Alternating Variable Method (AVM)

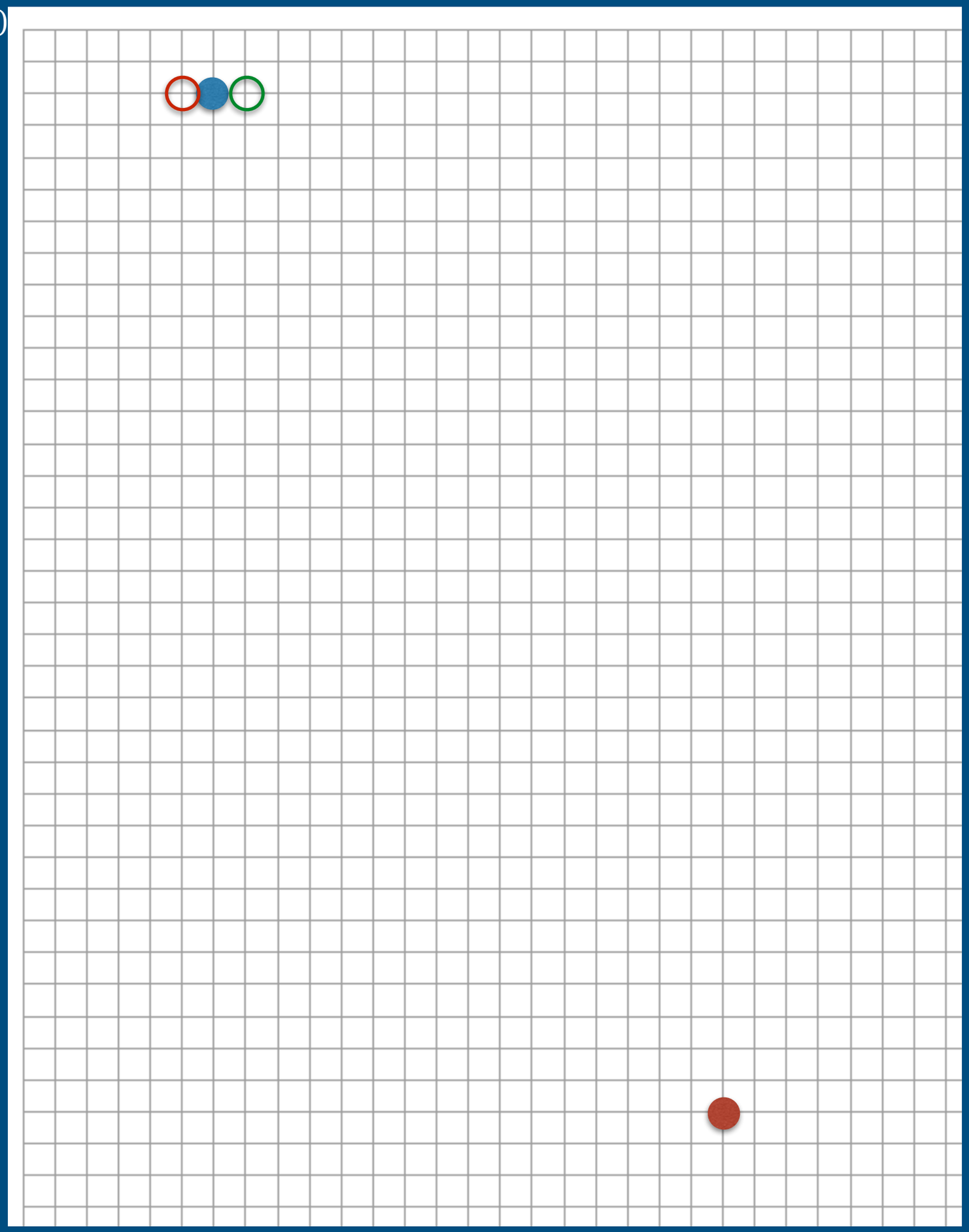
- A type of Pattern Search: searches for an input vector that can maximise/minimise a given objective function
- It has two operation modes: exploratory move, and pattern move.
- For each variable:
  - Use exploratory move to decide which direction results in fitter solutions
  - Use pattern move to accelerate to that direction

Alternating Variable Method

- Based on the known empirical results, AVM is one of the most effective algorithm for achieving C/C++ structural coverage

  - M. Harman and P. McMinn. A theoretical and empirical analysis of evolutionary testing and hill climbing for structural test data generation. In Proceedings of the International Symposium on Software Testing and Analysis (ISSTA 2007), pages pp. 73–83. ACM Press, July 2007.

  - M. Harman and P. McMinn. A theoretical and empirical study of search based testing: Local, global and hybrid search. IEEE Transactions on Software Engineering, 36(2): 226–247, 2010.



AVM: Exploratory Move

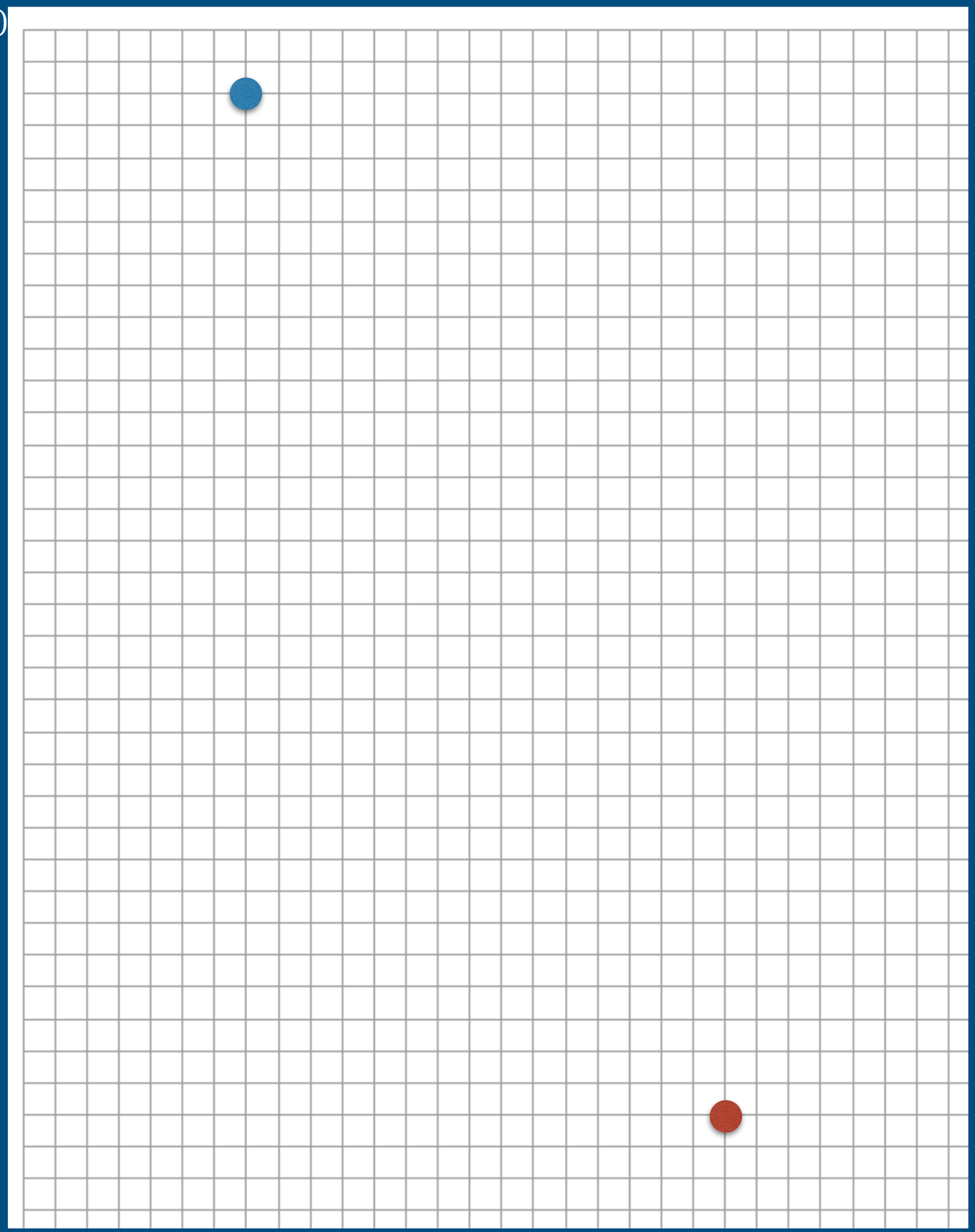
Starting from(6, 2), we want to search for the red dot at (22, 34). We can measure the distance to the goal.

First we try exploratory move for x: make the smallest change, and see which direction results in reduced distance. The initial distance is 35.77.

-1: (5, 2) **Increased** (36.23). X

+1: (7, 2) **Decreased** (35.34) O

Consequently, x needs to be increased at the moment.



AVM: Pattern Move

Now that we decided to increase x, try doubling the difference as long as the distance continues to decrease. At the beginning of the pattern move, x is equal to 7.

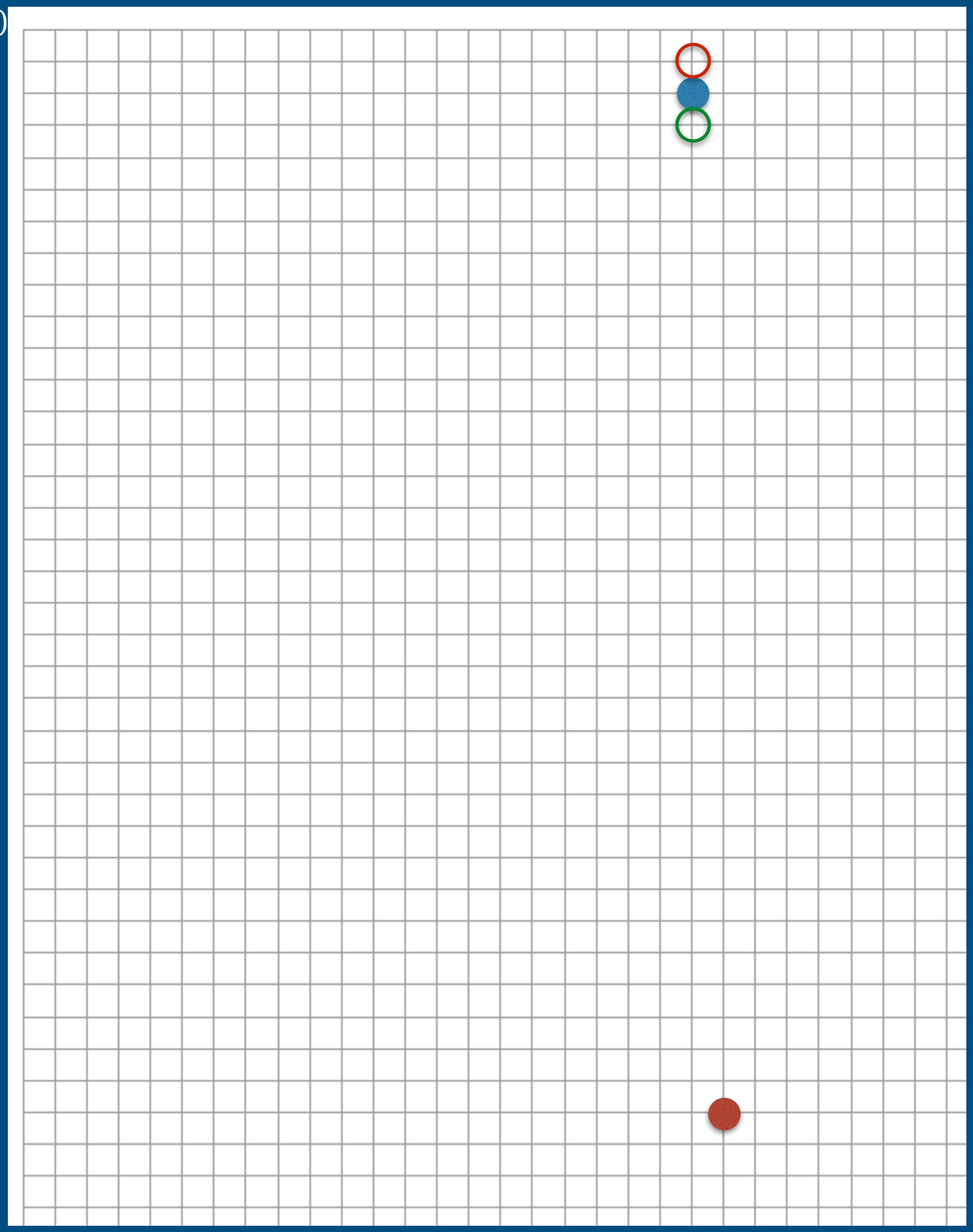
x = 9 (Δx=2): **decrease** (34.53)

x = 13 (Δx=4): **decrease** (33.24)

x = 21 (Δx=8): **decrease** (32.01)

x = 37?(Δx=16): **increase** (35.34)

With increment of 16, the distance starts to grow: this is called overshooting. In this case, we cancel the last pattern move, and start the exploratory move for the next variable, y.



AVM: Exploratory Move

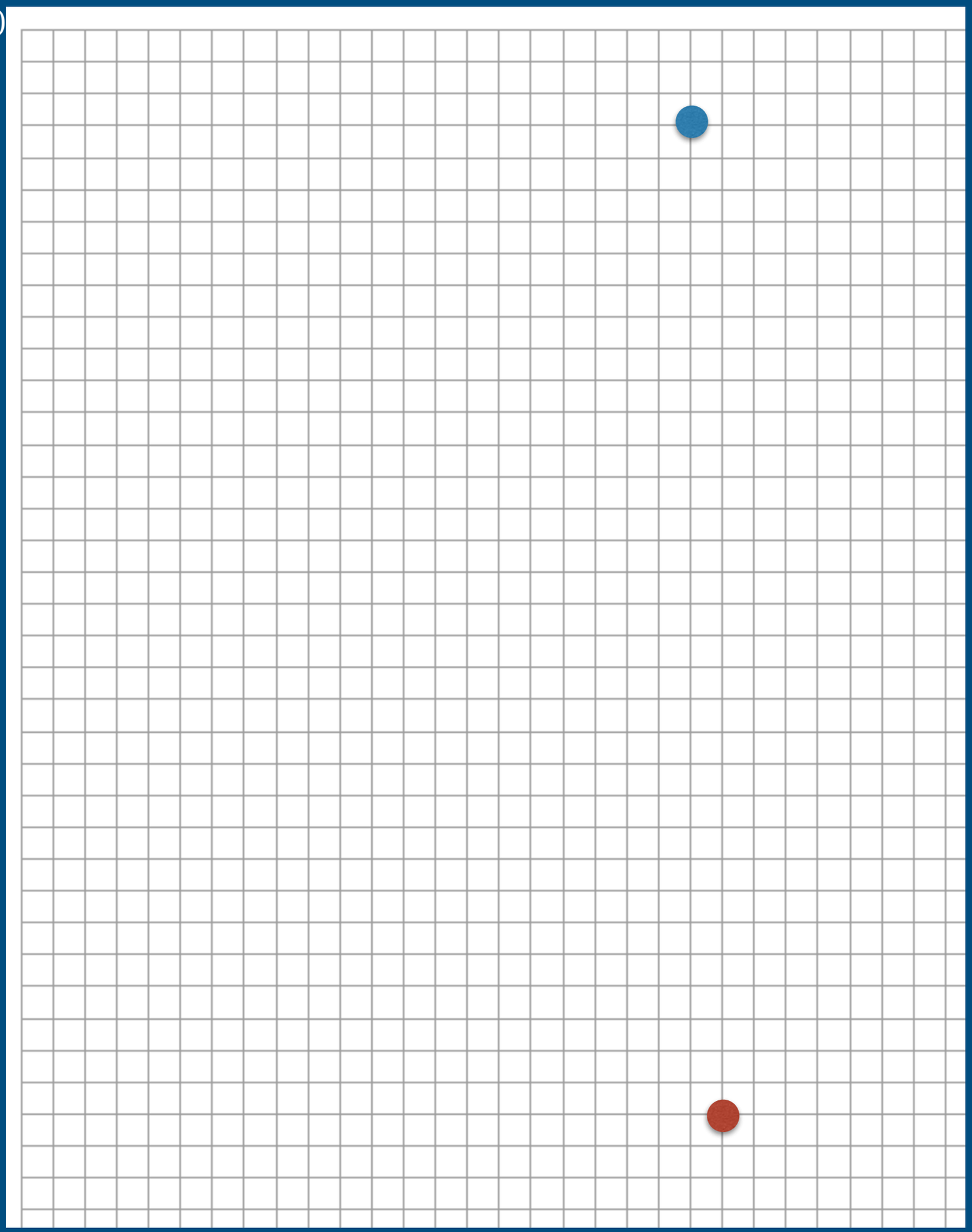
We now change y by 1 and decide the direction. The distance from the last location, (21, 2), is 32.01.

-1: (21, 1) **increase** (33.01). X

+1: (21, 3) **decrease** (31.01) O

So y needs to be increased.

(0, 0)



AVM: Pattern Move

We increase the variable y with pattern moves now. Initially y is 3.

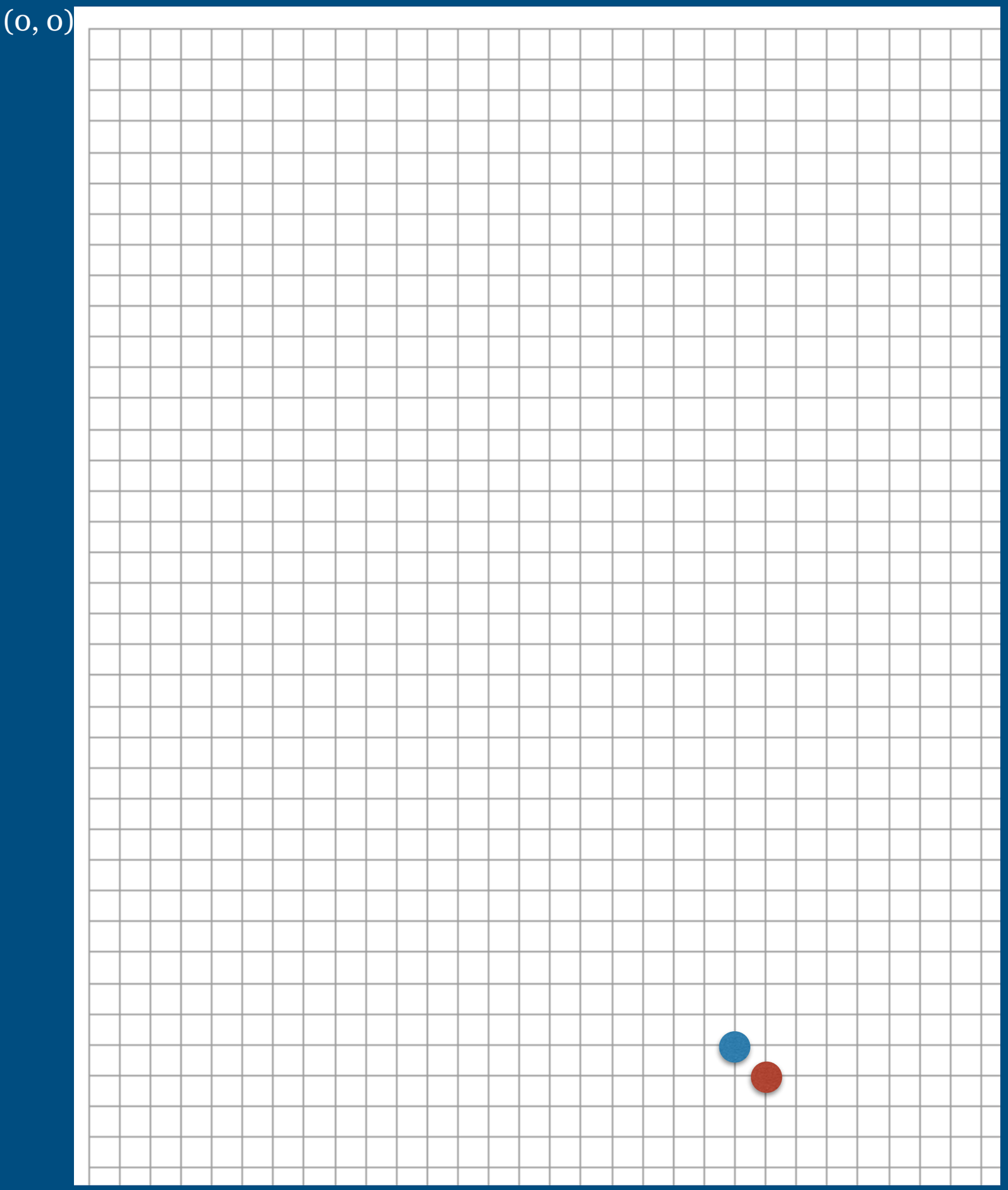
y = 5 (Δy=2): **decrease** (29.01)

y = 9 (Δy=4): **decrease** (25.01)

y = 17 (Δy=8): **decrease** (17.02)

y = 33(Δy=16): **decrease** (1.41)

y = 65(Δy=32): **Overshooting!**



AVM: Exploratory Move

After overshooting of y, we start the exploratory move for x. We decide to increase, but as soon as we try +2, it overshoots. After cancellation of this, we have the correct x.

After one more exploratory move for y, we reach the goal.

Alternating Variable Method

- For a reference implementation and basic applications, see: http://avmframework.org

- P. McMinn and G. M. Kapfhammer. AVMf: An open-source framework and implementation of the alternating variable method. In International Symposium on Search-Based Software Engineering (SSBSE 2016), volume 9962 of Lecture Notes in Computer Science, pages 259–266. Springer, 2016.