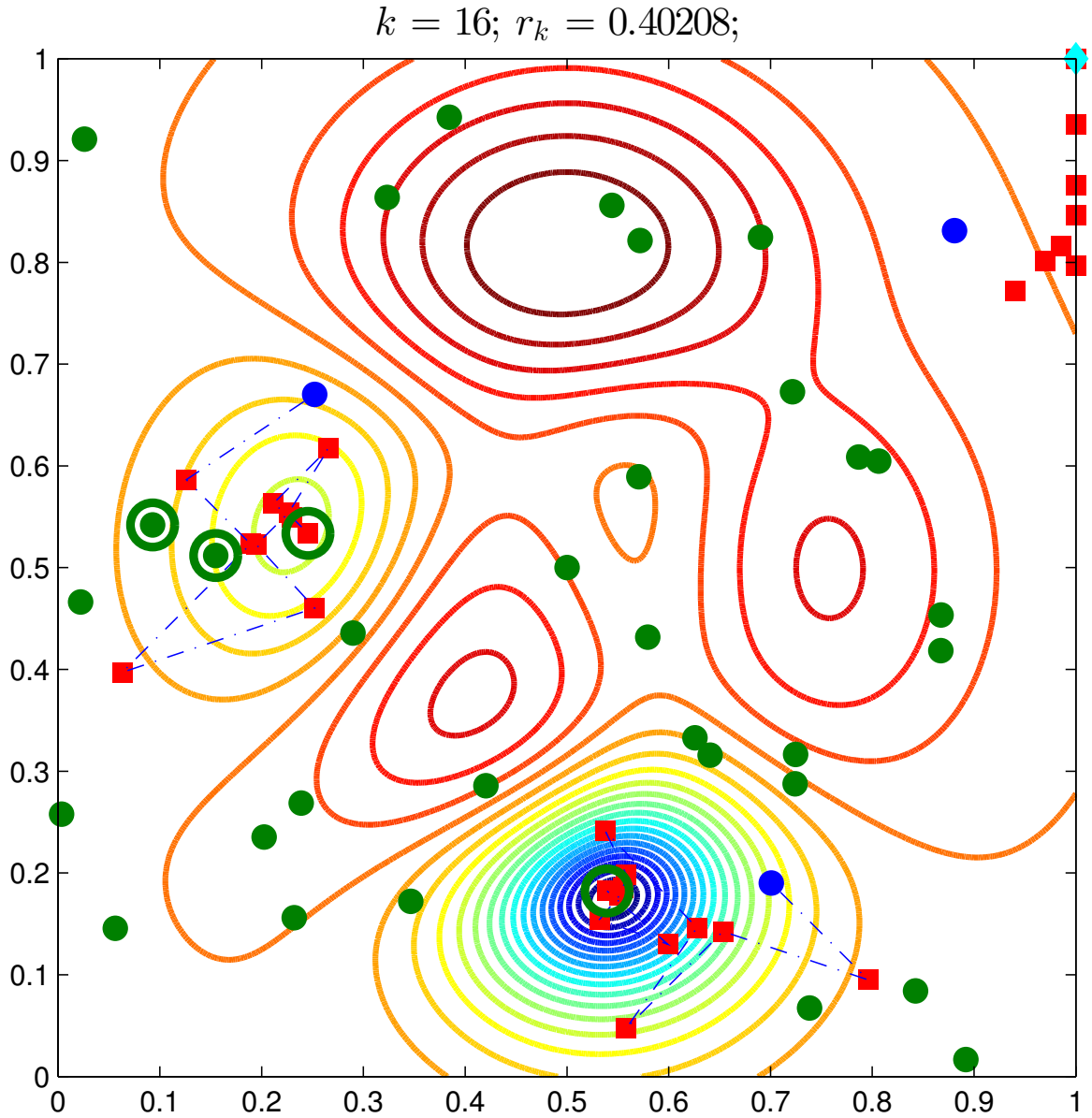
# Our Method



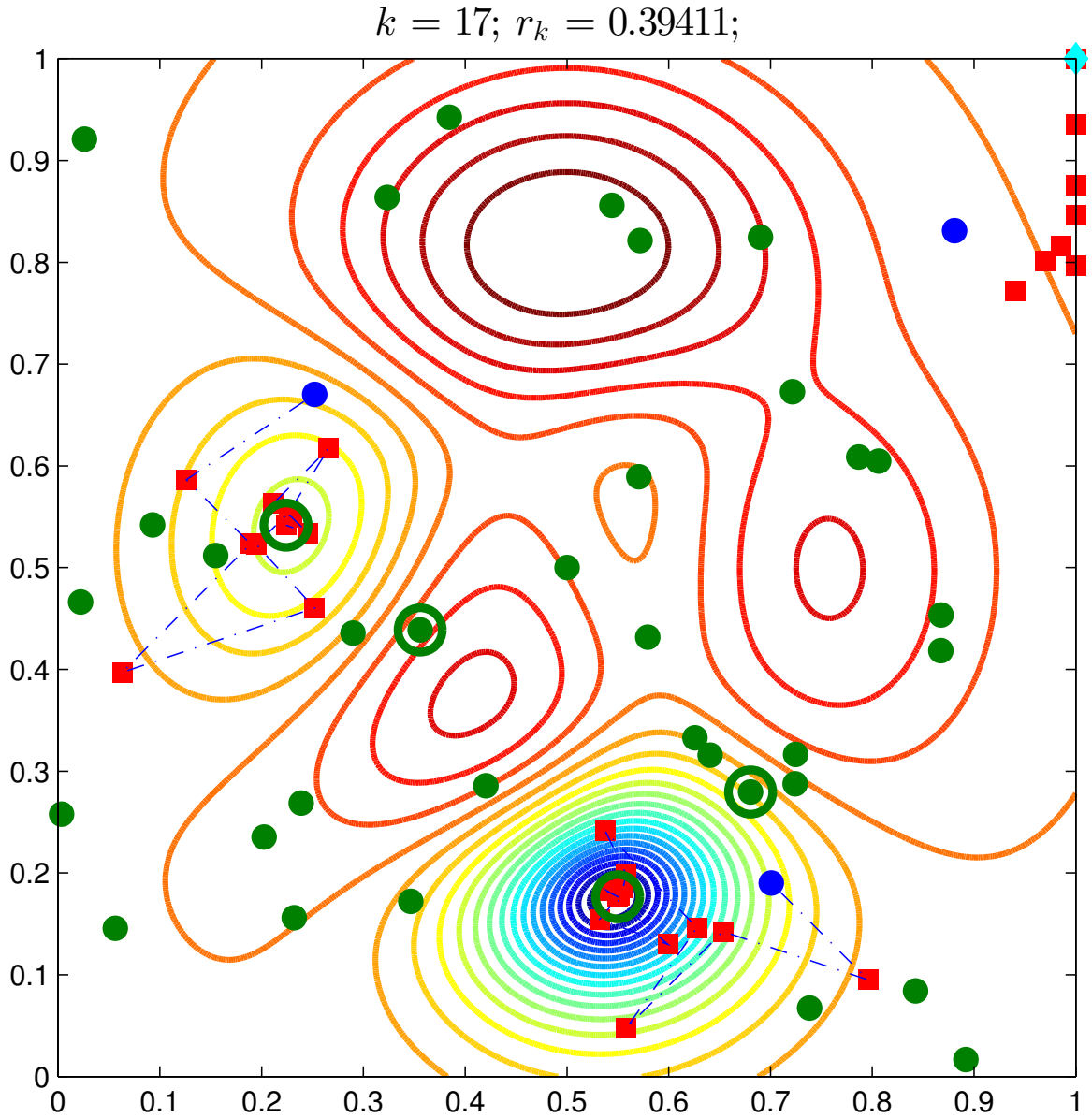
# Our Method



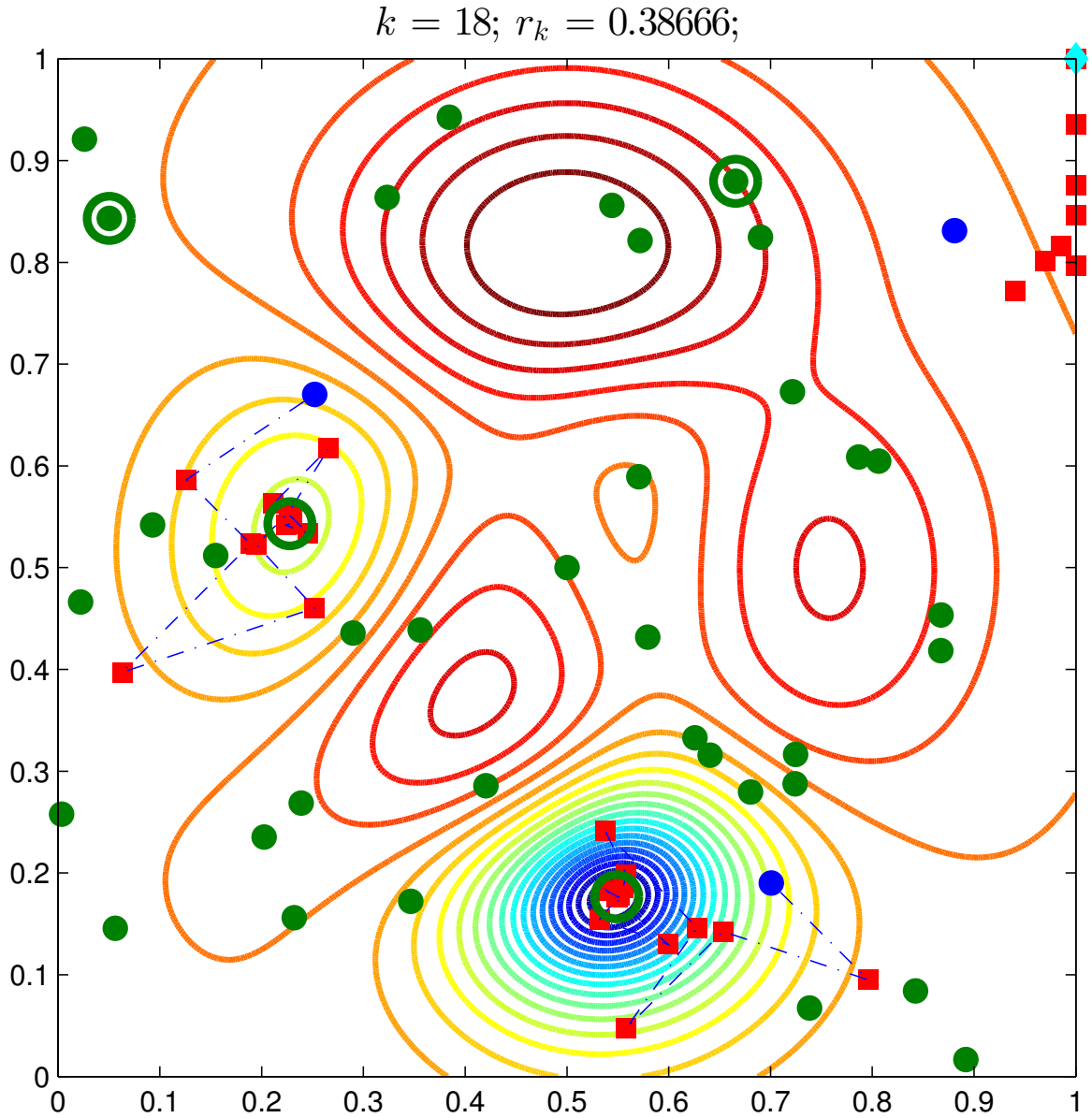
# Our Method



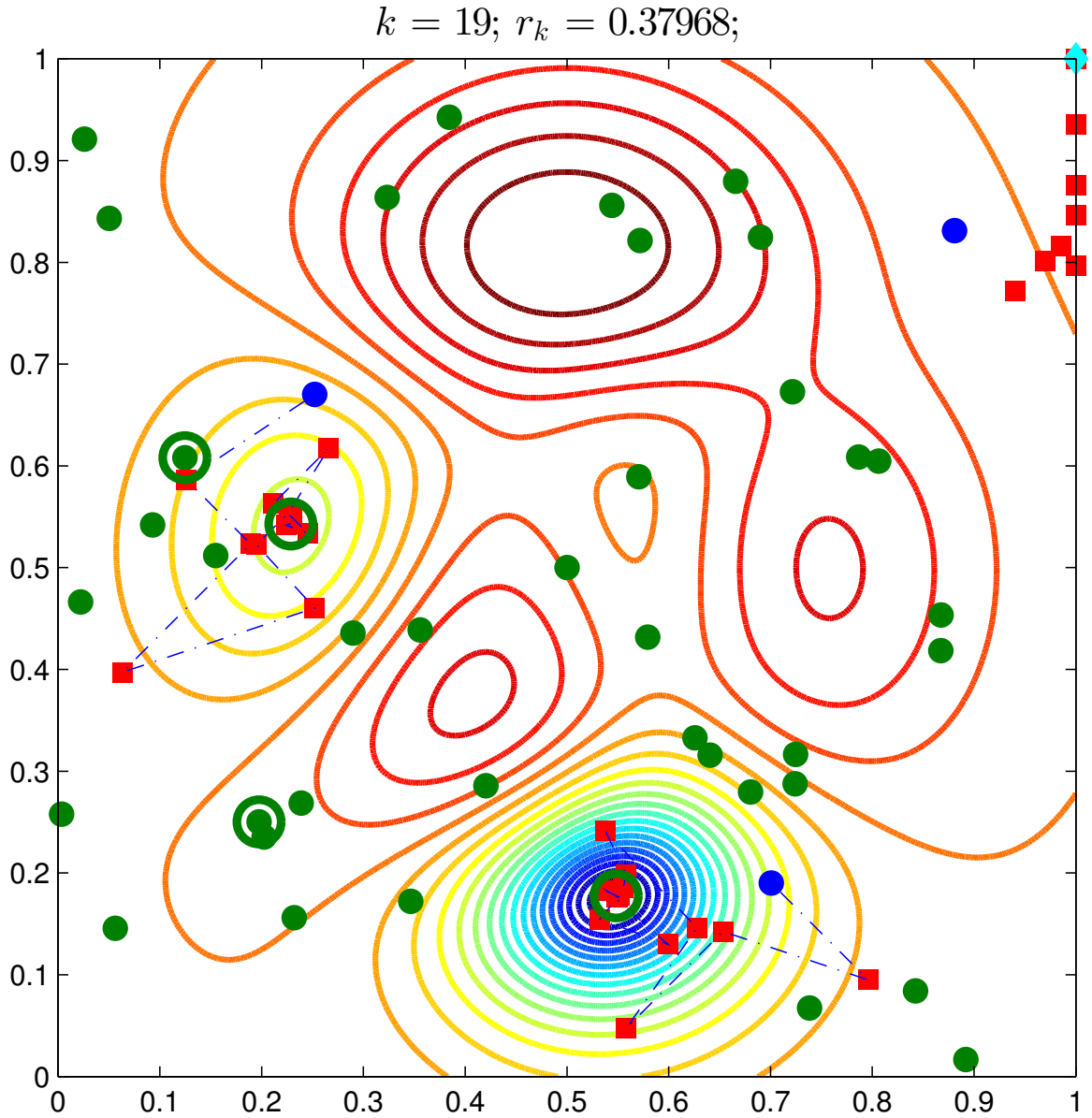
# Our Method



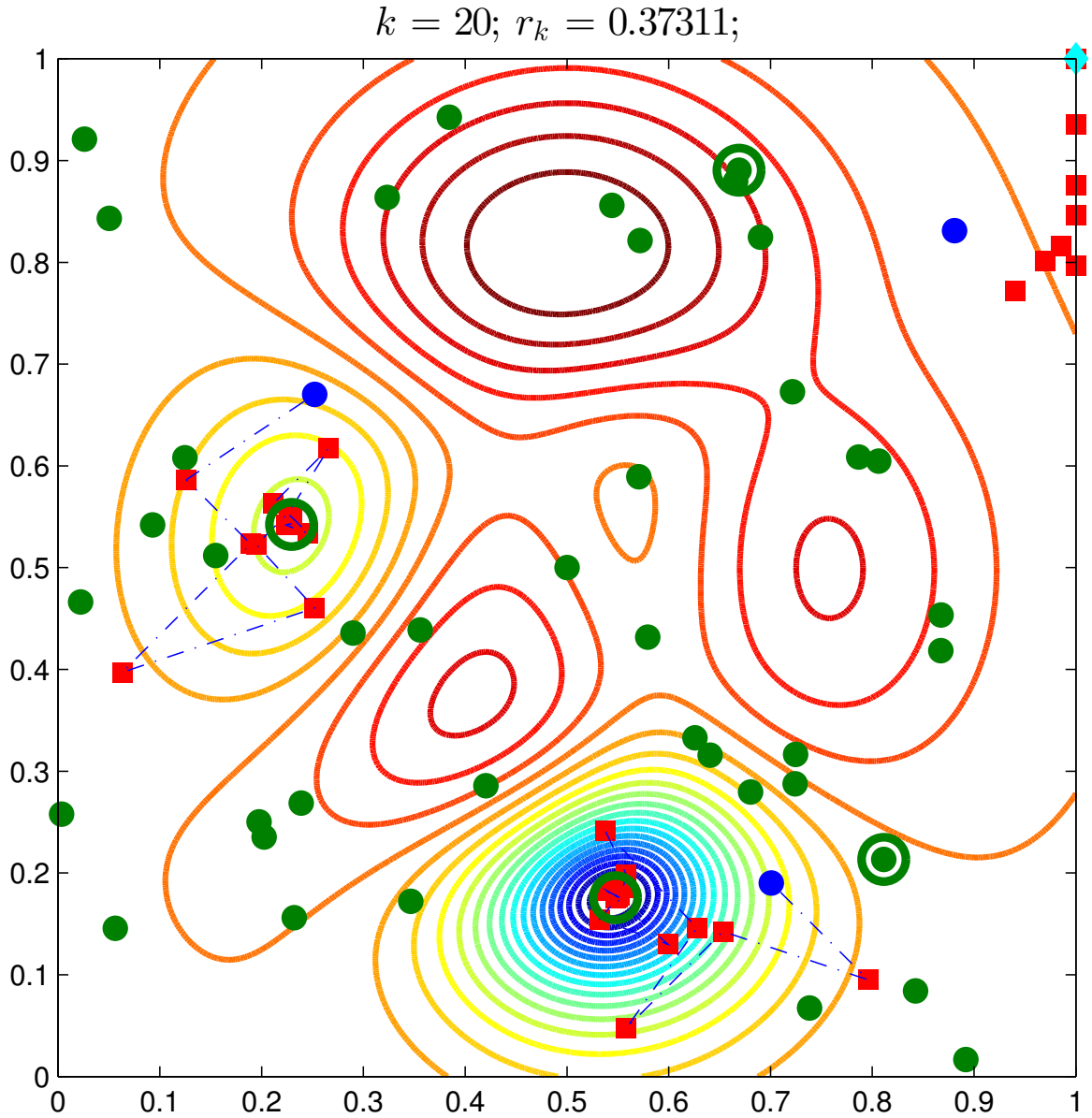
# Our Method



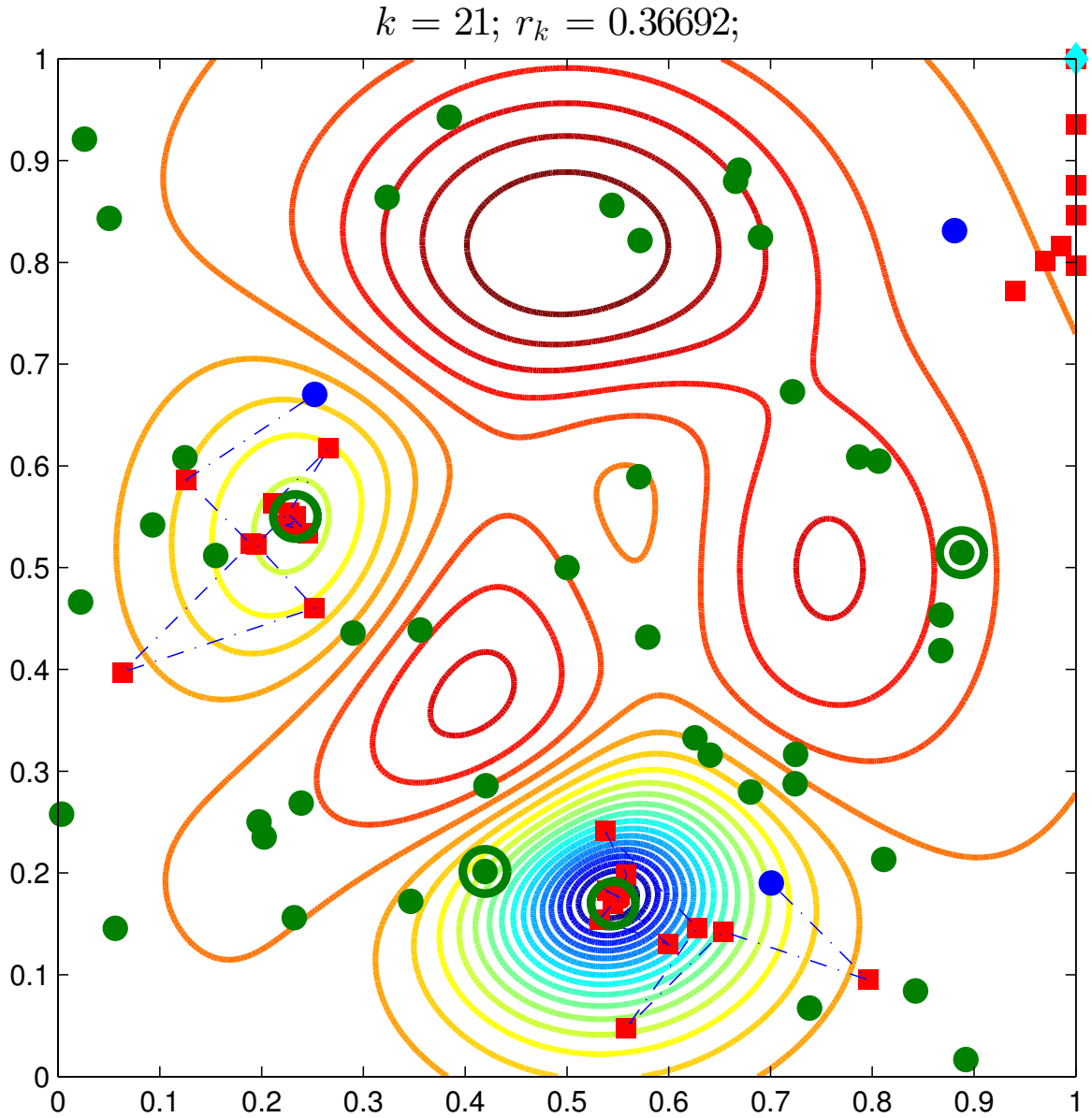
# Our Method



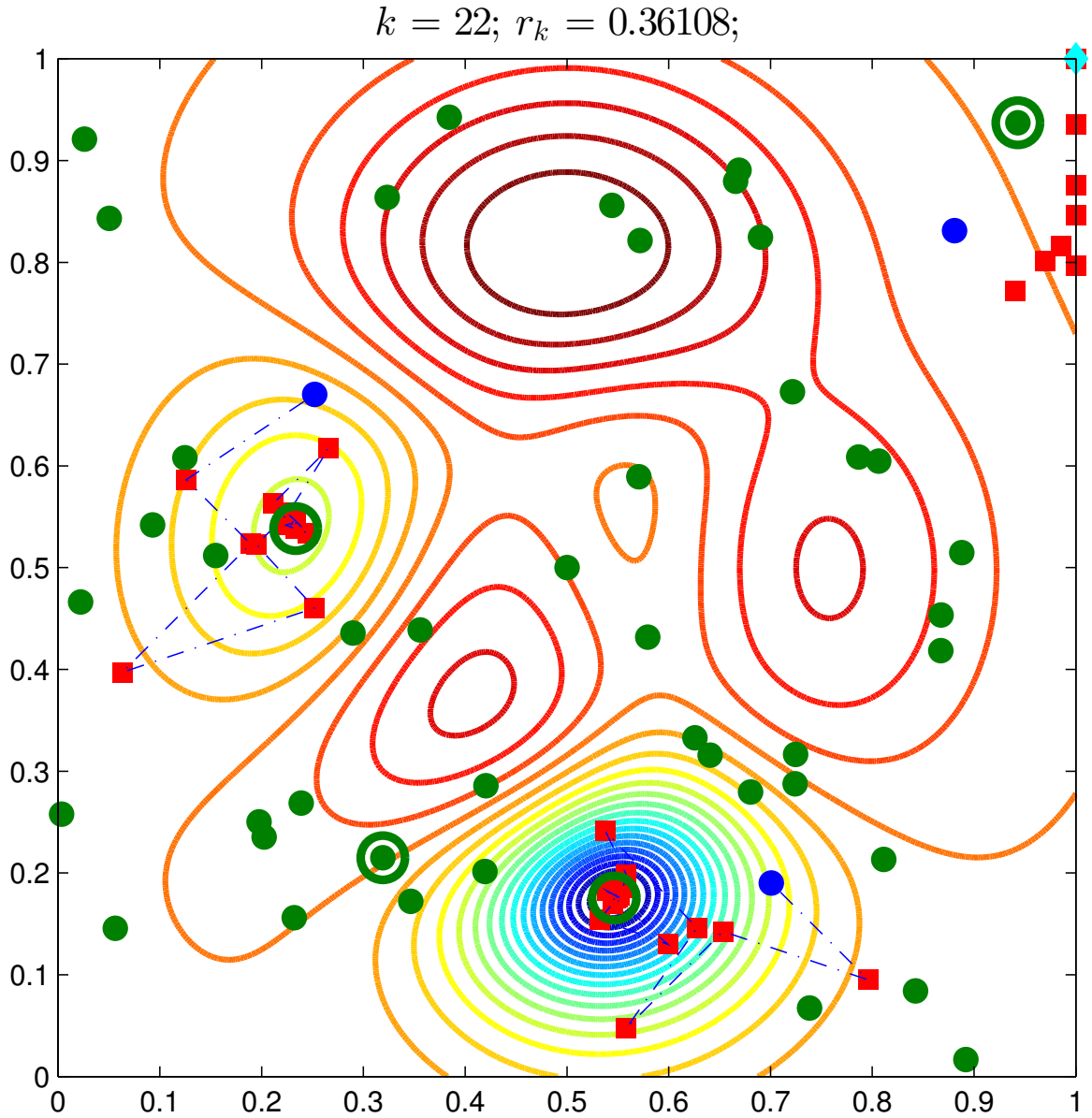
# Our Method



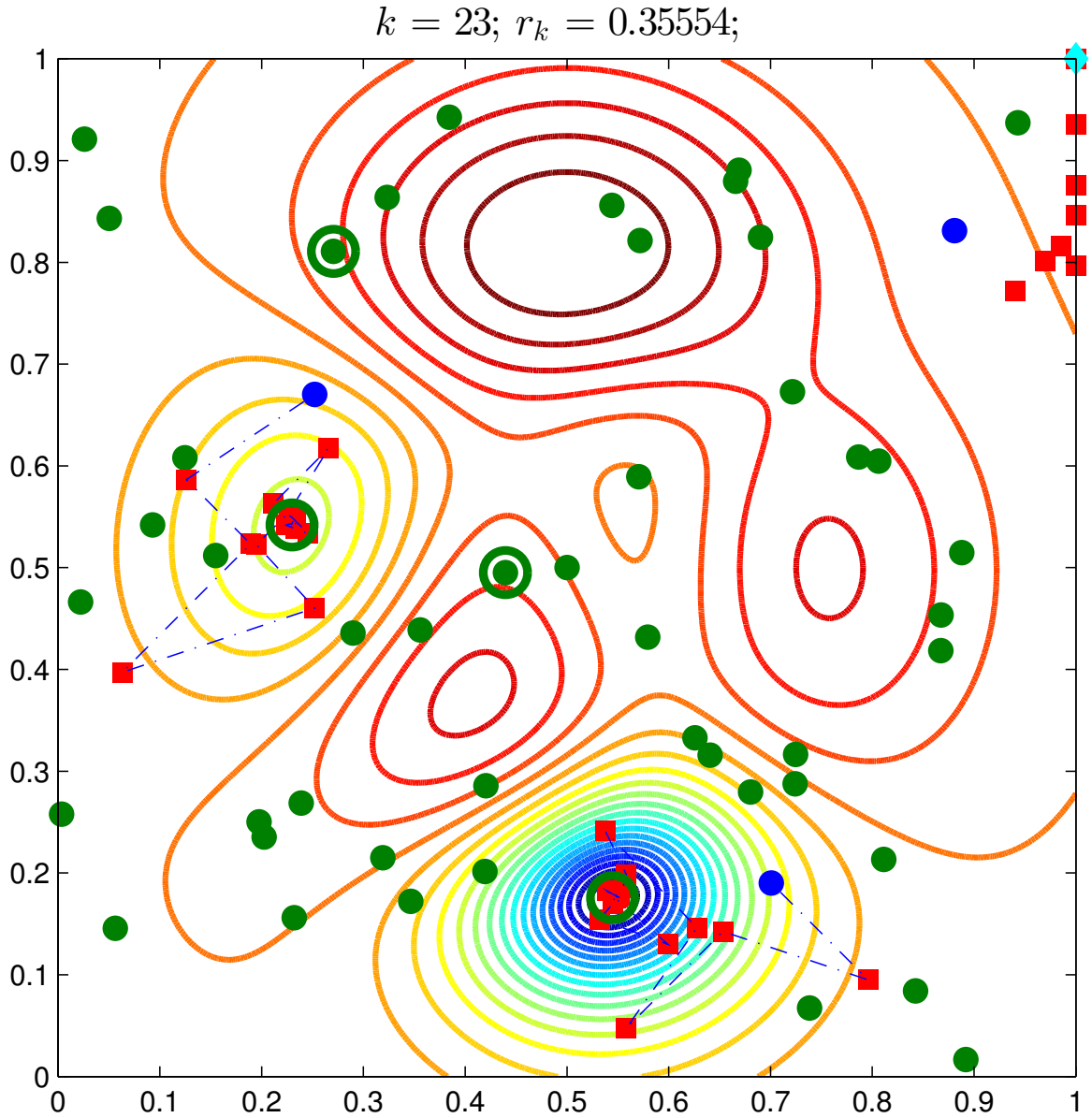
# Our Method



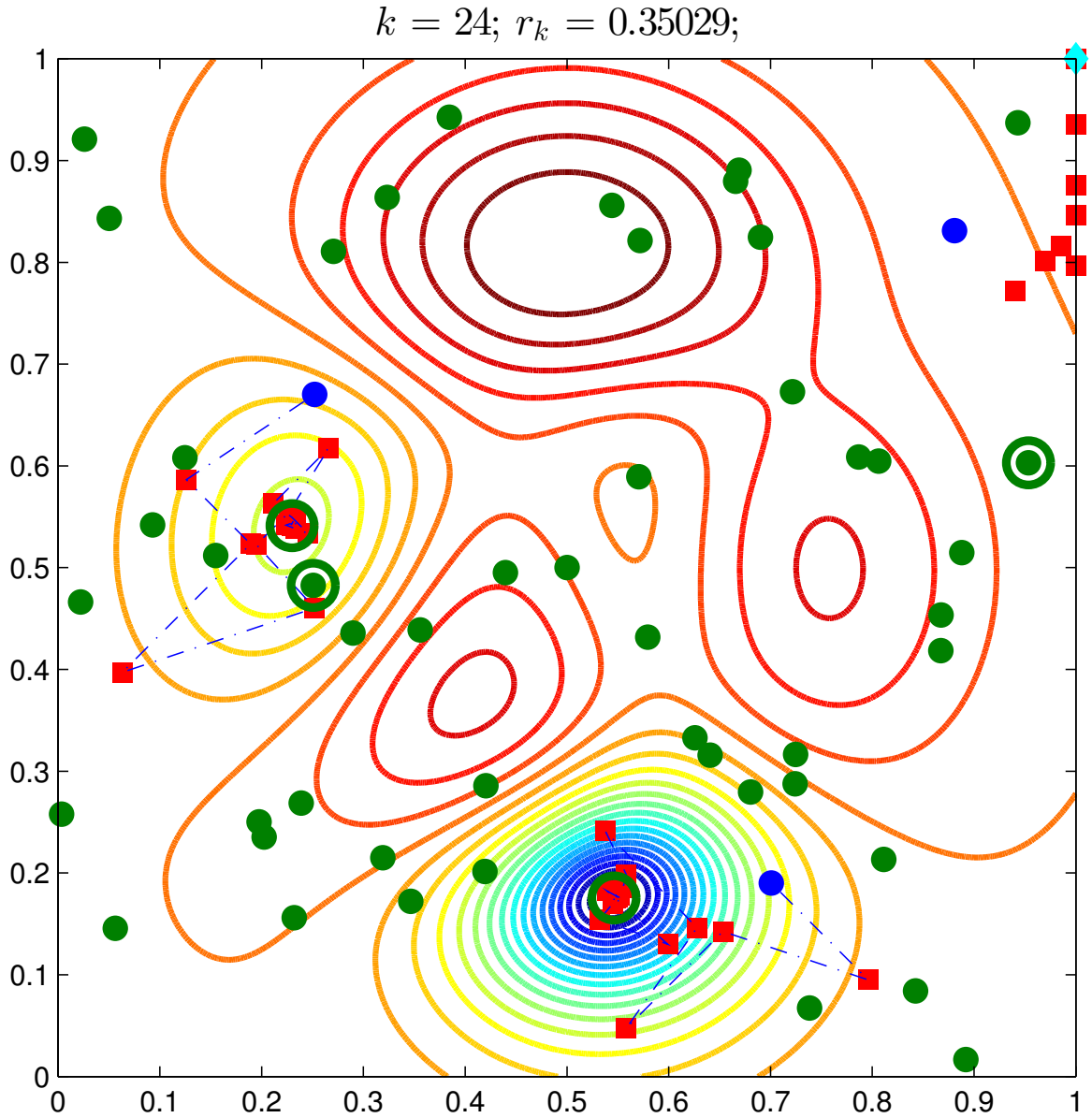
# Our Method



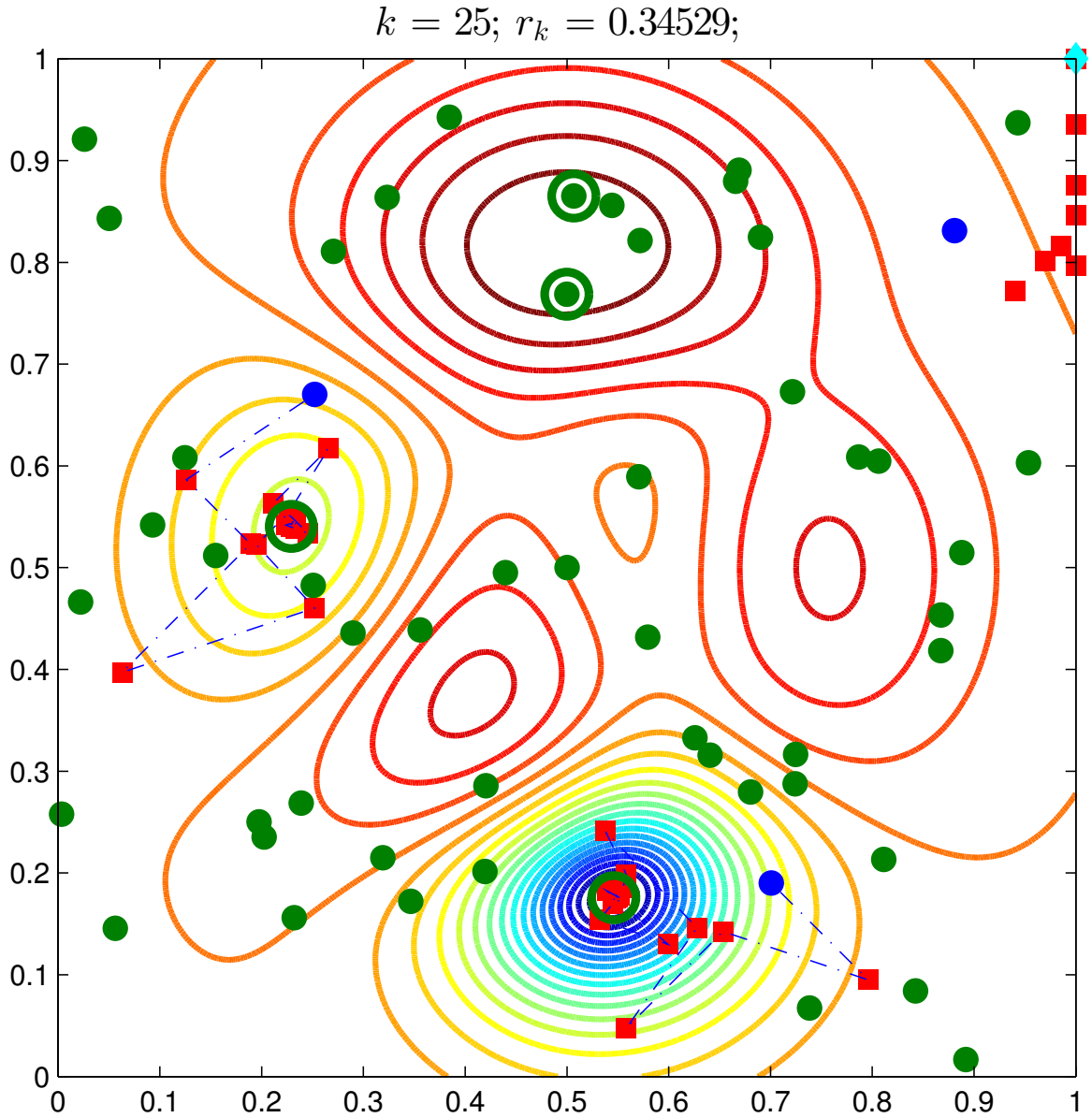
# Our Method



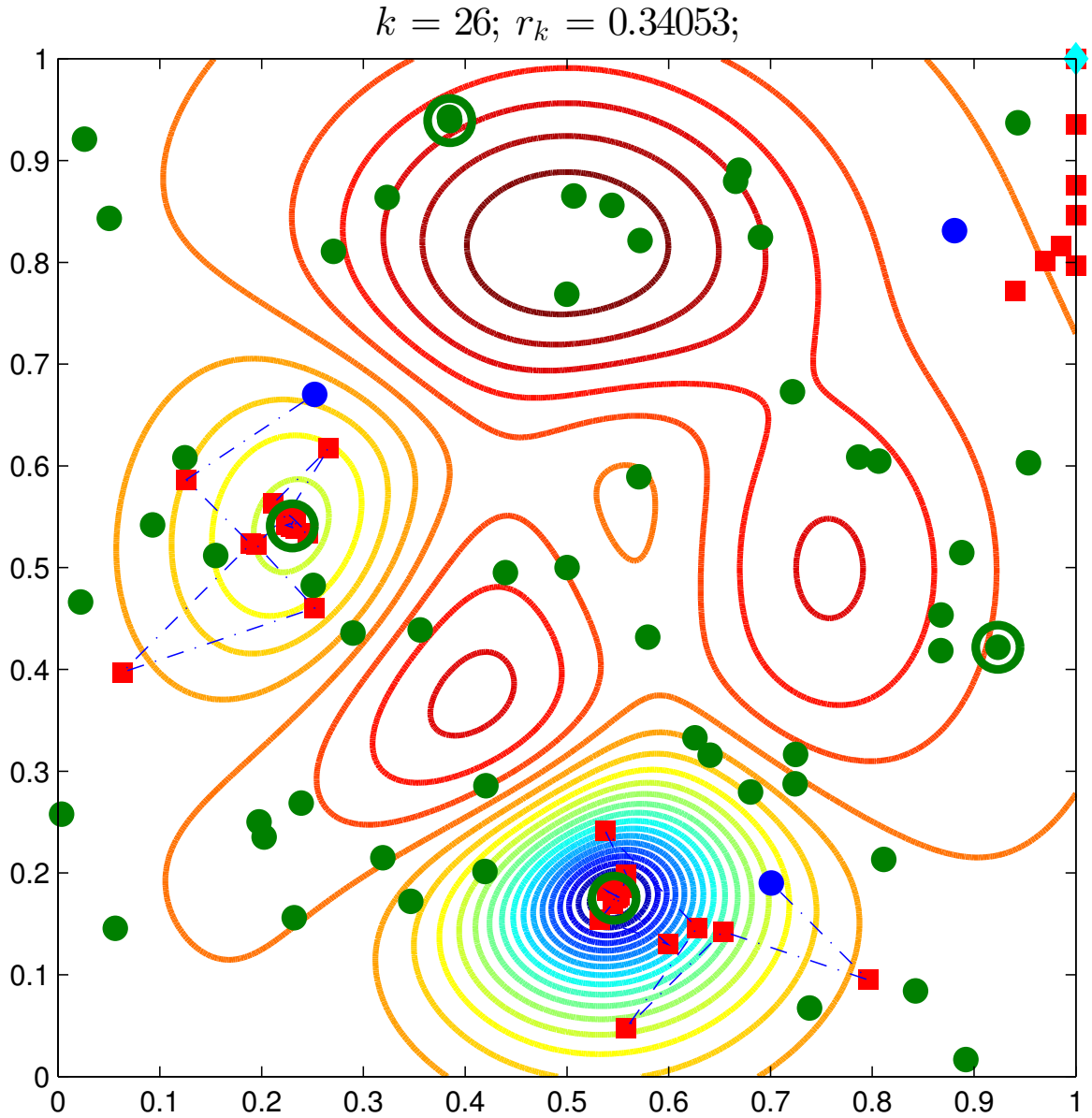
# Our Method



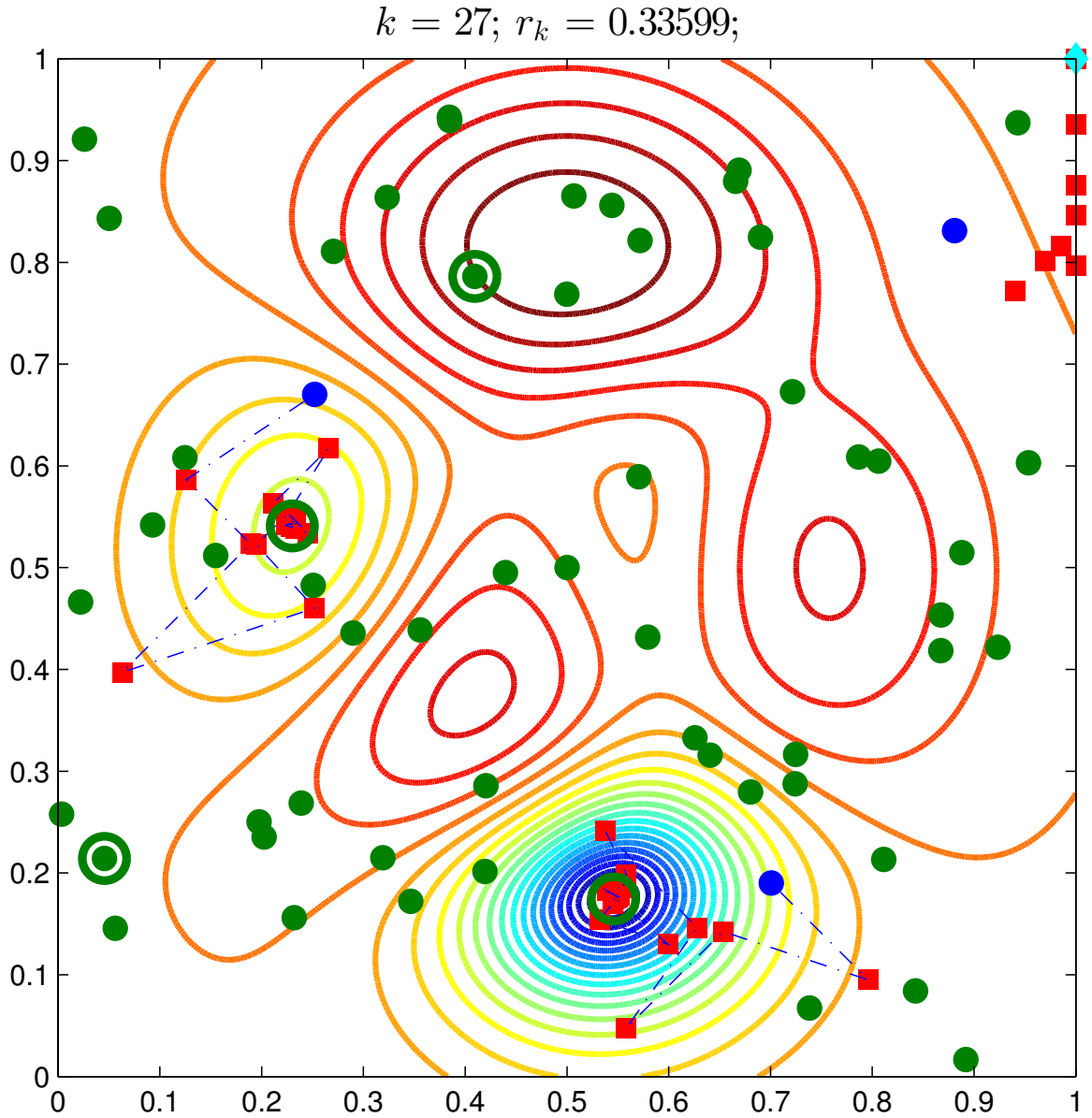
# Our Method



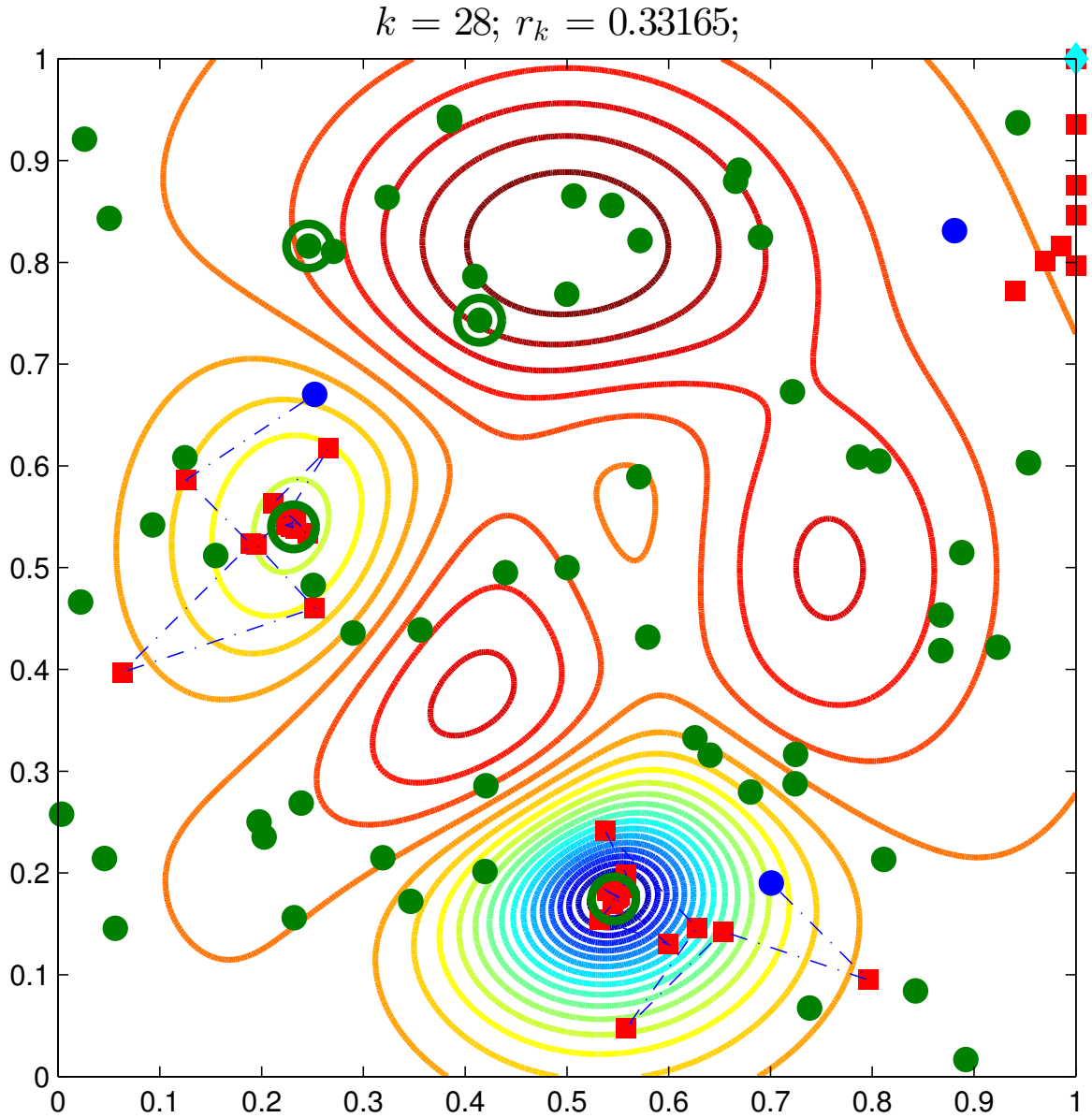
# Our Method



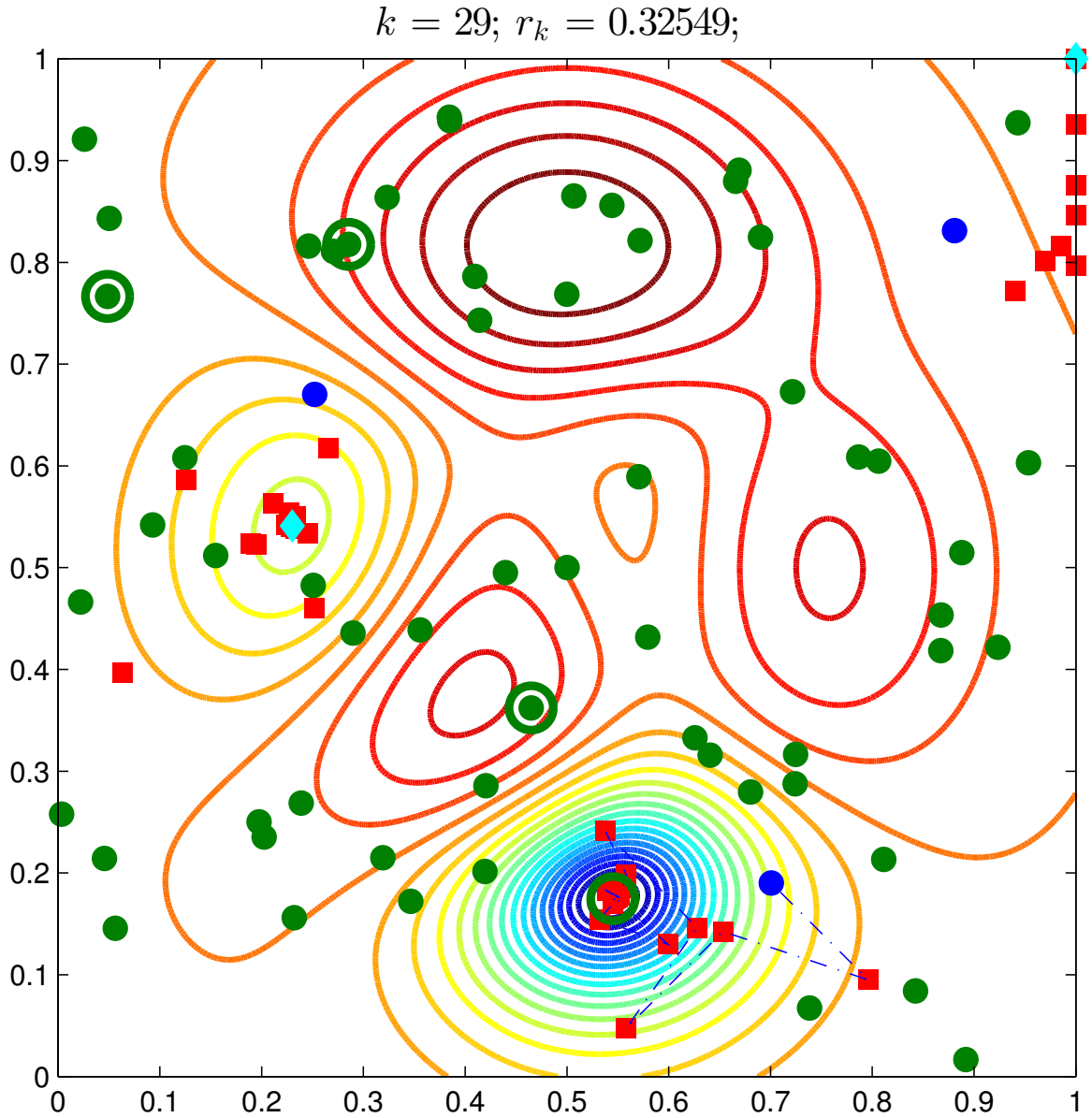
# Our Method



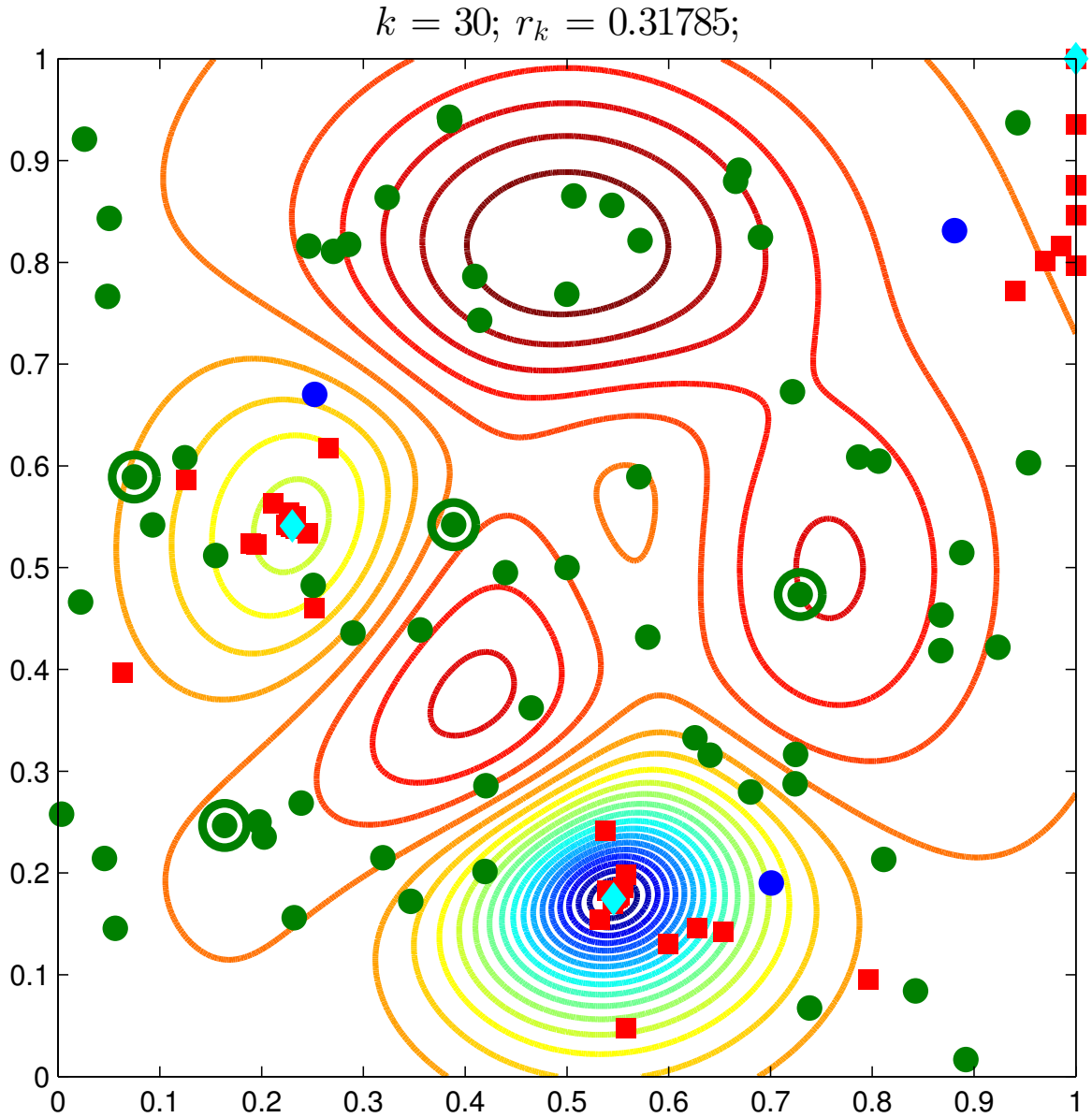
# Our Method



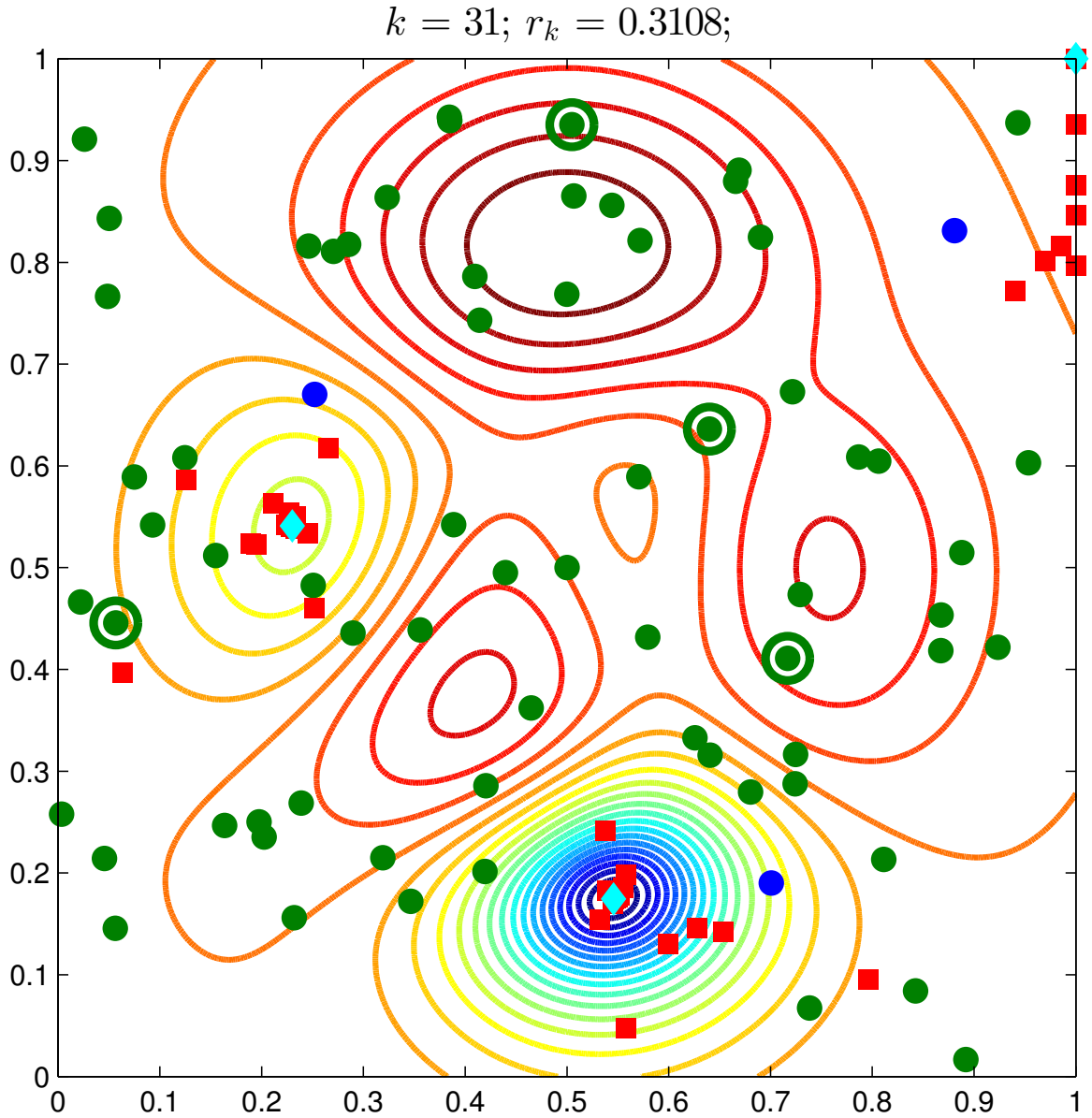
# Our Method



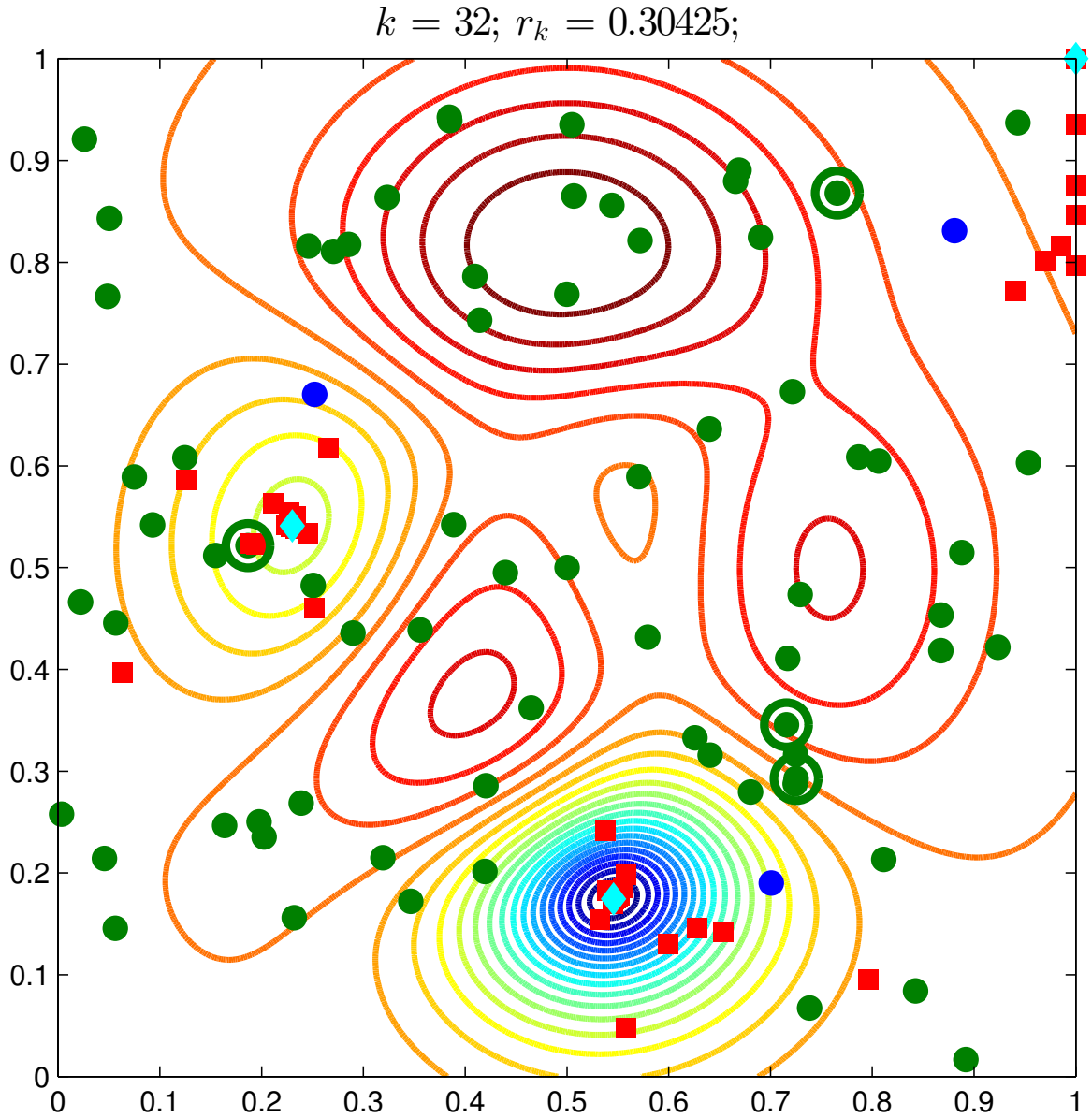
# Our Method



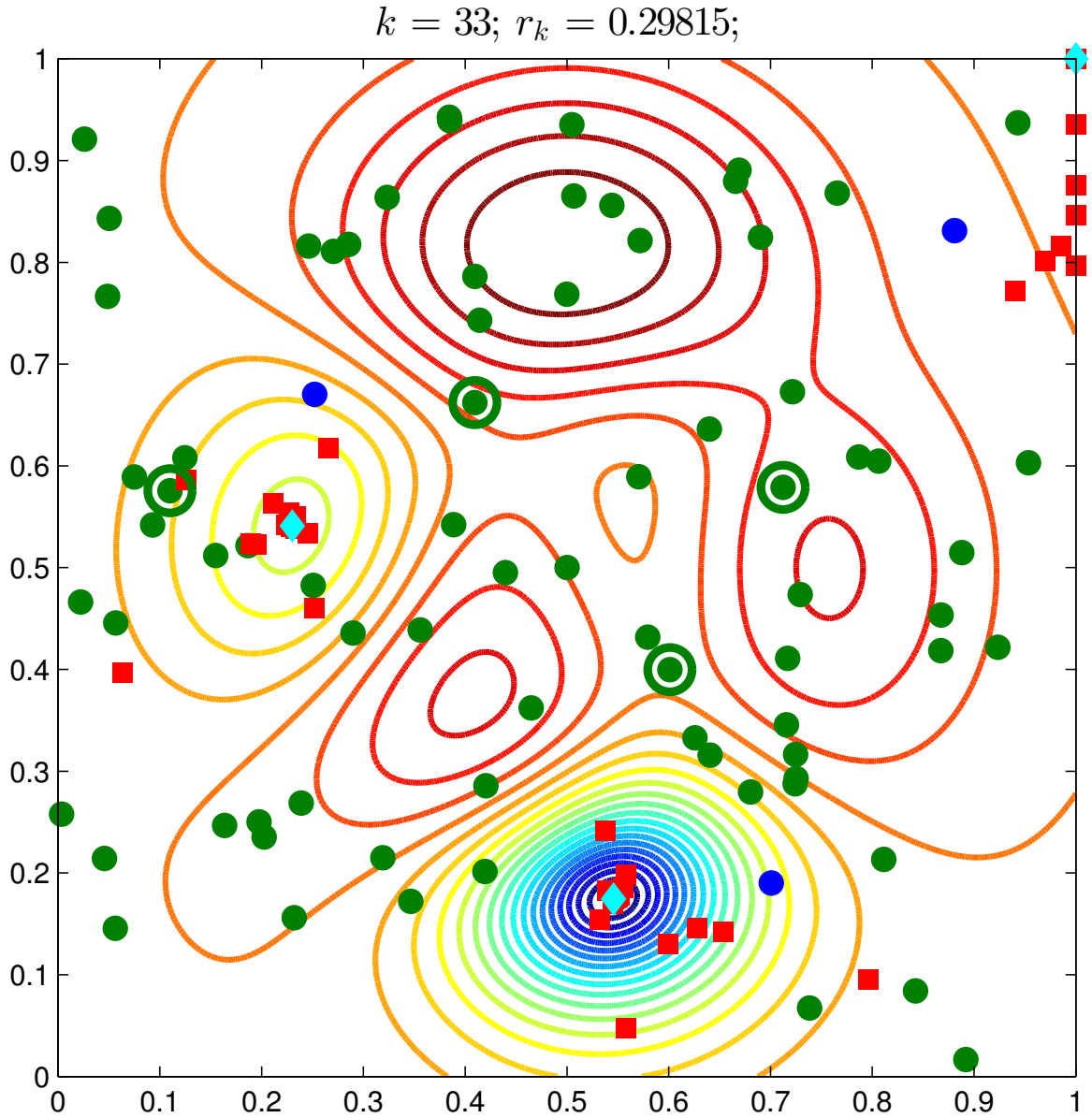
# Our Method



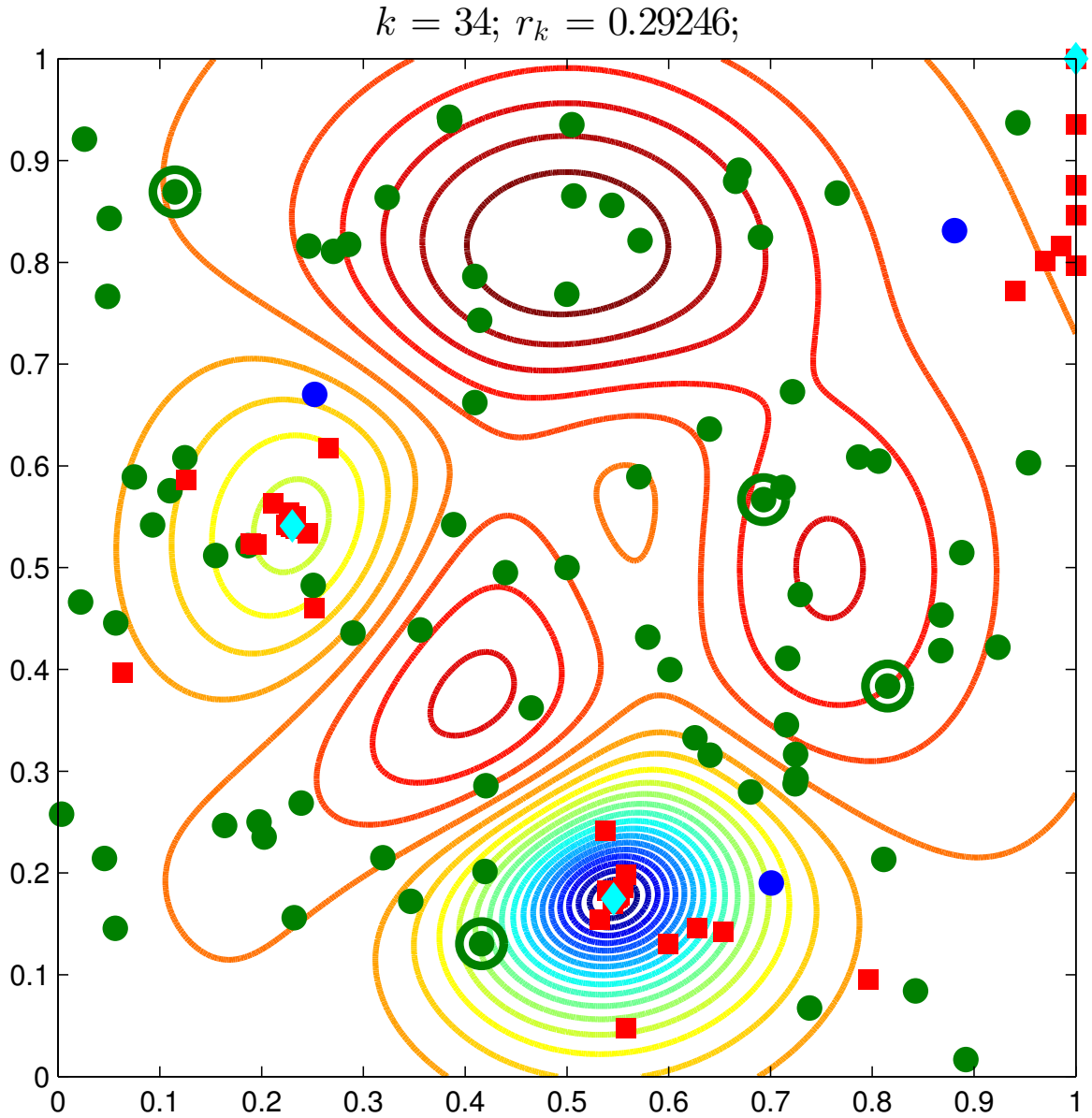
# Our Method



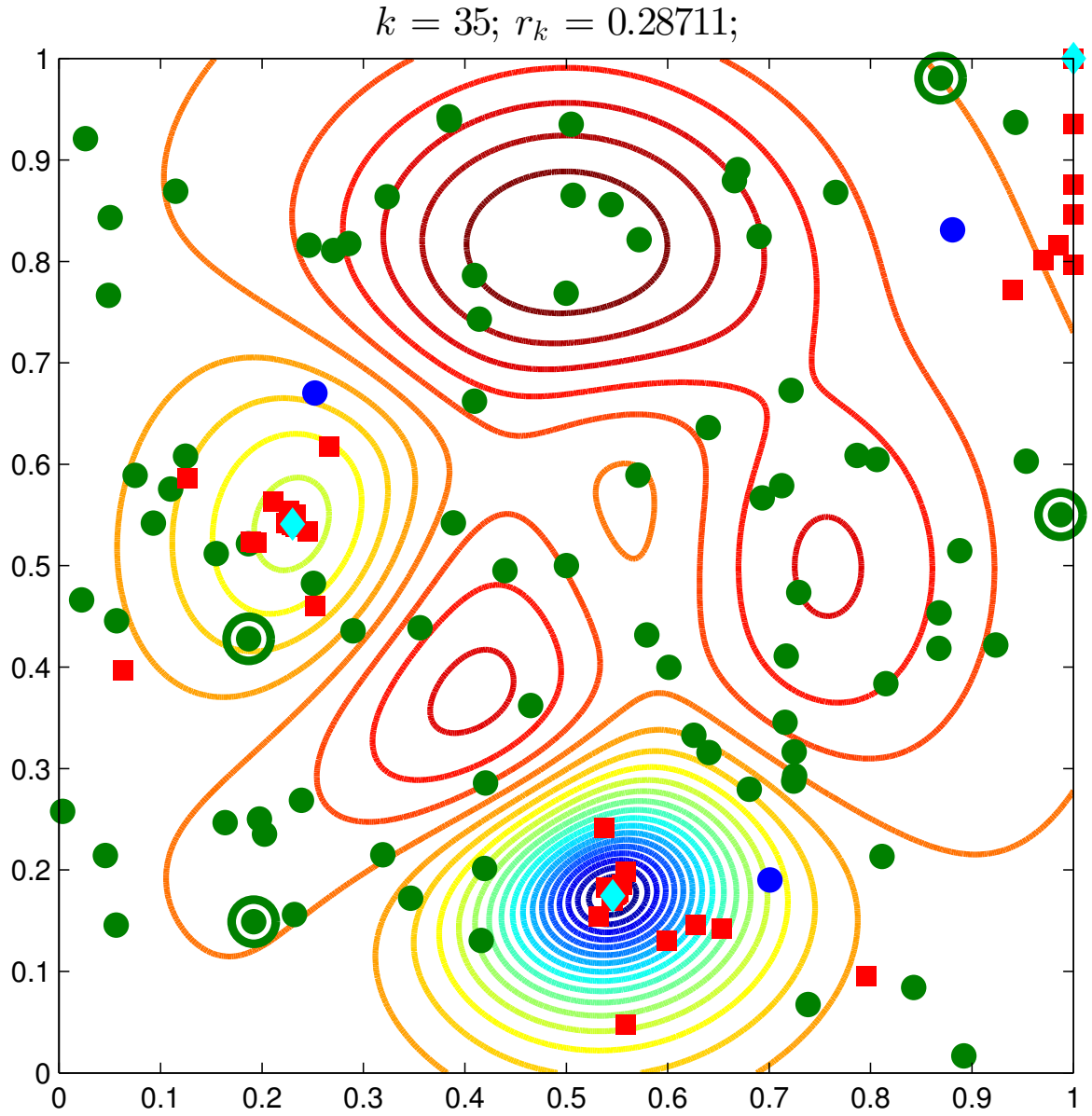
# Our Method



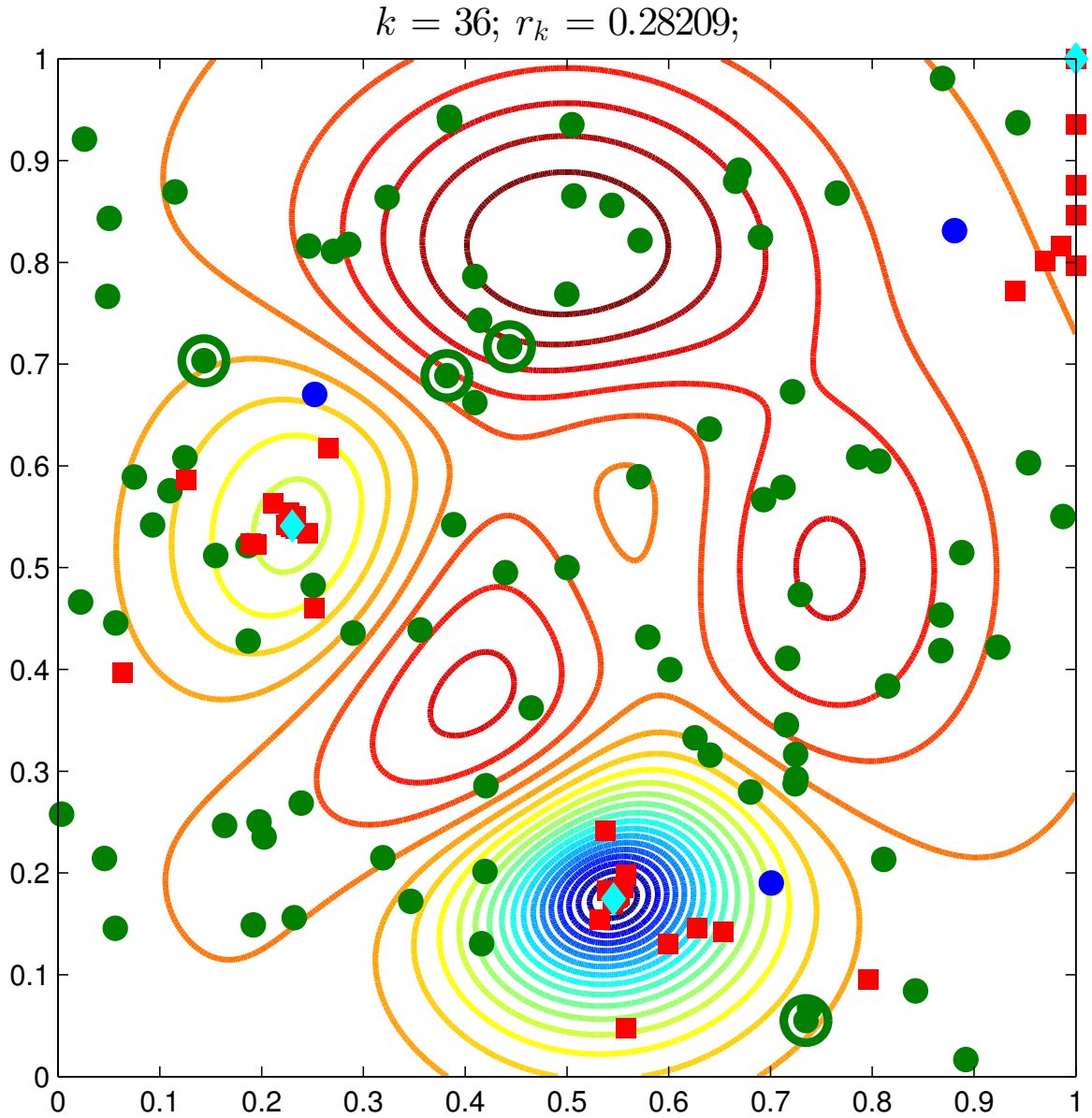
# Our Method



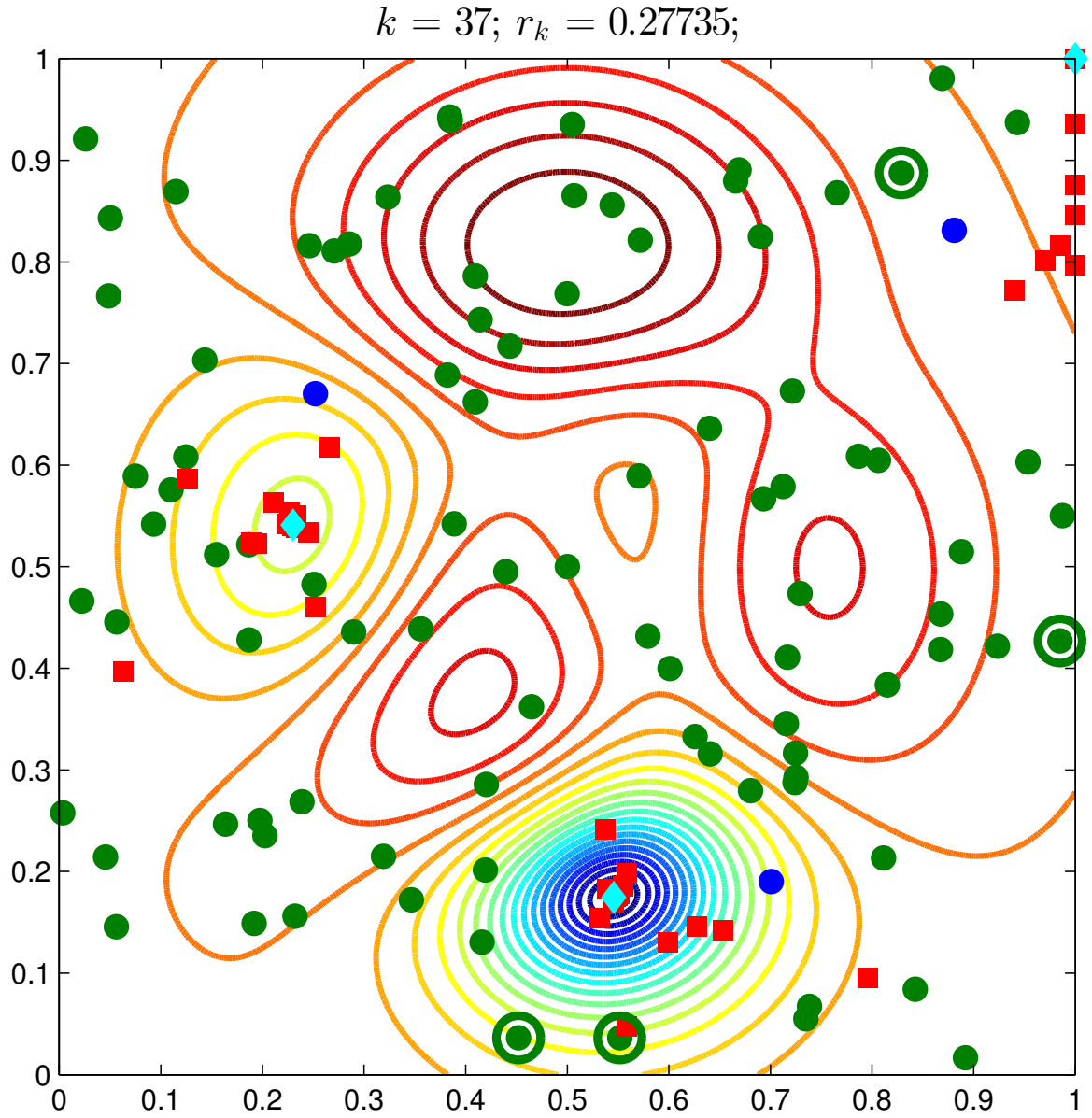
# Our Method



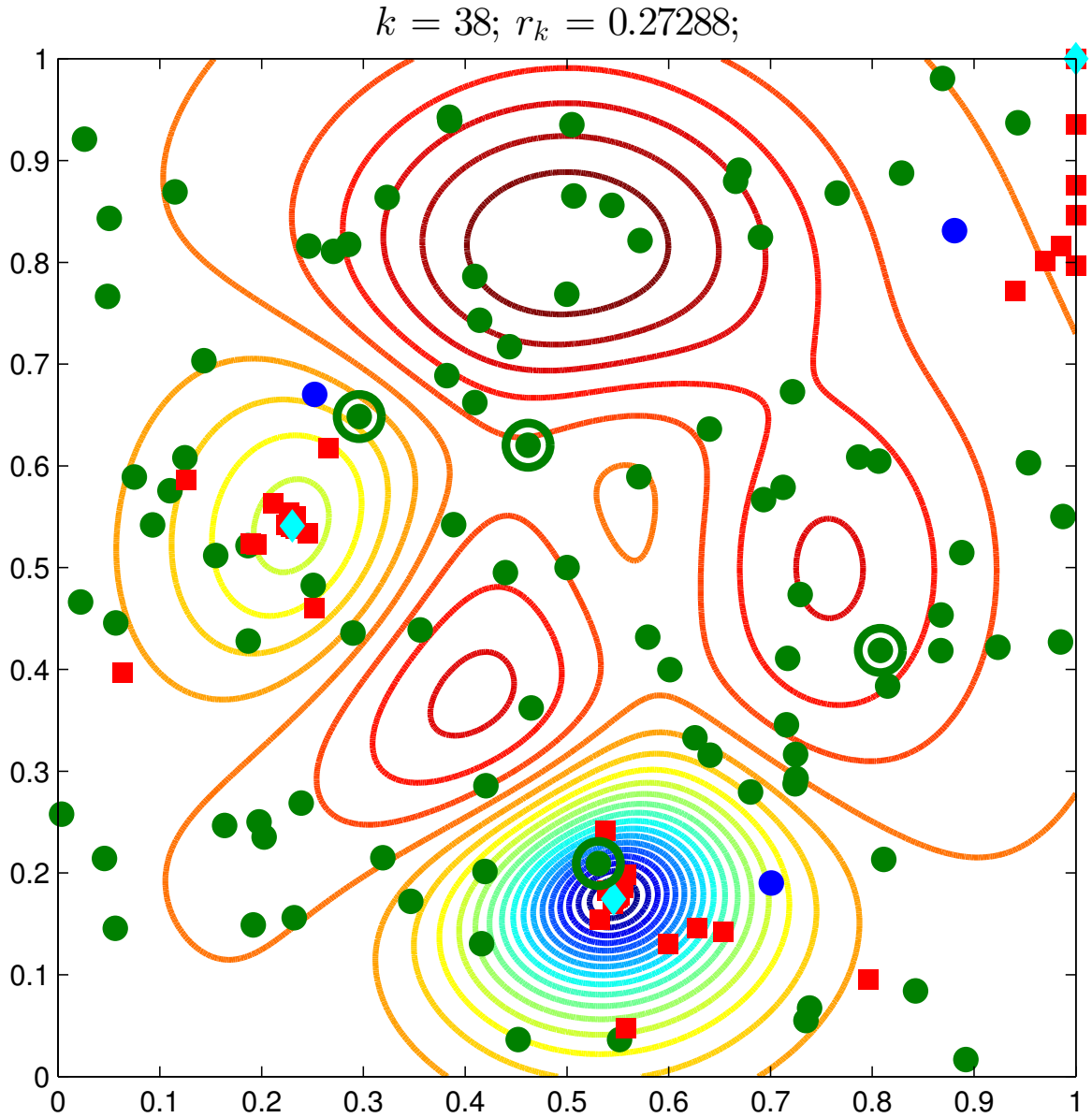
# Our Method



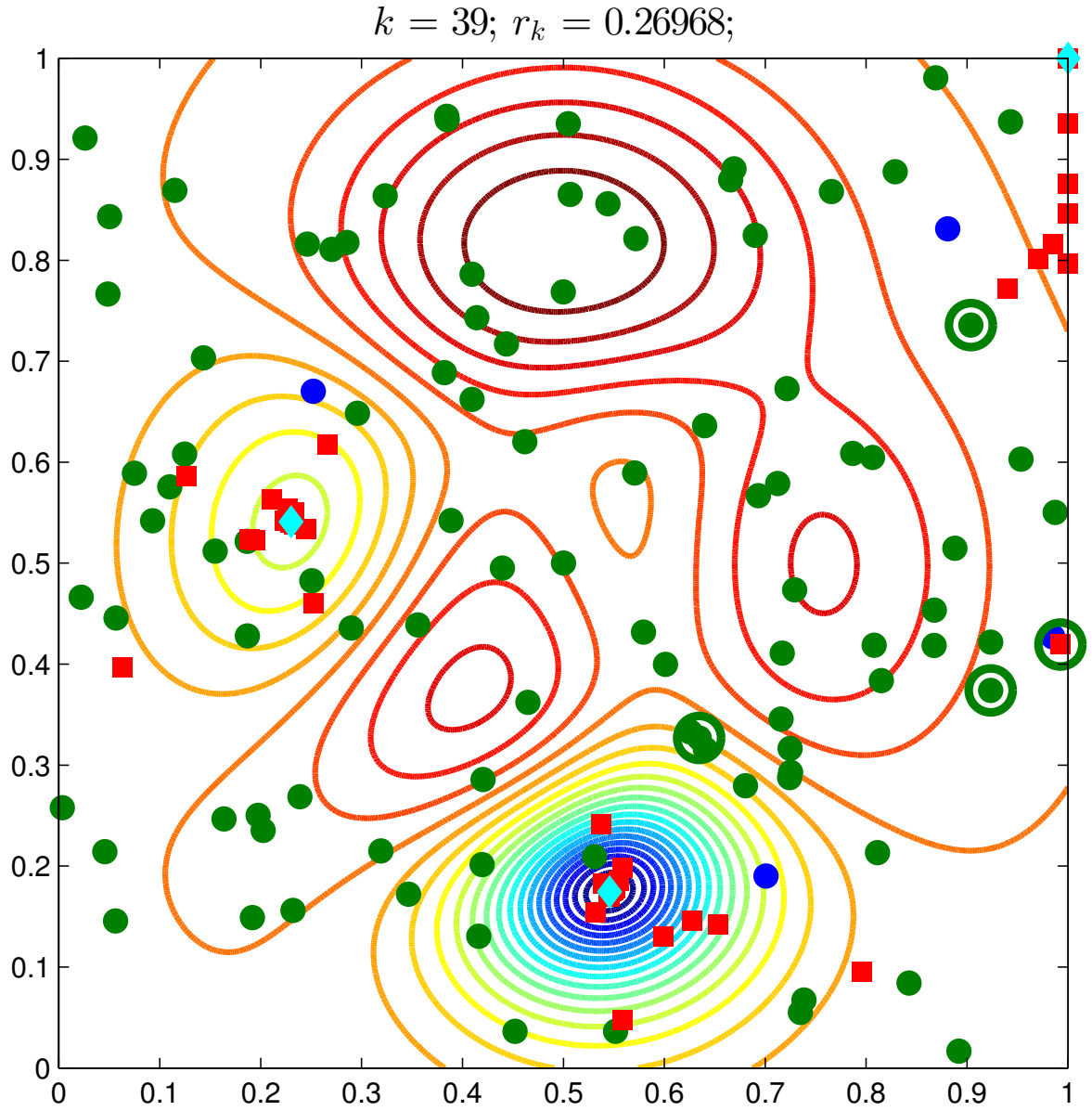
# Our Method



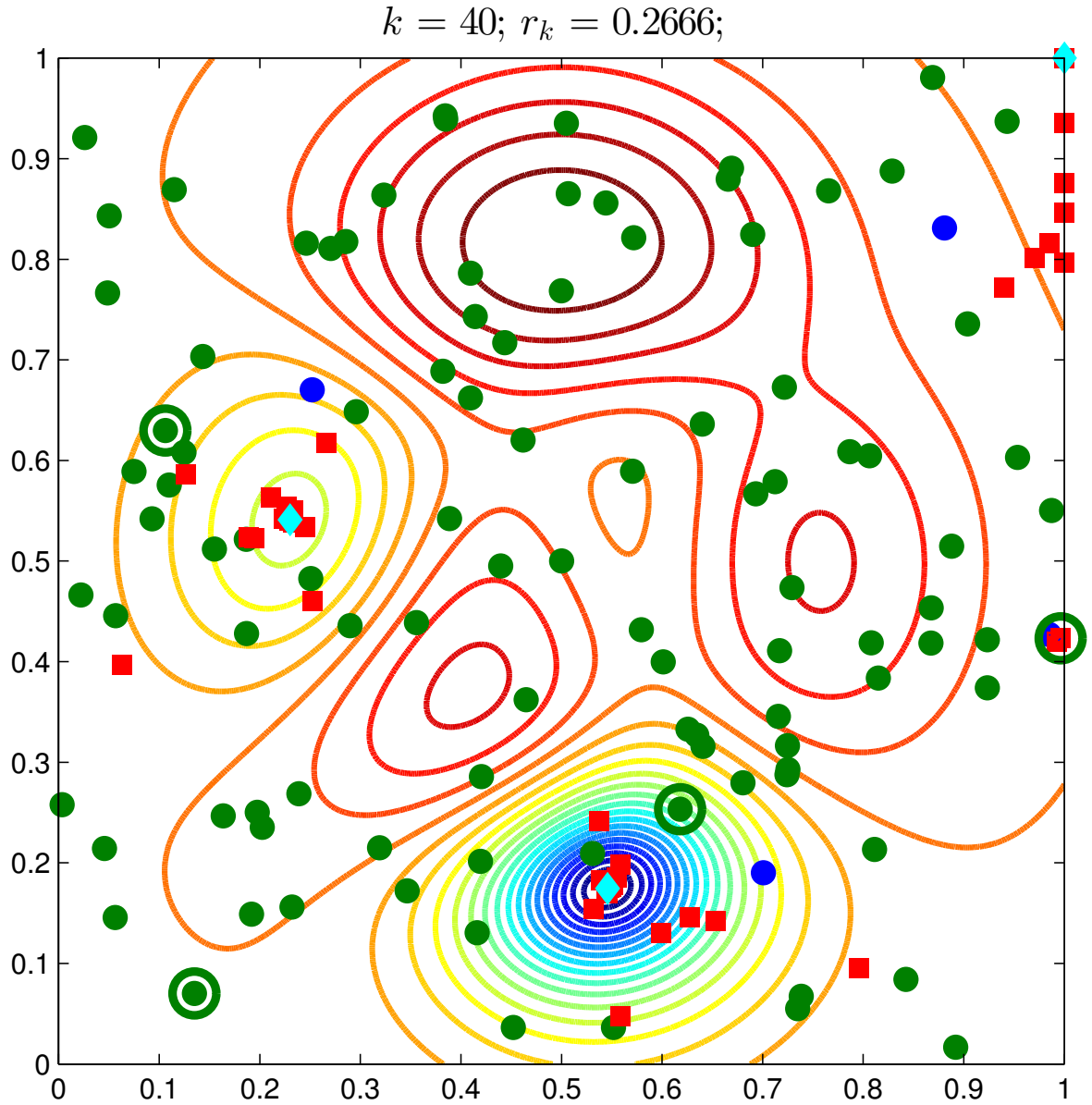
# Our Method



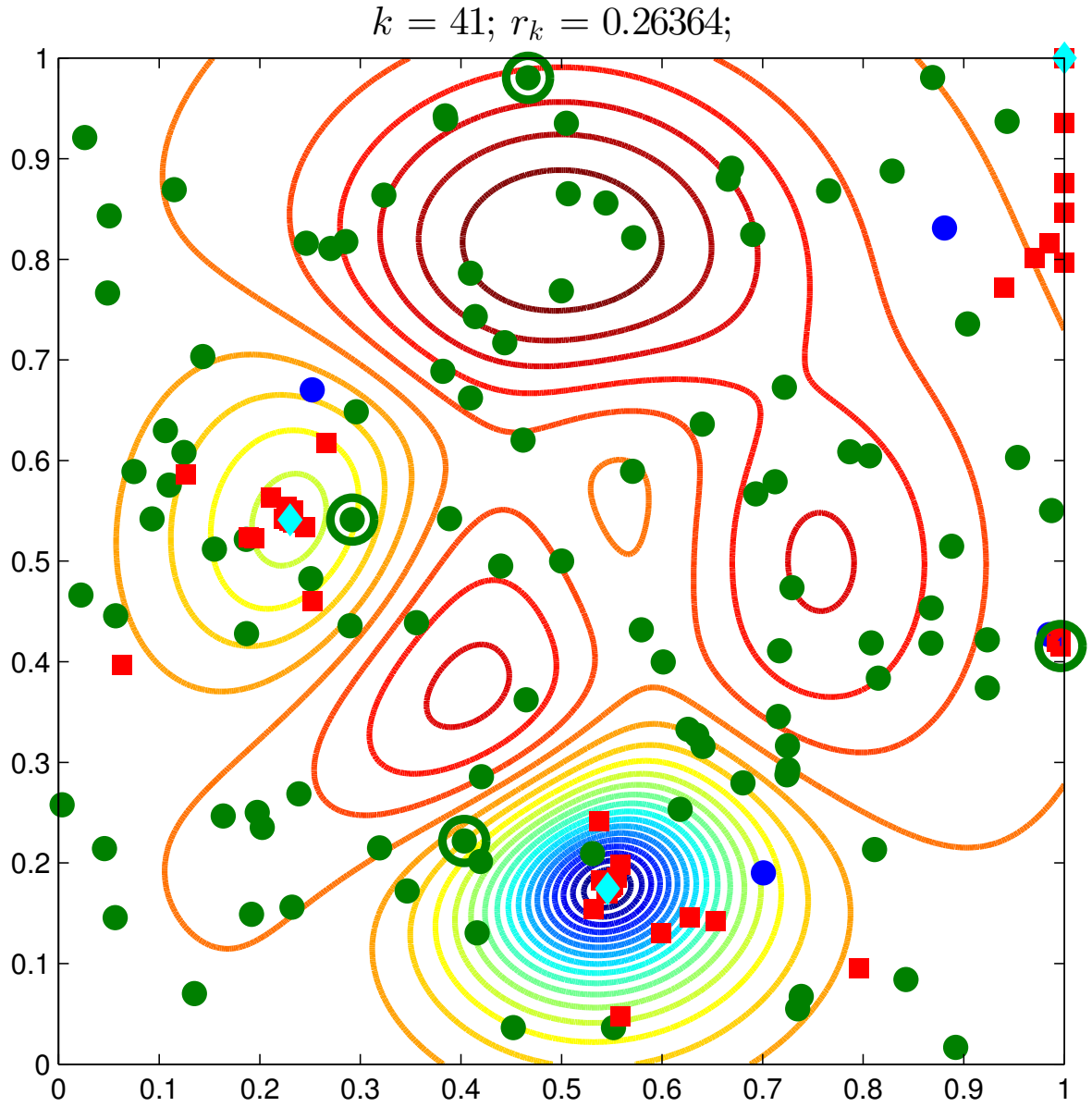
# Our Method



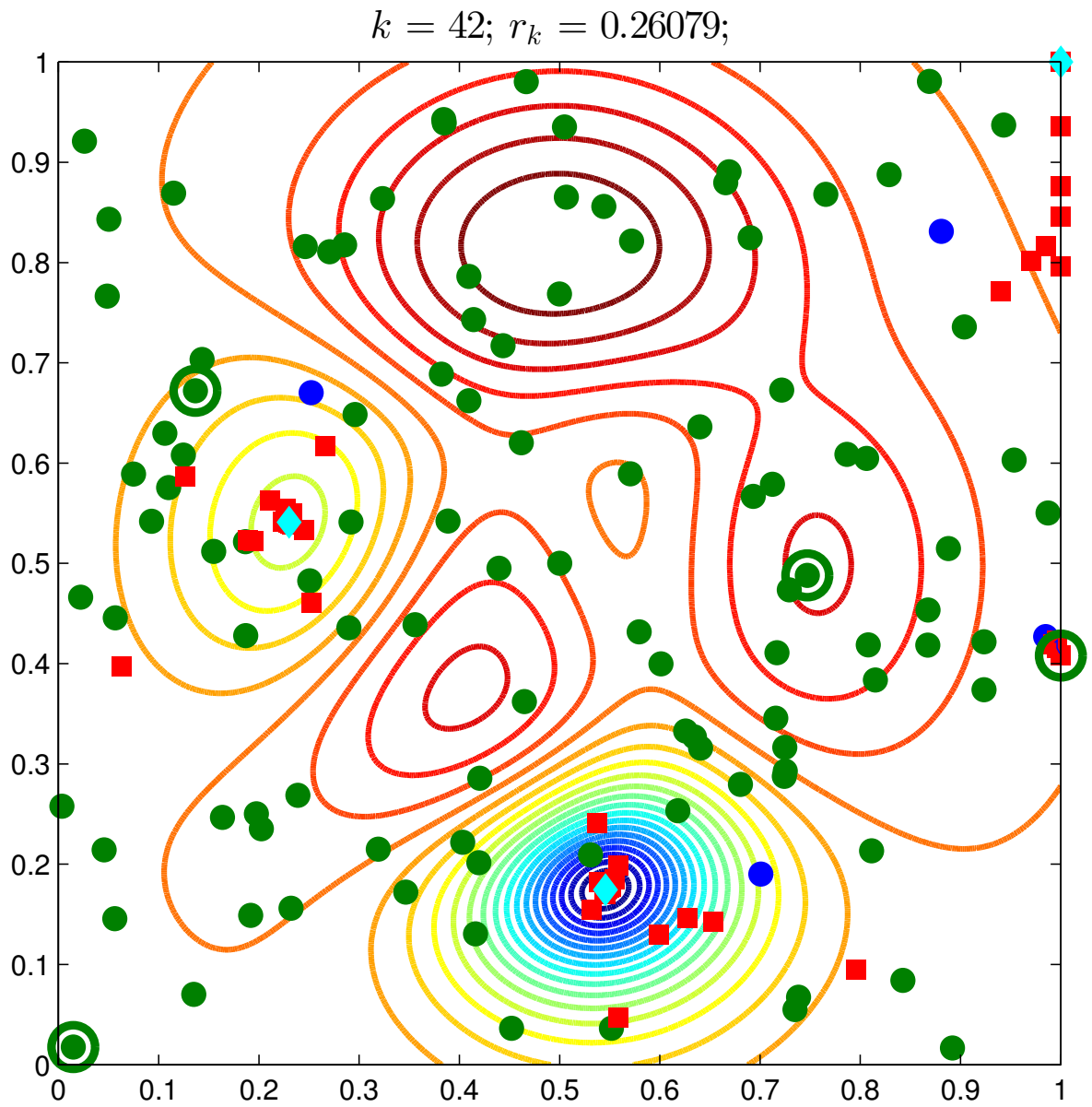
# Our Method



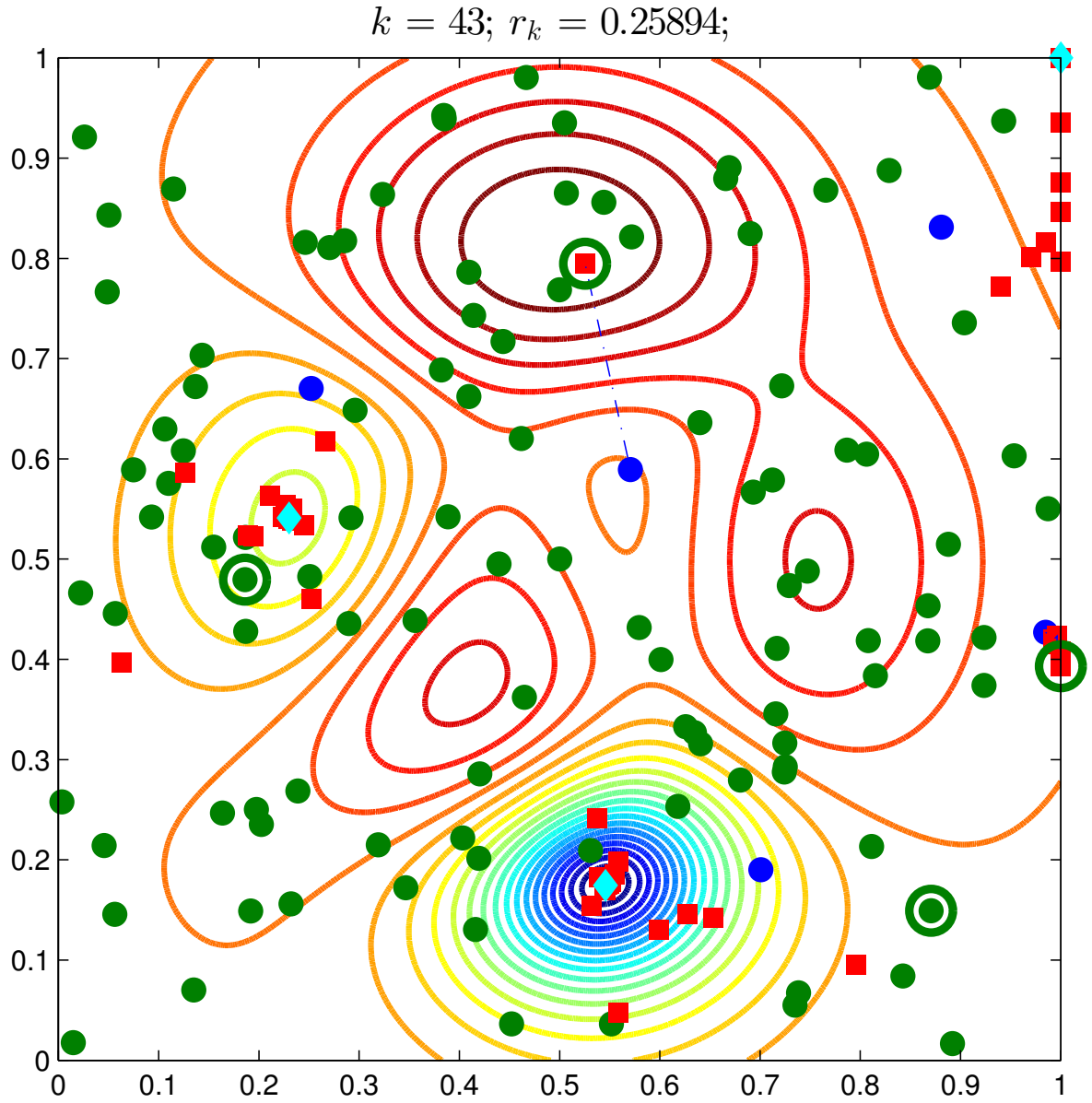
# Our Method



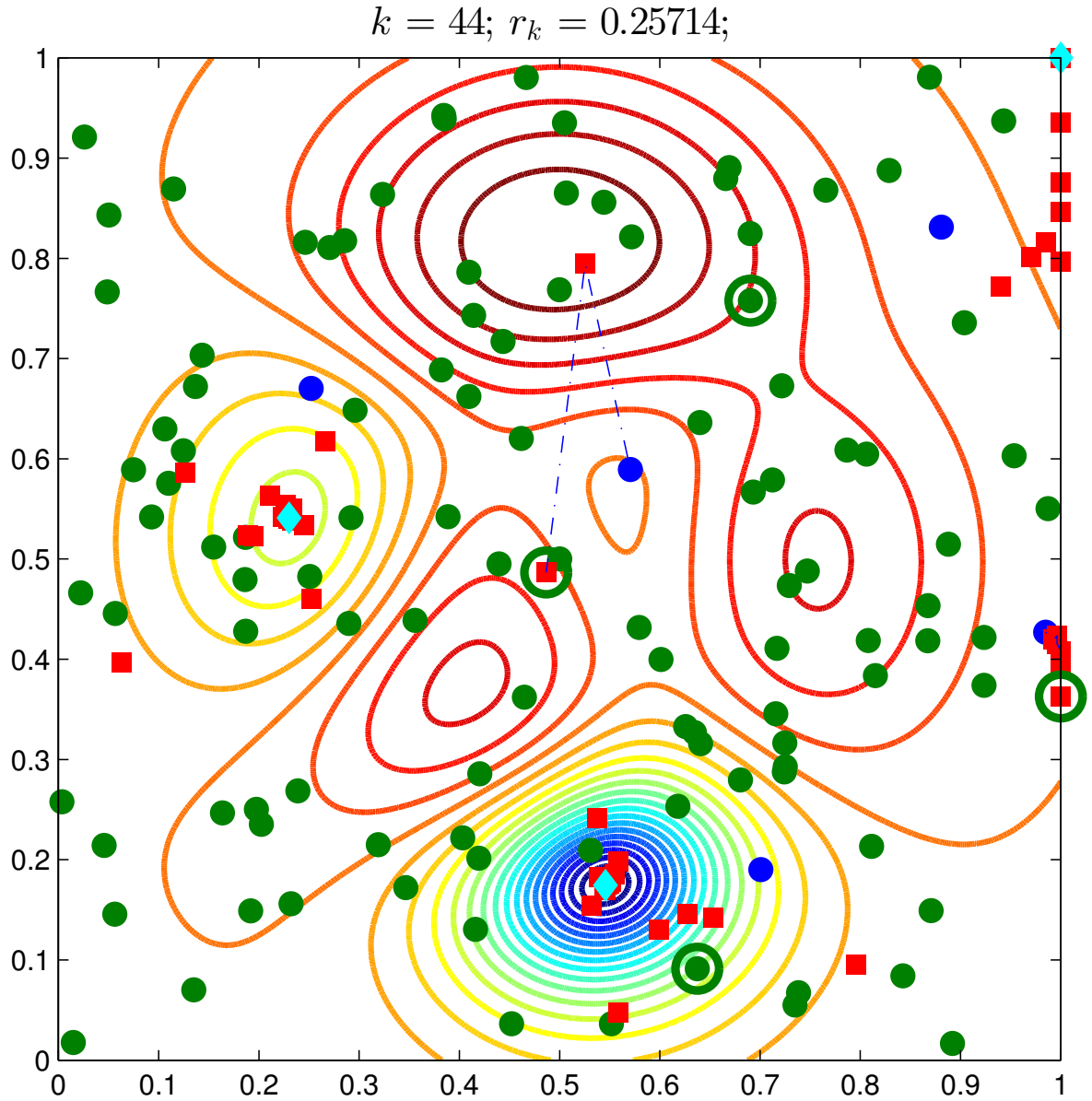
# Our Method



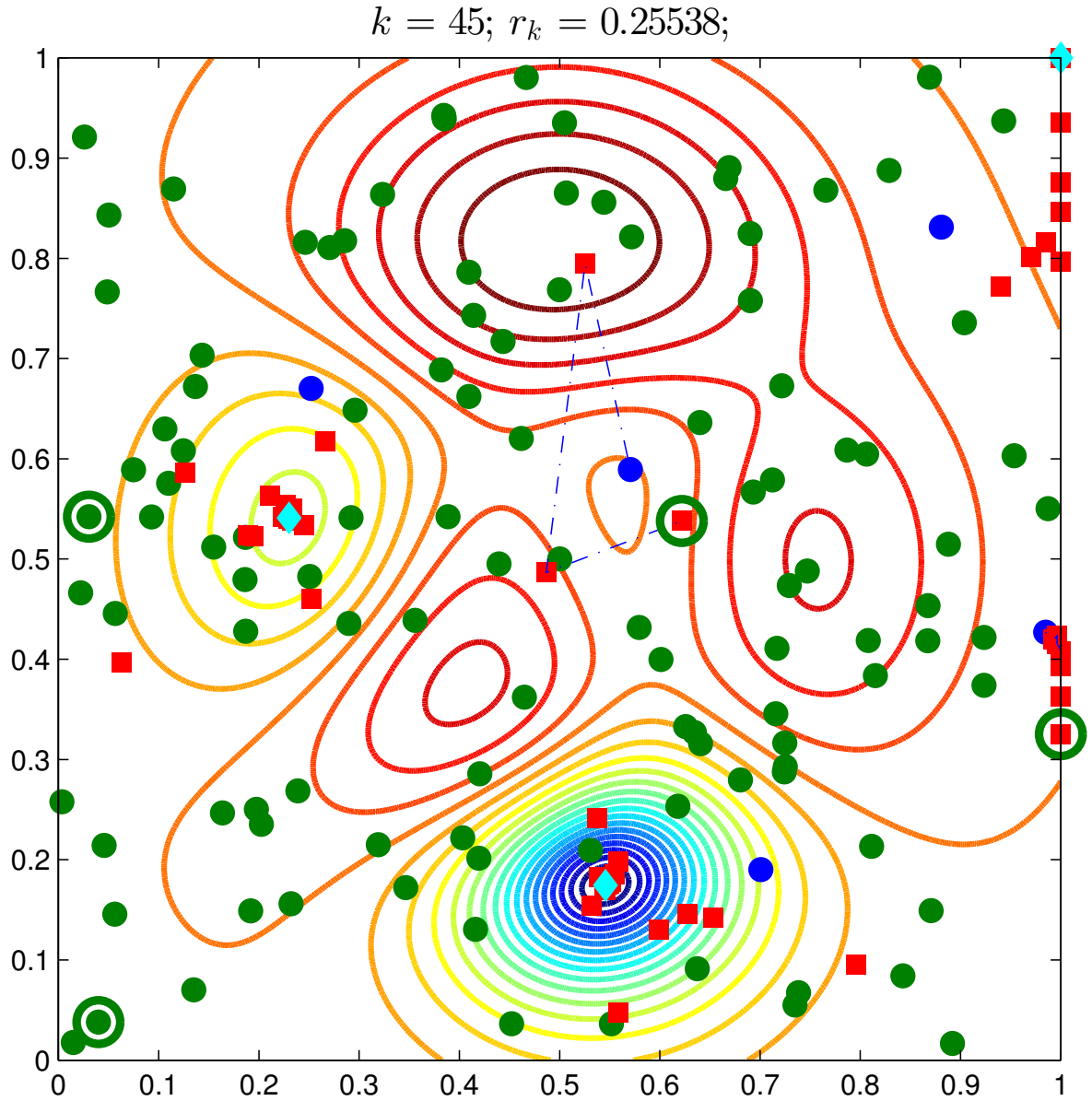
# Our Method



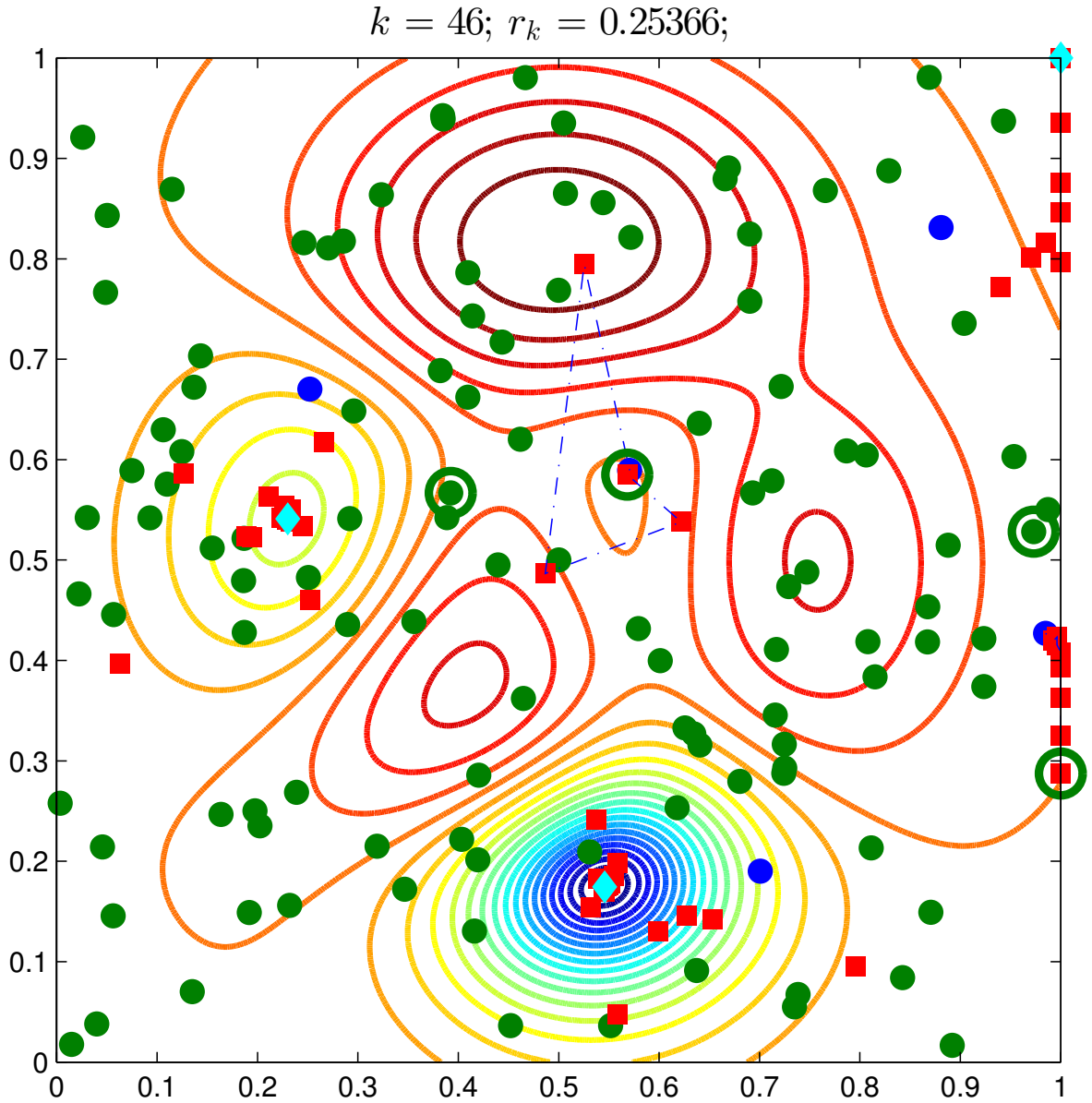
# Our Method



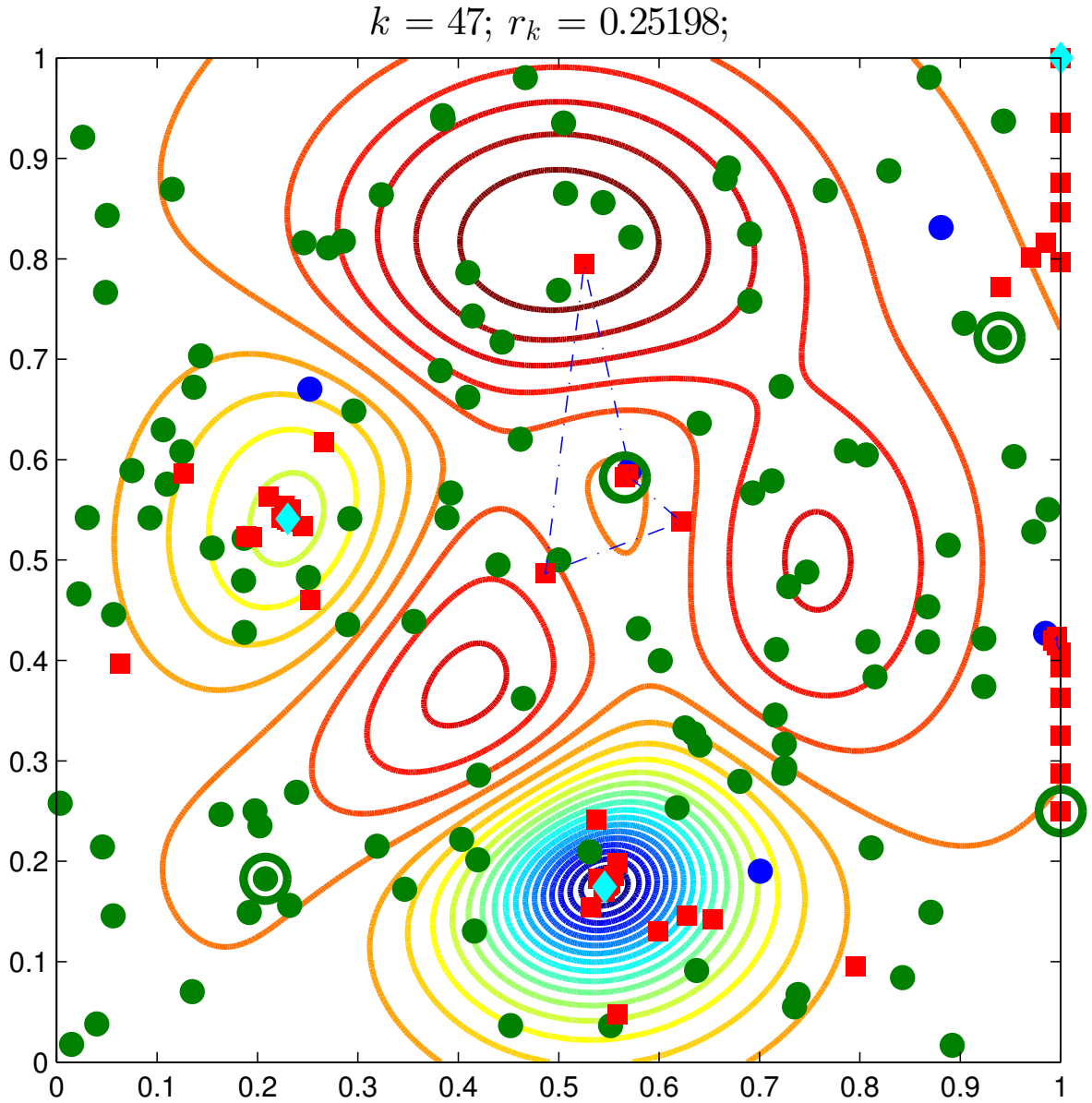
# Our Method



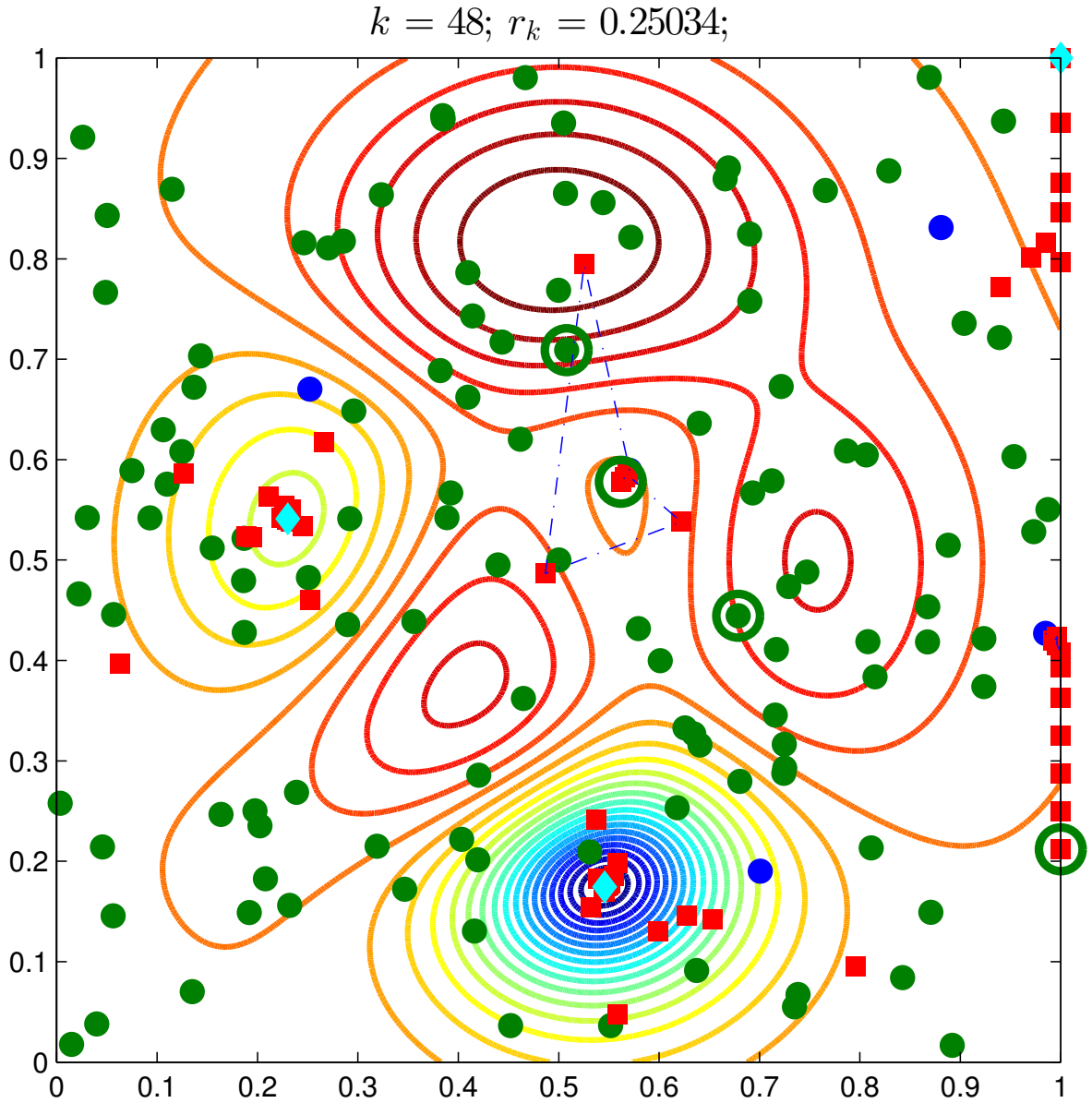
# Our Method



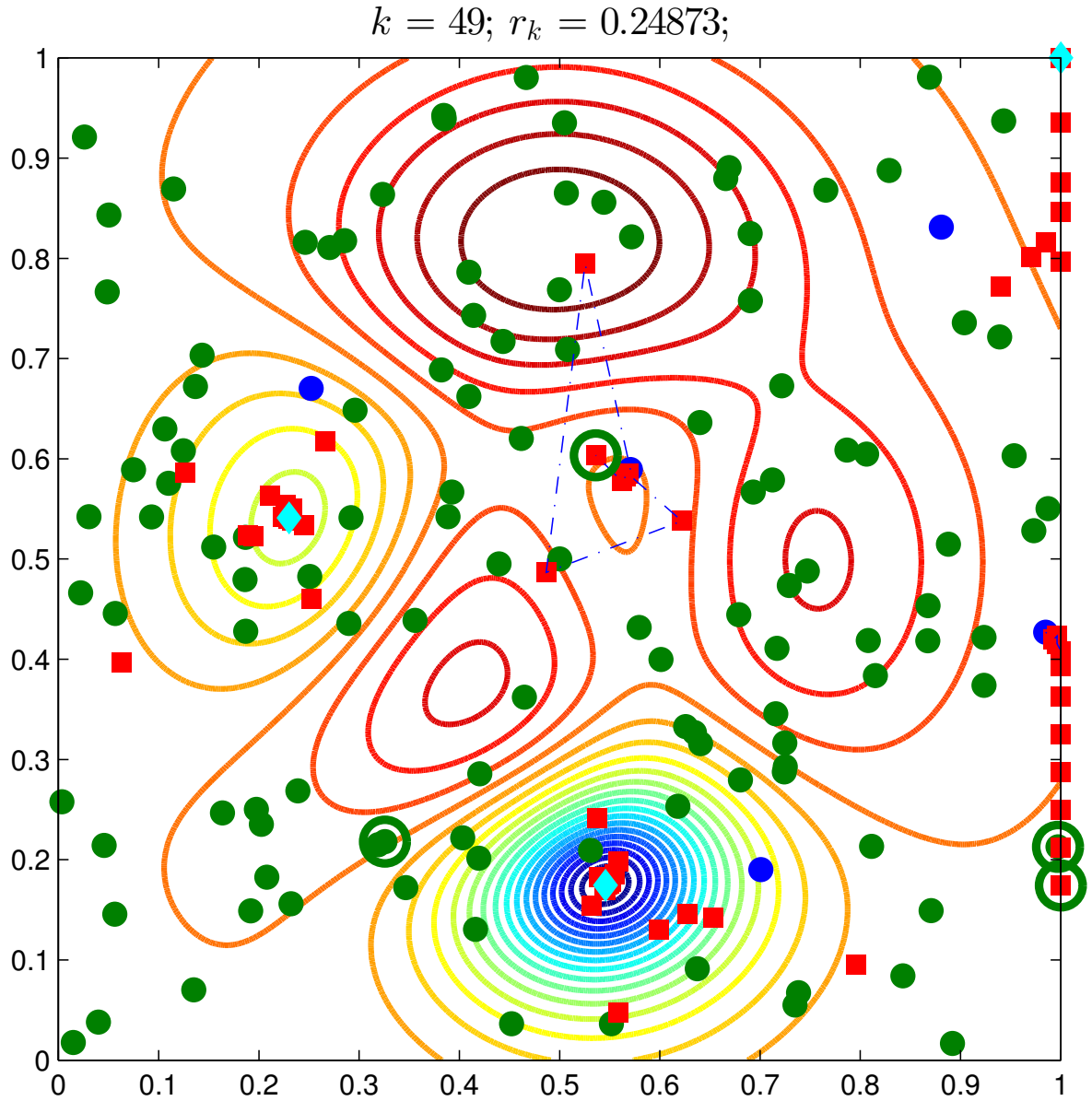
# Our Method



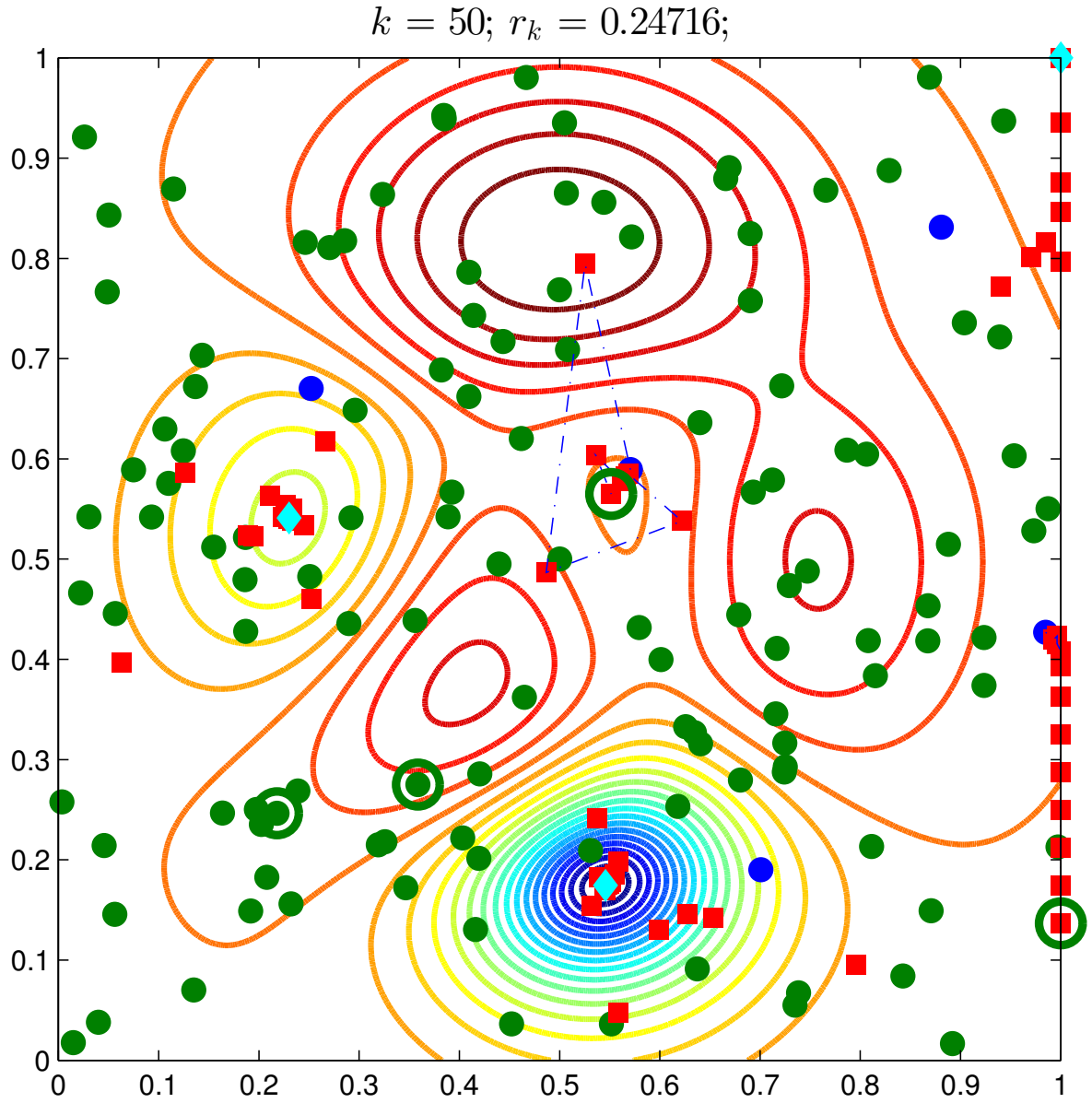
# Our Method



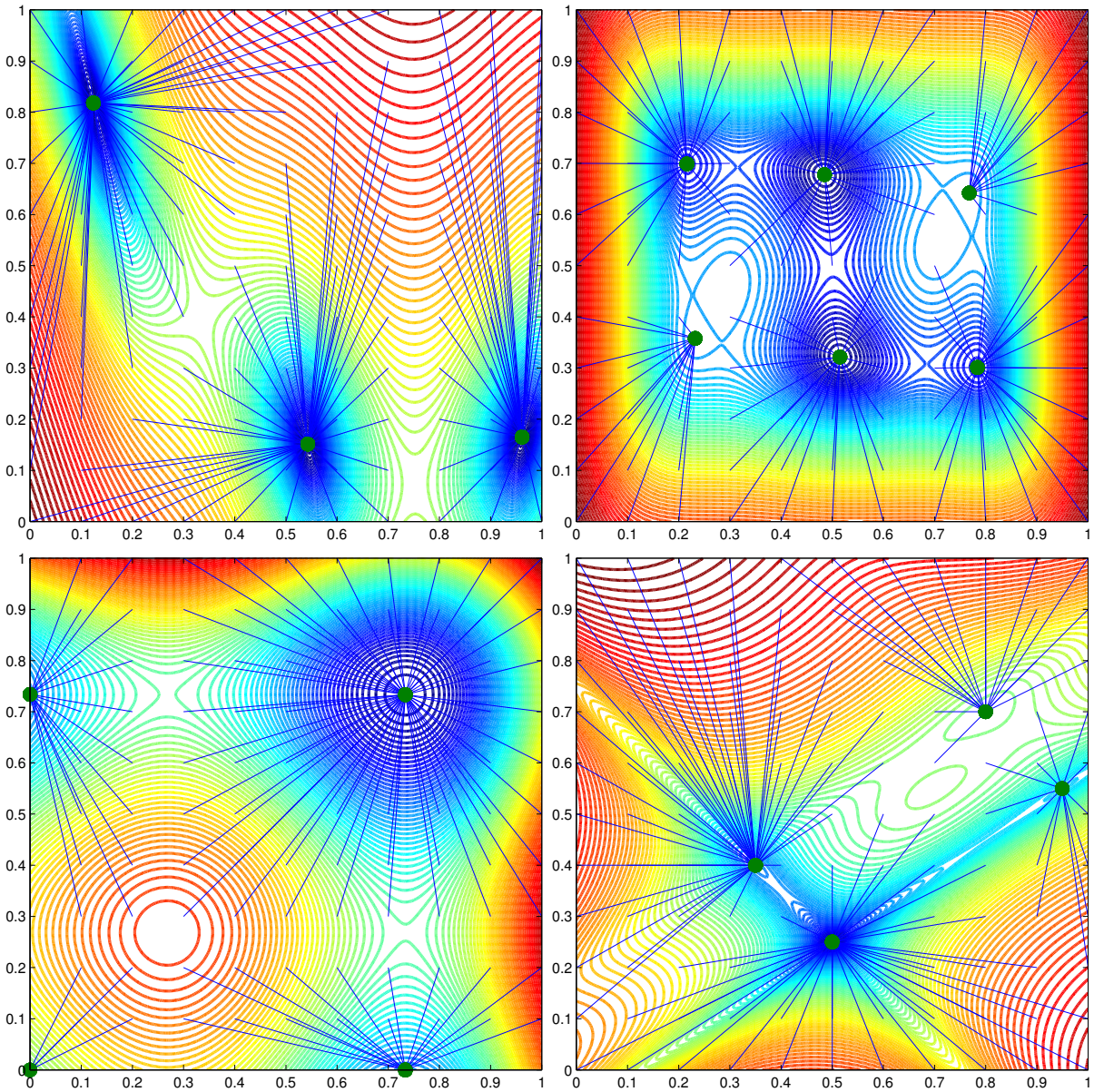
# Our Method



# Our Method



# Measuring Performance




ORBIT started from points in a 10x10 grid

# Measuring Performance

Algorithms compared

- Ours (4 parallel workers)
- Direct (serial)
- pVTDirect (4 parallel workers)

- Since Ours involves a random sampling stream, each problem was dupilcated 10 times.
- Each method evaluates the centroid first.

# Measuring Performance

**Define**

Let $\mathcal{X}^*$ be the set of all local minima of $f$.
Let $f^*_{(i)}$ be the $i$th smallest value $\{f(x^*)|x^* \in \mathcal{X}^*\}$.
Let $x^*_{(i)}$ be the element of $\mathcal{X}^*$ corresponding to the value $f^*_{(i)}$.

A method has found the global minimum within a level $\tau > 0$ at batch iteration $k$ if it has found a point $\hat{x}$ satisfying:

$$f(\hat{x}) - f^*_{(1)} \leq (1 - \tau)\left(f(x_0) - f^*_{(1)}\right),$$

where $x_0$ is the starting point for problem $p$.

# Measuring Performance

**Define**

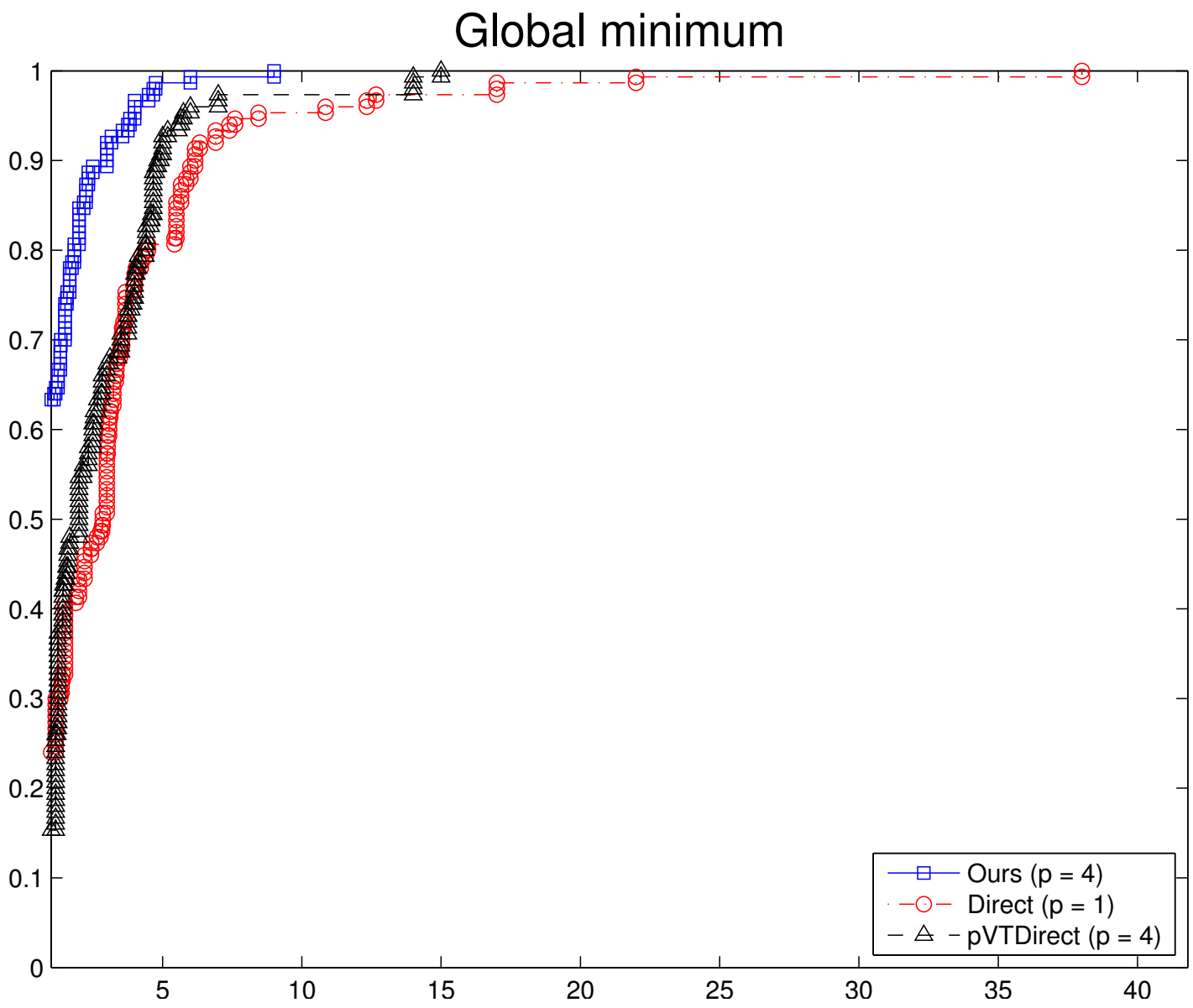
Let $\mathcal{X}^*$ be the set of all local minima of $f$.
Let $f^*_{(i)}$ be the $i$th smallest value $\{f(x^*)|x^* \in \mathcal{X}^*\}$.
Let $x^*_{(i)}$ be the element of $\mathcal{X}^*$ corresponding to the value $f^*_{(i)}$.

A method has found the $j$ best local minima within a level $\tau > 0$ at batch iteration $k$ if:

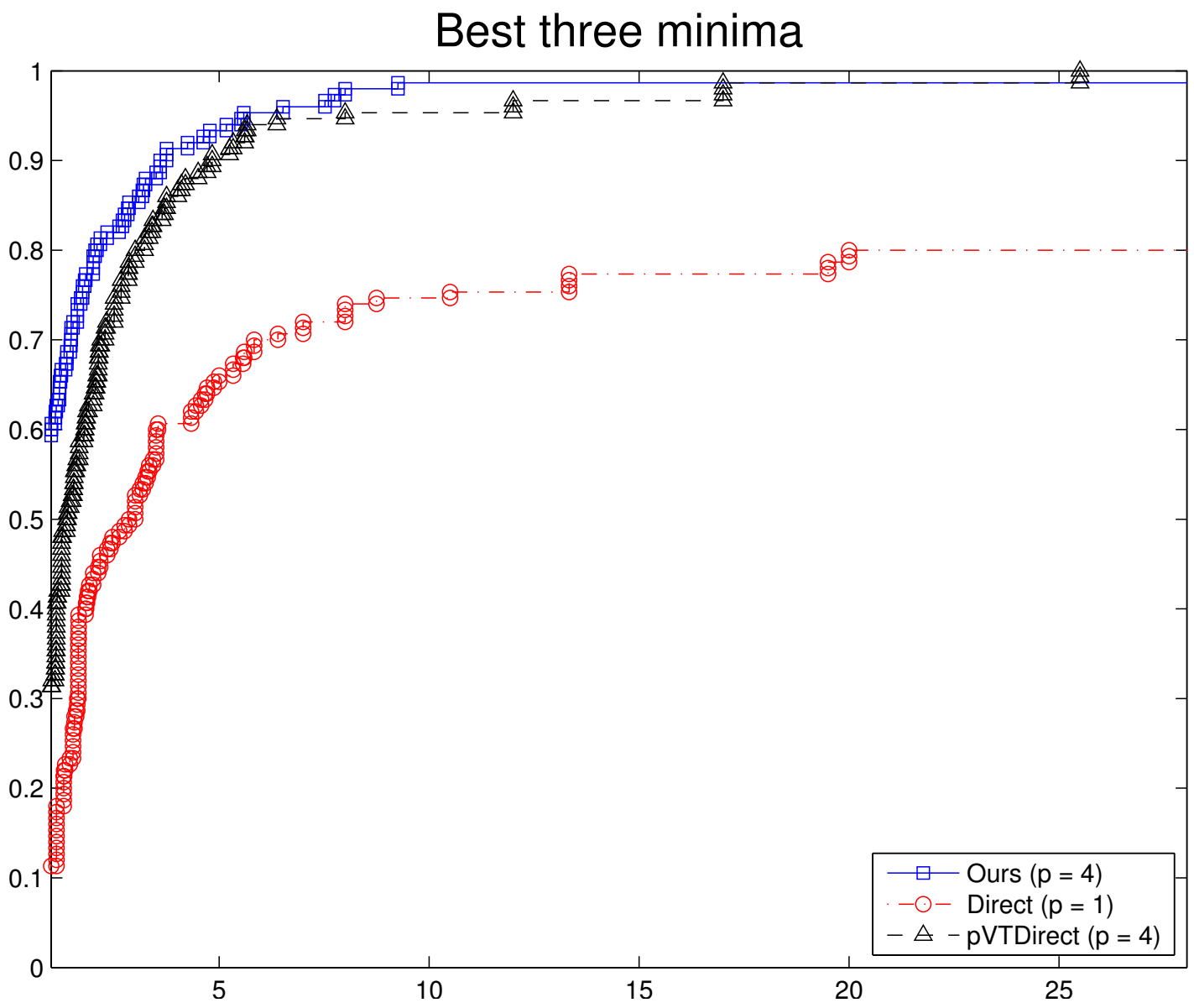
$$\left|\left\{x^*_{(1)}, \ldots, x^*_{(\bar{j})}\right\} \bigcap \left\{x^*_{(i)} | \exists \hat{x} \text{ s.t. } \left\|\hat{x} - x_{(i)}\right\| \leq \tau\right\}\right| \geq j$$

where $\bar{j}$ is the largest integer such that $f^*_{(\bar{j})} = f^*_{(j)}$.

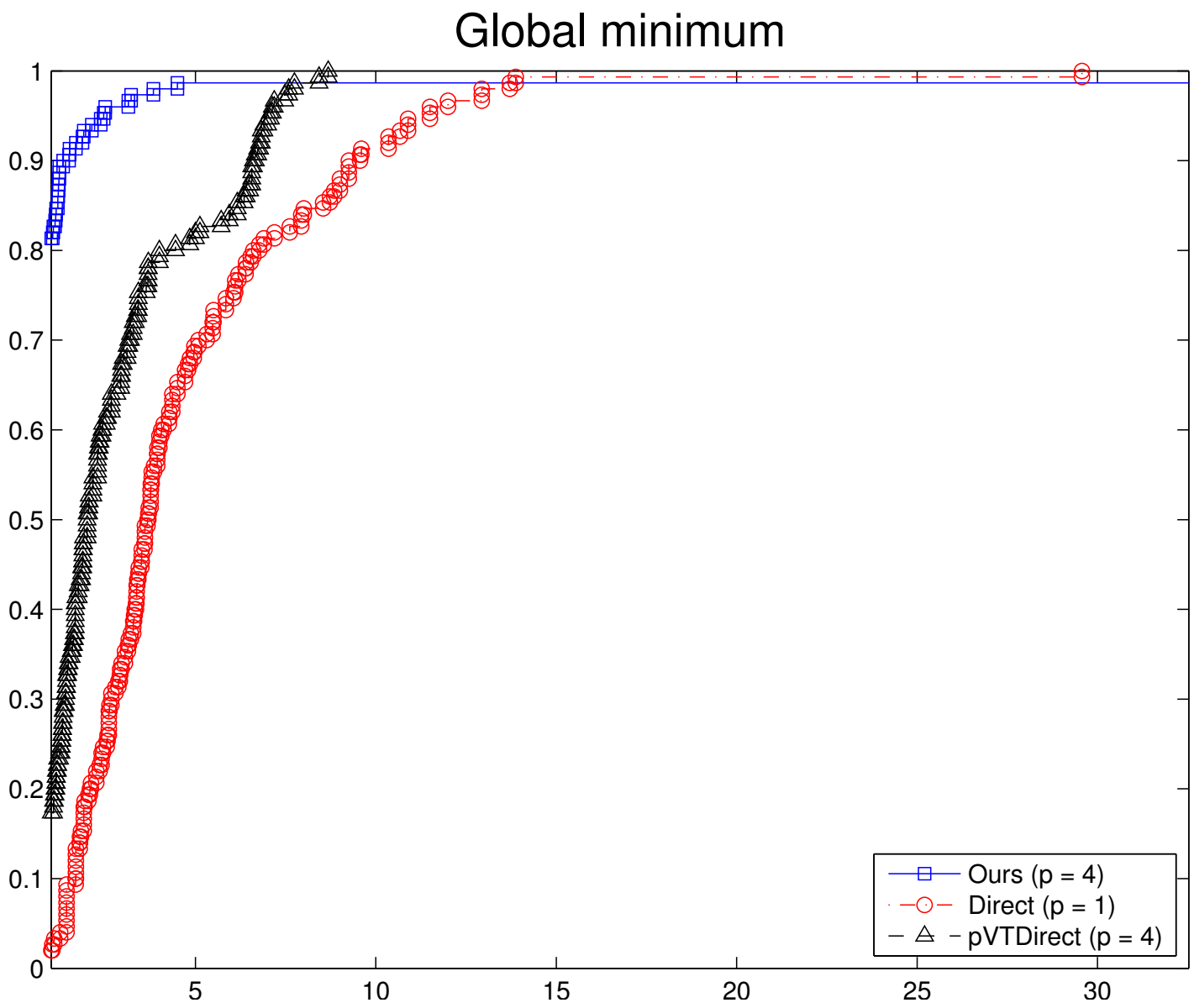
# Measuring Performance: $\tau = 10^{-1}$



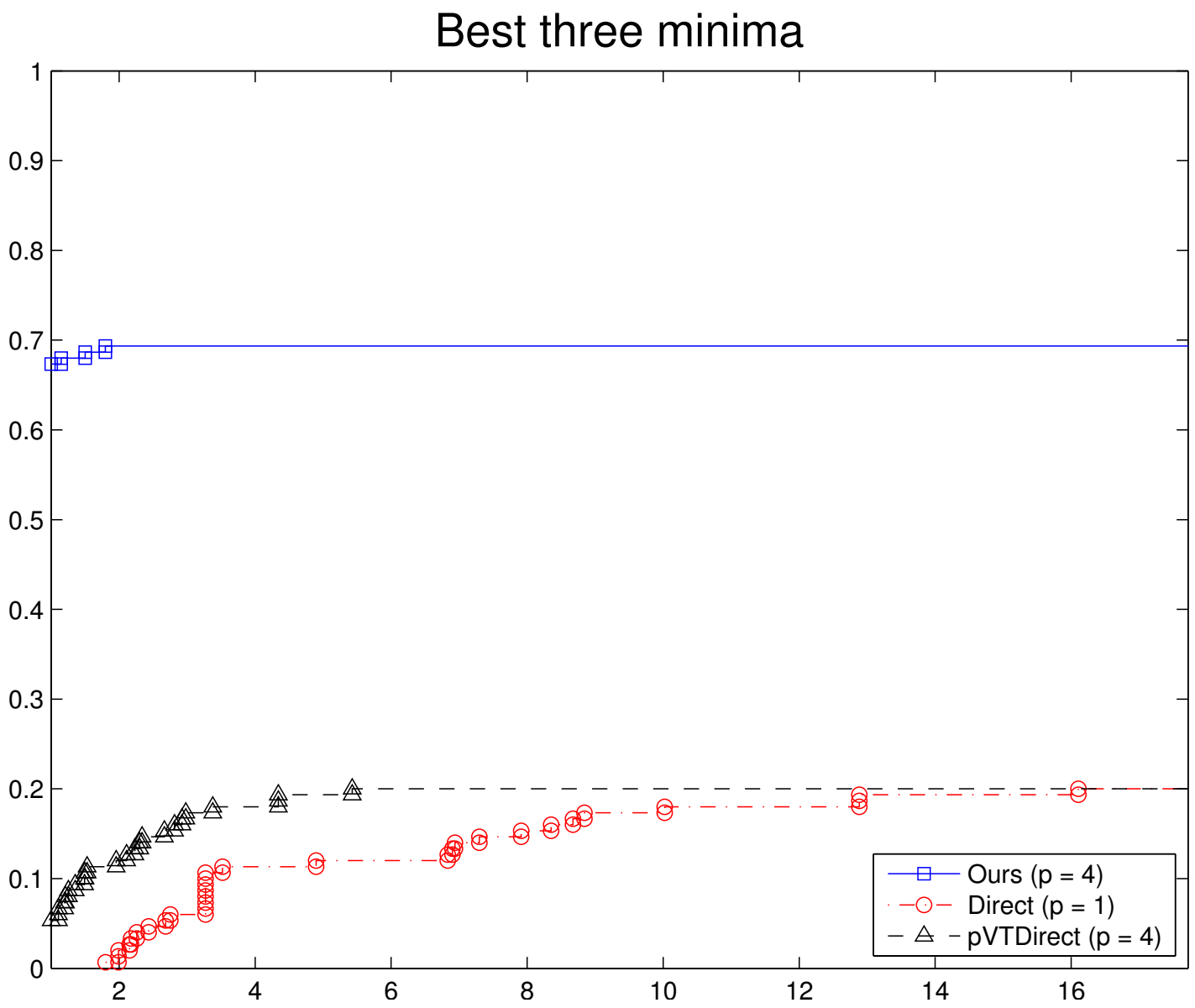
# Measuring Performance: $\tau = 10^{-1}$



# Measuring Performance: $\tau = 10^{-3}$



# Measuring Performance: $\tau = 10^{-3}$



# Conclusions and Future Work

- Concurrent function evaluations is necessary for finding many local minima.
- Local models efficiently use previous function evaluations.

Questions:

- Finding (or designing) the best local solver for our framework.
- Allocating resources depending on how the function evaluation scales for "timely" optimization.

Research supported by US Department of Energy Office of Science