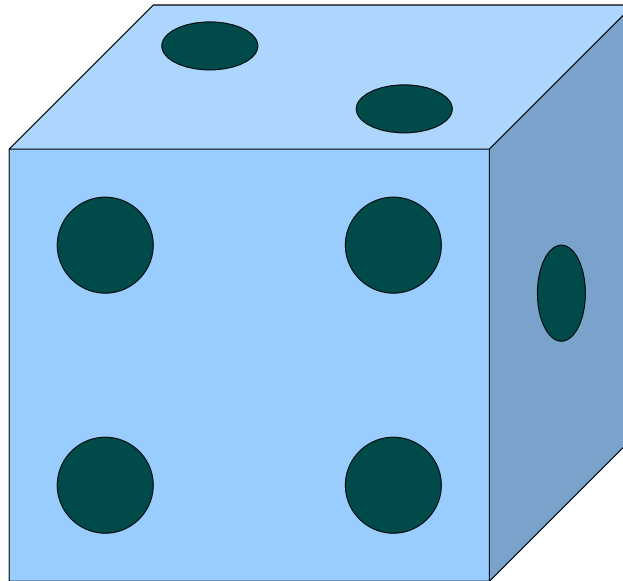
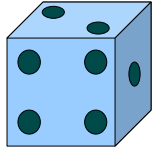


# E.A. Evolutionary Algorithms

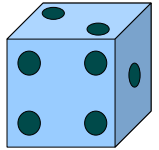


En andere zoek algorithmen met virtuele dobbelstenen

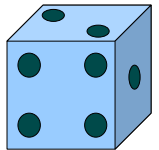
# Overzicht



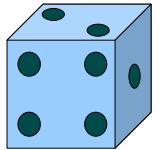
Zoeken?  
Parameter spaces. Local Optima



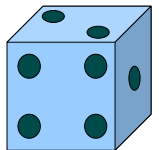
Hill climbing (berg beklimmen)



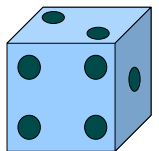
Simulated Annealing



GA – Genetic Algorithm  
EA – Evolutionary Algorithm

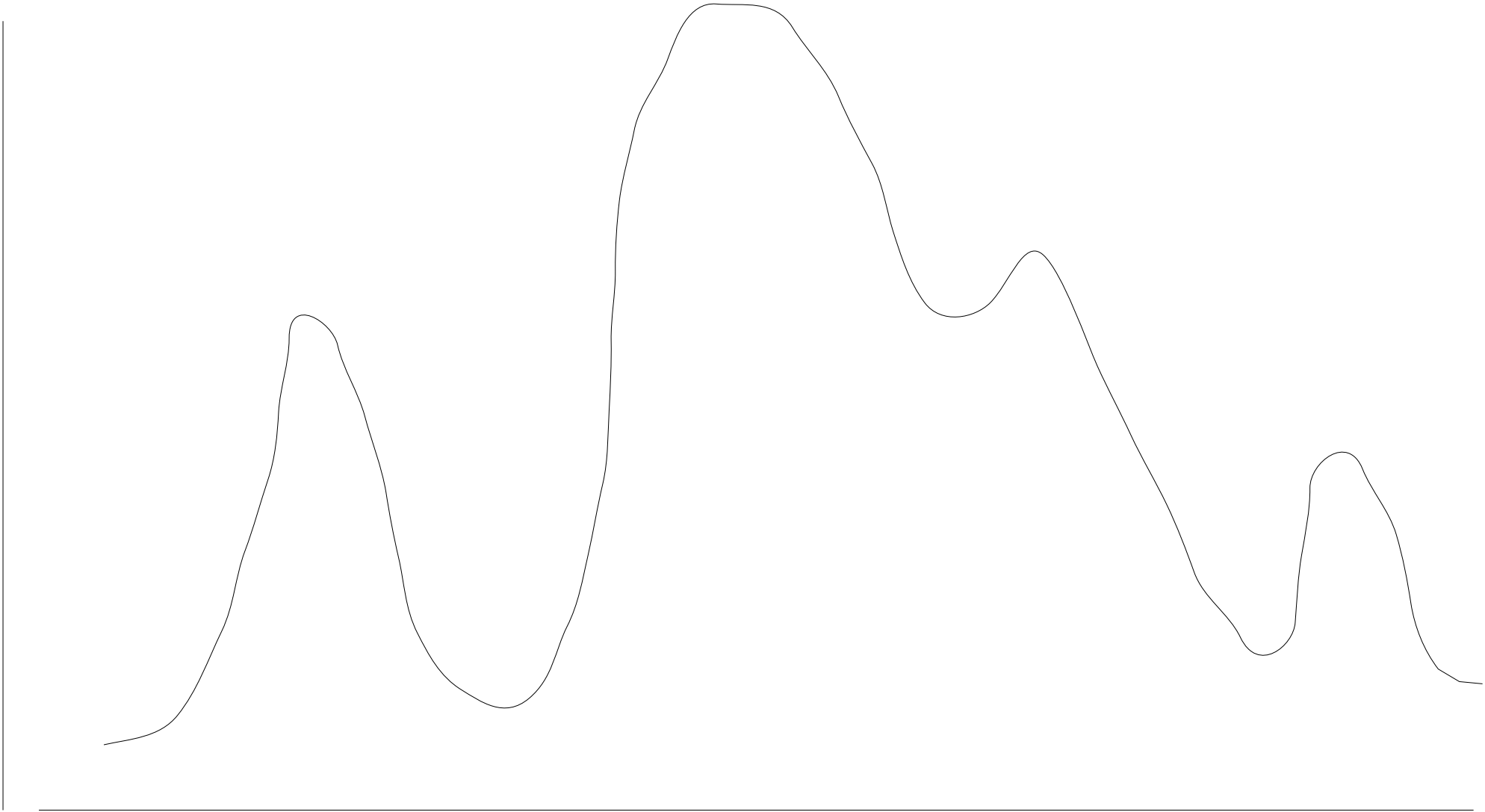


GP – Genetic Programming



Swarms  
Multi Agent systems

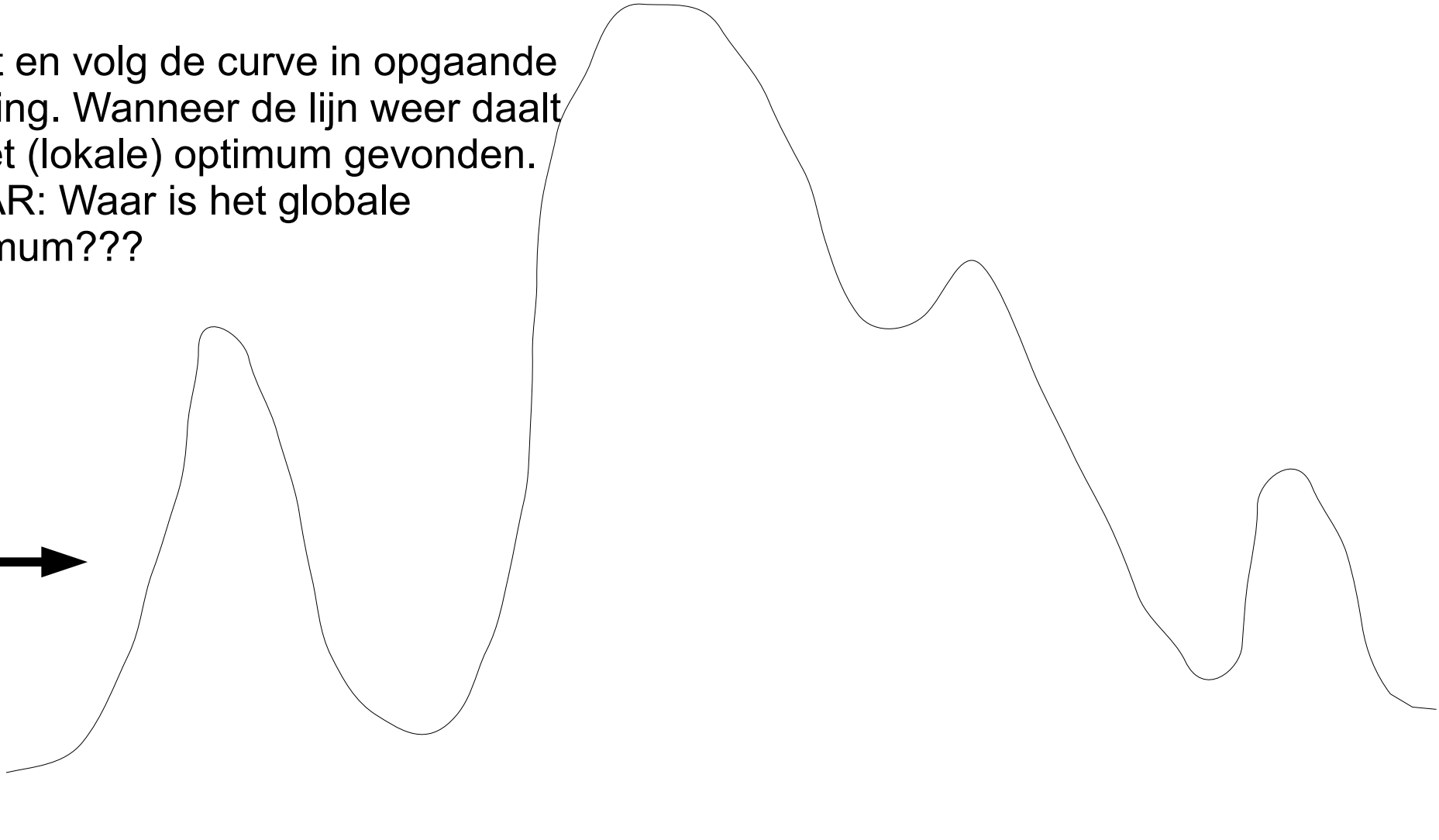
# Zoek $X$ zodat $Y$ maximaal is



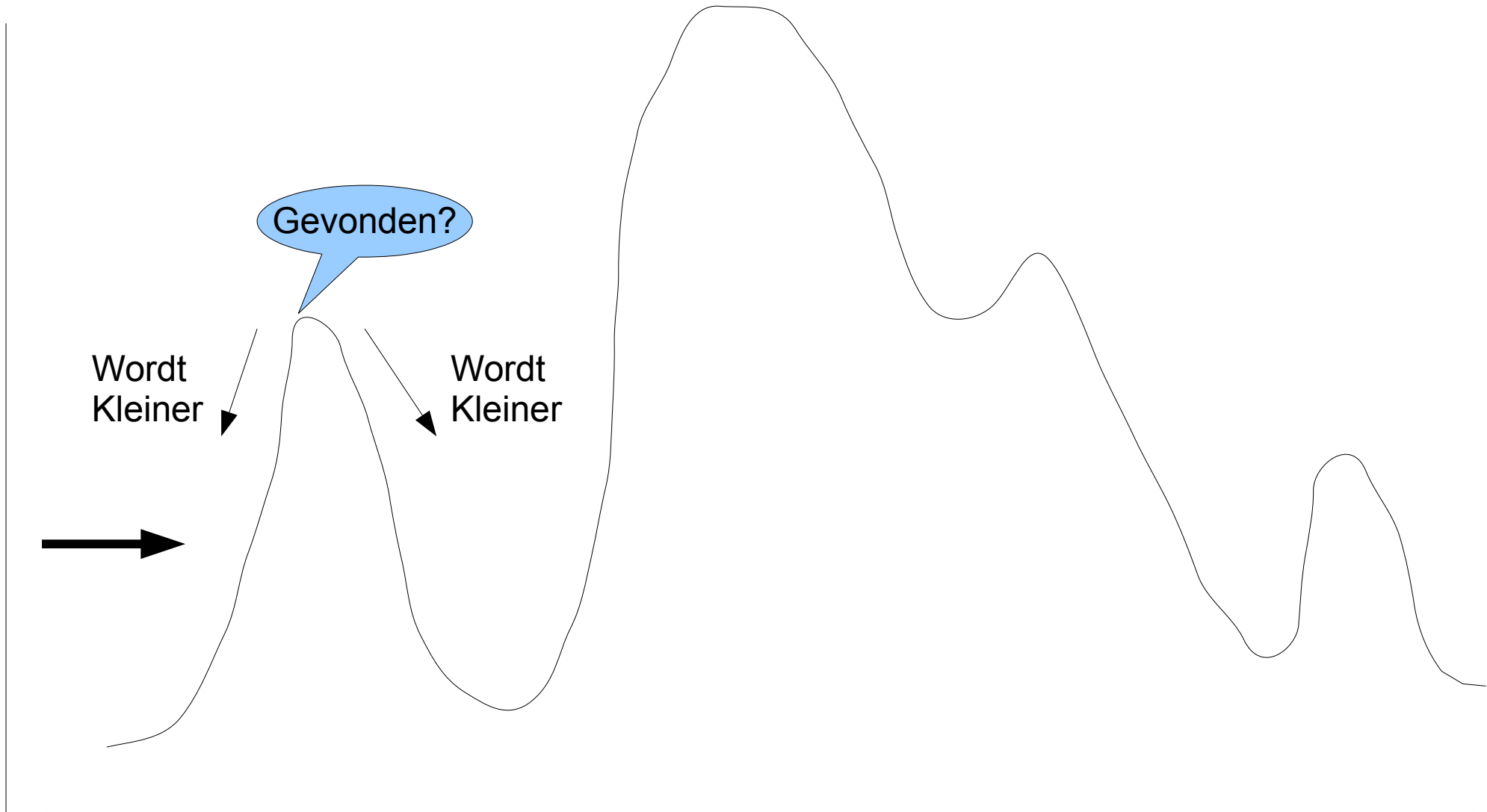
# Zoek X zodat Y maximaal is

## Hill climbing

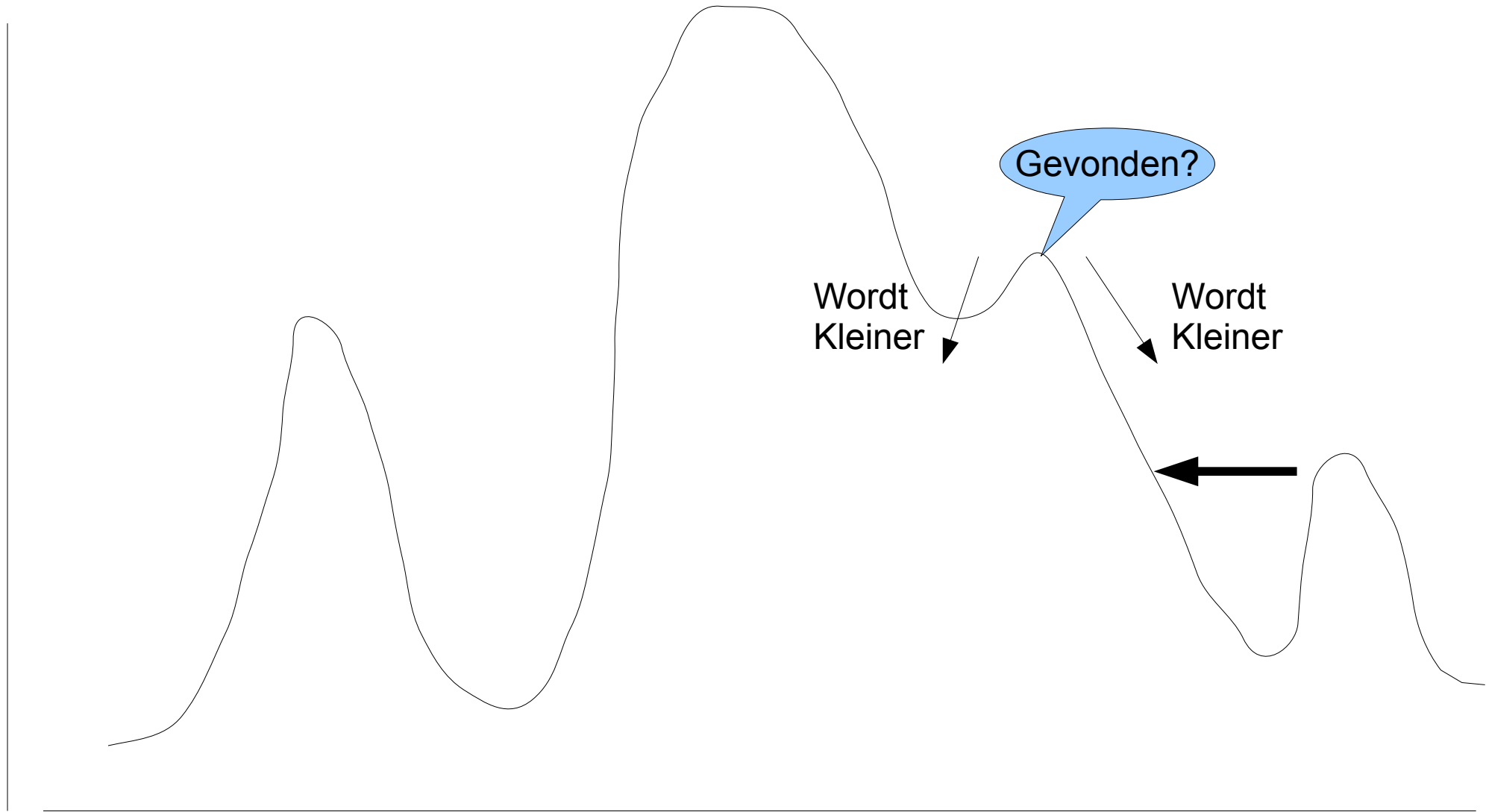
Start en volg de curve in opgaande richting. Wanneer de lijn weer daalt is het (lokale) optimum gevonden.  
MAAR: Waar is het globale optimum???



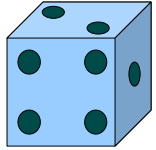
# Zoek X zodat Y maximaal is



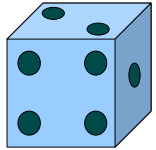
# Zoek X zodat Y maximaal is



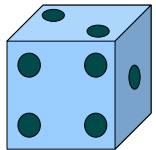
# Wanneer EA (etc) toepassen



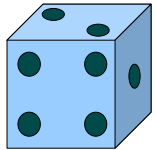
Veel lokale optima.  
Onbekende curven (globale maximum is onbekend).



Veel parameters (100 tot zelfs 1000'en)



Interactie tussen verschillende parameters

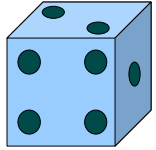


Combinatorial explosion. Alles doorzoeken onmogelijk!

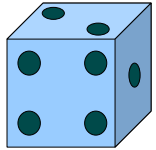
100 schakelaars is  $2^{100}$  mogelijkheden.  
Voor analoge waarden in procenten is  $100^{100}$  mogelijkheden.

Dit soort getallen zijn groter als het aantal atomen in het zonnestelsel!

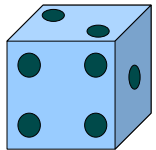
# SA - Simulated Annealing



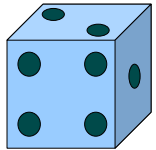
Geinspireerd op Annealing (uitgloeien) van metaal.  
Gecontroleerd afkoelen van metaal verhoogd kwaliteit



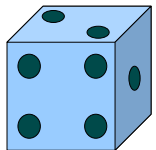
Belangrijkste parameters: virtuele temperatuur (en delta)  
 $T$  en  $\Delta T$



Begin met hoge temperatuur (grote stappen in zoek ruimte)  
Verlaag temperatuur langzaam.



Bewaar beste resultaat. Sta toe om soms met slechter  
resultaat door te gaan om lokale minima te ontlopen

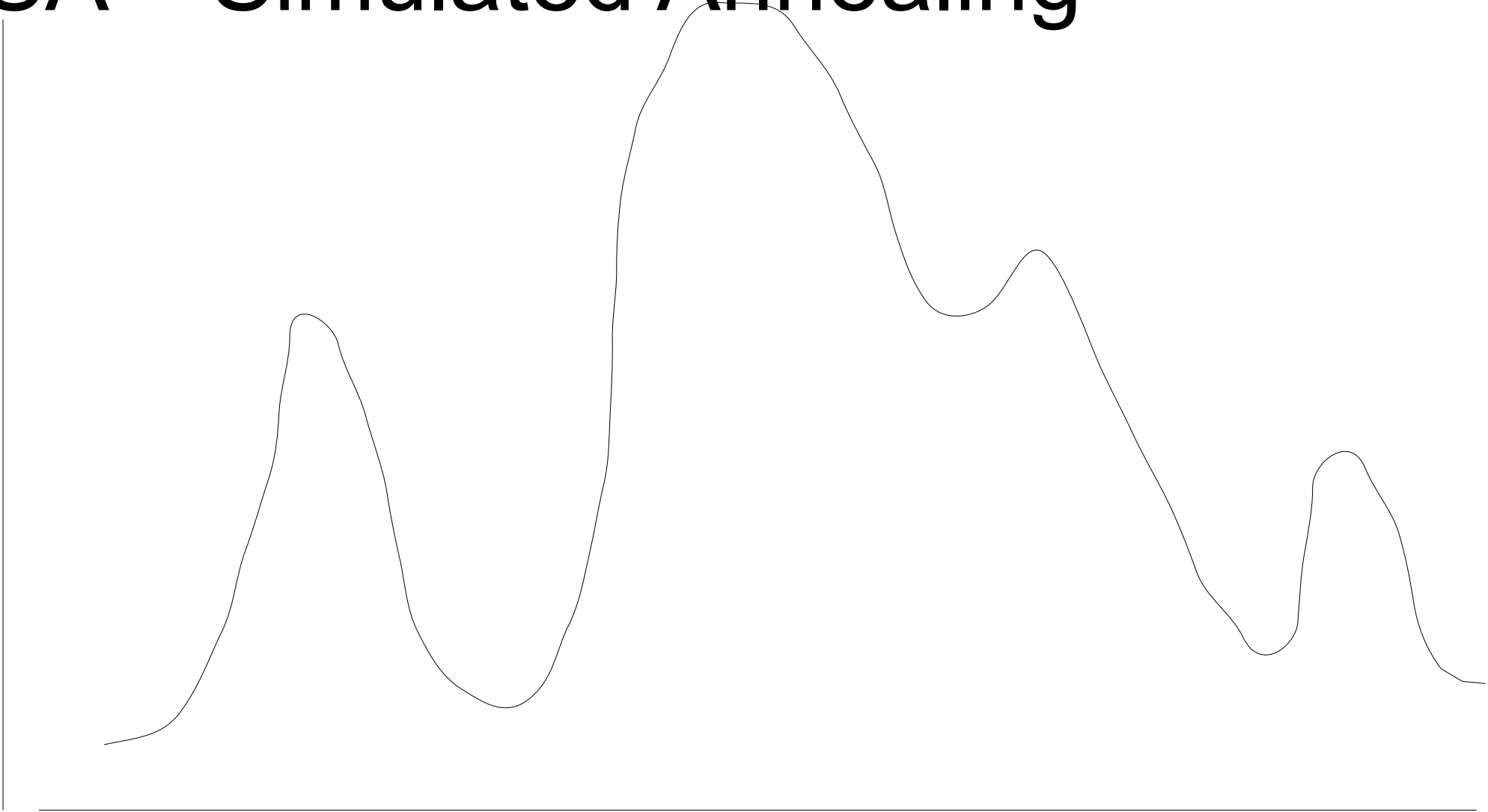


Geen garantie dat minima bereikt wordt!  
Meestal is een goede oplossing goed genoeg,  
Absoluut beste niet altijd nodig!

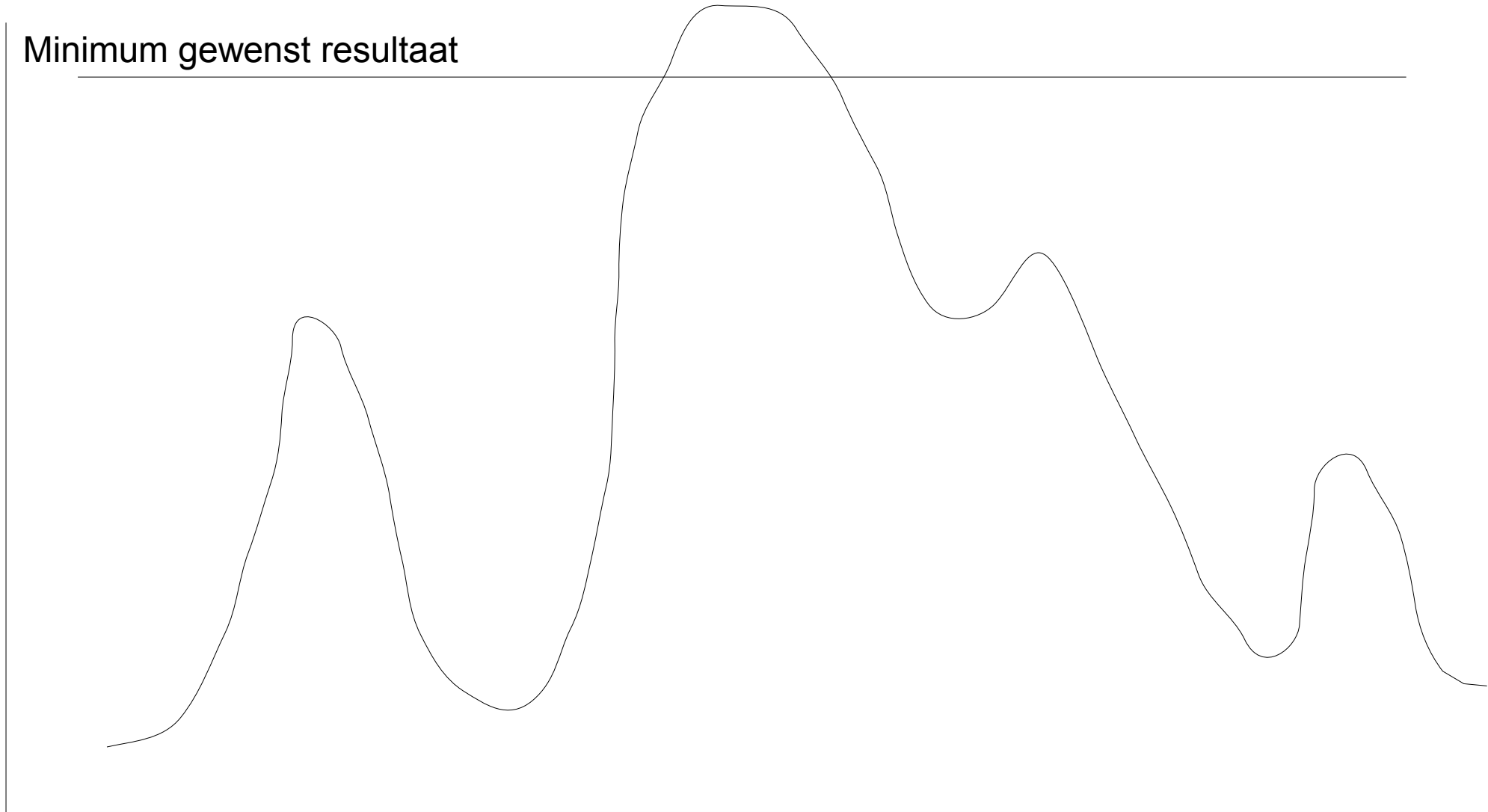
(meestal onmogelijk om “beste” te vinden!)



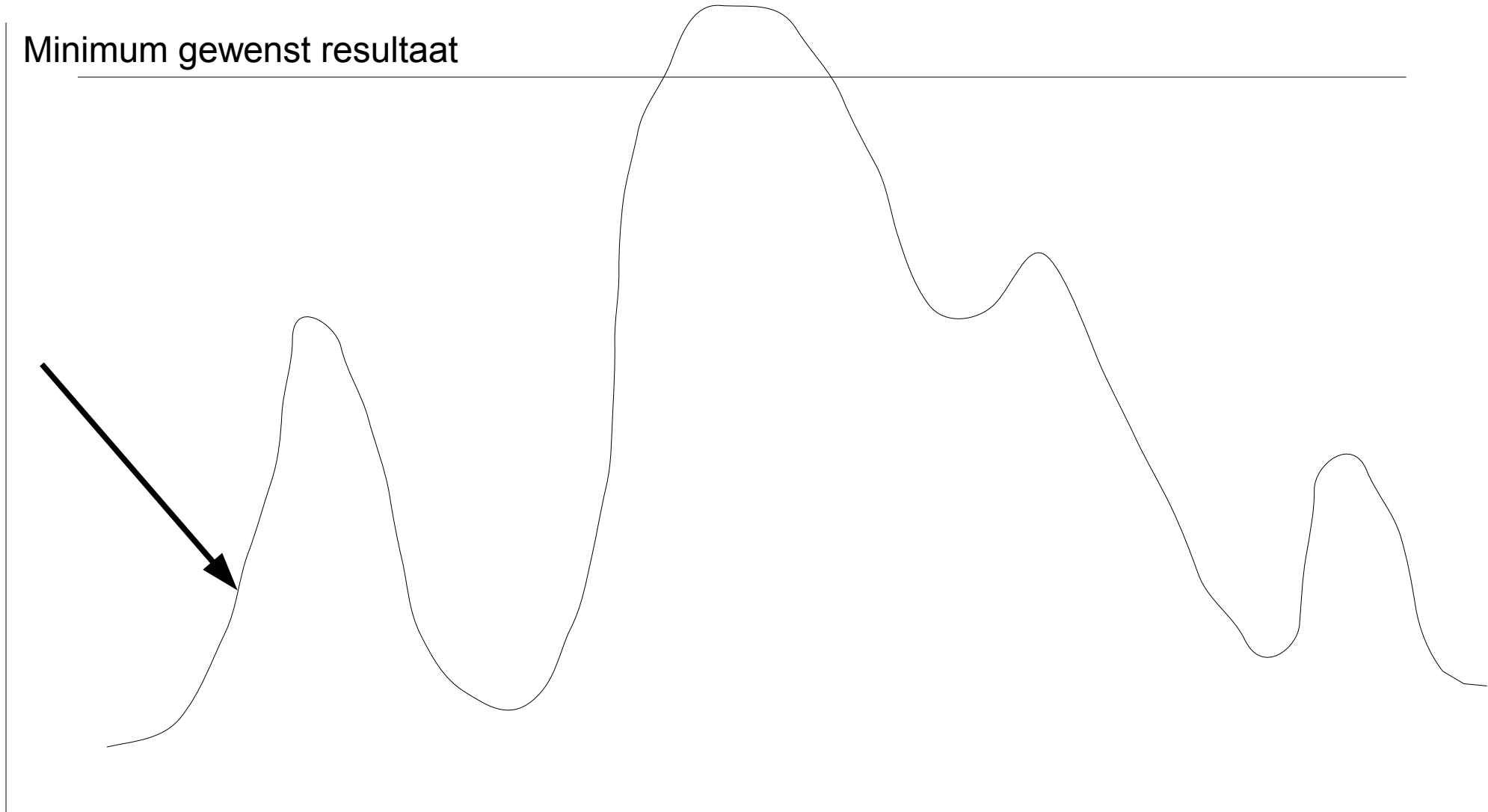
Zoek X zodat Y maximaal is  
SA – Simulated Annealing



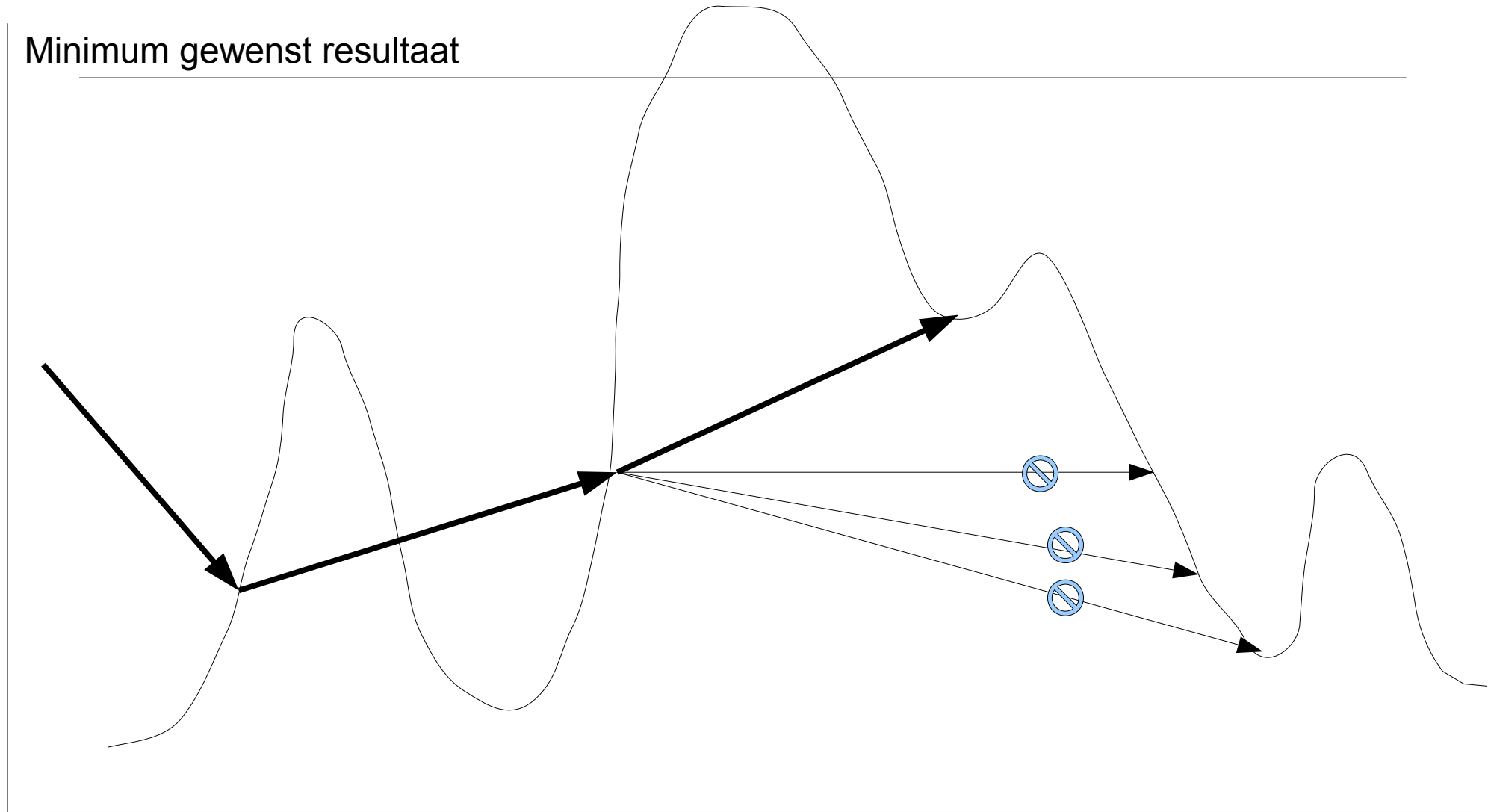
# Zoek X zodat Y maximaal is



# Zoek X zodat Y maximaal is



# Zoek X zodat Y maximaal is

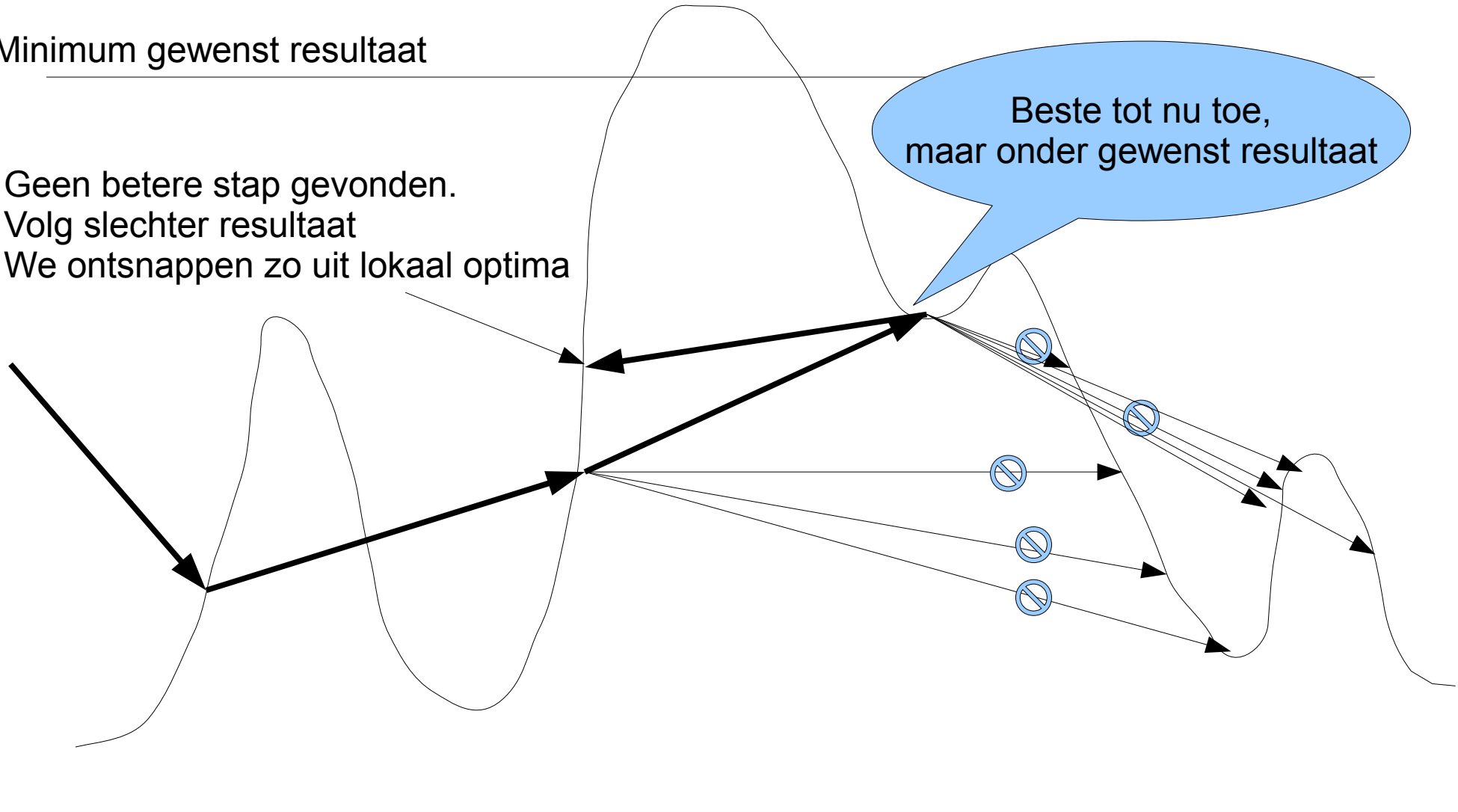


# Zoek X zodat Y maximaal is

Minimum gewenst resultaat

Geen betere stap gevonden.  
Volg slechter resultaat  
We ontsnappen zo uit lokaal optima

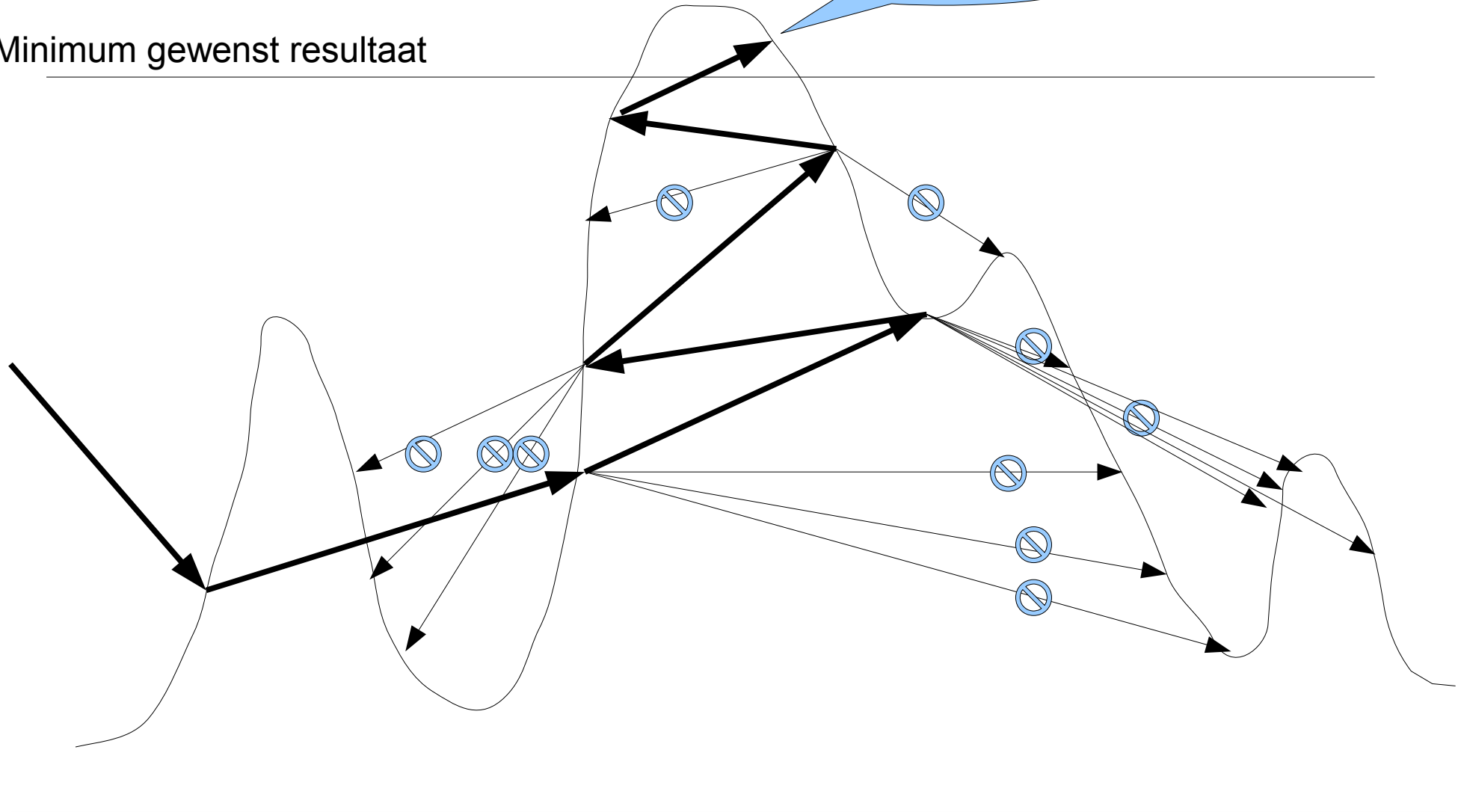
Beste tot nu toe,  
maar onder gewenst resultaat



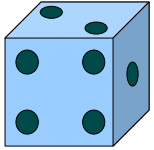
# Zoek X zodat Y maximaal is

Ok = boven minimum

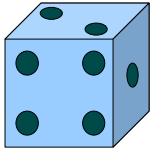
Minimum gewenst resultaat



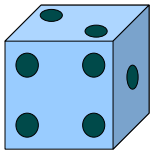
# Simulated Annealing



Simpel om te implementeren  
Zoekt redelijk snel



KISS principe. Probeer SA als eerste.



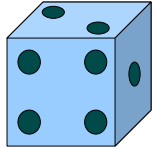
Niet bruikbaar (of in iedergeval minder geschikt):

- Zeer veel lokale optima die allemaal even goed lijken
- Als parameters elkaar beïnvloeden.
- Als er meer als 1 stop of zoek criteria is

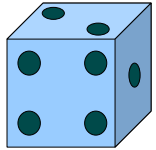
bijvoorbeeld:

- beeld kwaliteit vs file-grote (compressie)
- precisie van resultaat vs CPU cycles. (formules)

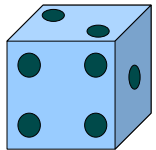
# Goede dobbelstenen



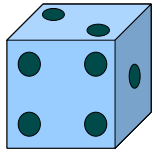
Tijd om iets te vertellen over “virtuele” dobbelstenen.  
We hebben veel, HEEL veel willekeurige waarden nodig.



Gebruik geen rand() of andere simpele functies uit je runtime.  
Ook geen LFSR,  $\langle \text{prime} \rangle * \text{old}$ , EXOR of andere home-brew



Vaak relatie tussen bits in simpele oplossingen.  
Repeteerd vaak al na “slechts”  $2^{31}$  aanroepen.



Gebruik een crypto strong random number generator.  
Bijvoorbeeld:  
Mersenne-Twister (relatief snel en goed)

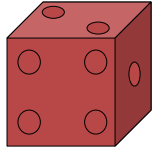
MT19937 algo repeteerd pas na  $2^{19937}-1$  aanroepen!



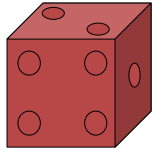
## Mersenne-Twister



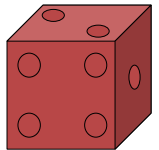
# Overzicht



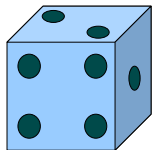
Zoeken?  
Parameter spaces. Local Optima



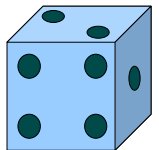
Hill climbing



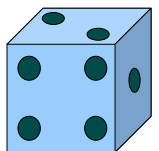
Simulated Annealing



GA – Genetic Algorithm  
EA – Evolutionary Algorithm



GP – Genetic Programming

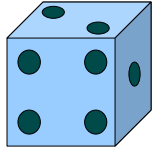


Swarms  
Multi Agent systems

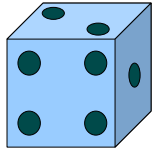


Start demo

# GA – Genetic Algorithm

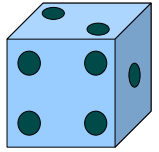


Parameters —————> Bits  
Genome (virtueel DNA)



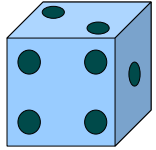
Evaluatie functie  
De stop conditie of condities

Een goede  
evaluatie functie  
is de kern van  
deze methode

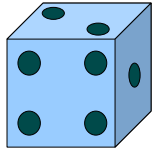


- Populatie grote
- Mutatie rate
- Cross over
- Bewaar beste? (elite)

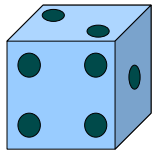
# GA – Genetic Algorithm



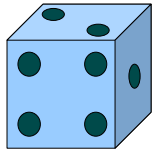
Genereer initiele populatie  
Voer evaluatie uit op alle leden uit de populatie



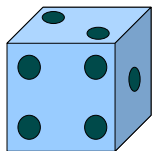
Cross over (virtuele sex), genereer kinderen  
+ Mutatie



Voer evaluatie uit op alle kinderen



Sorteer populatie op fitness, bewaar beste  
- Kies beste uit verzameling ouders + kinderen  
- Kies beste uit kinderen  
- Kies beste uit kinderen + beste ouder(s)

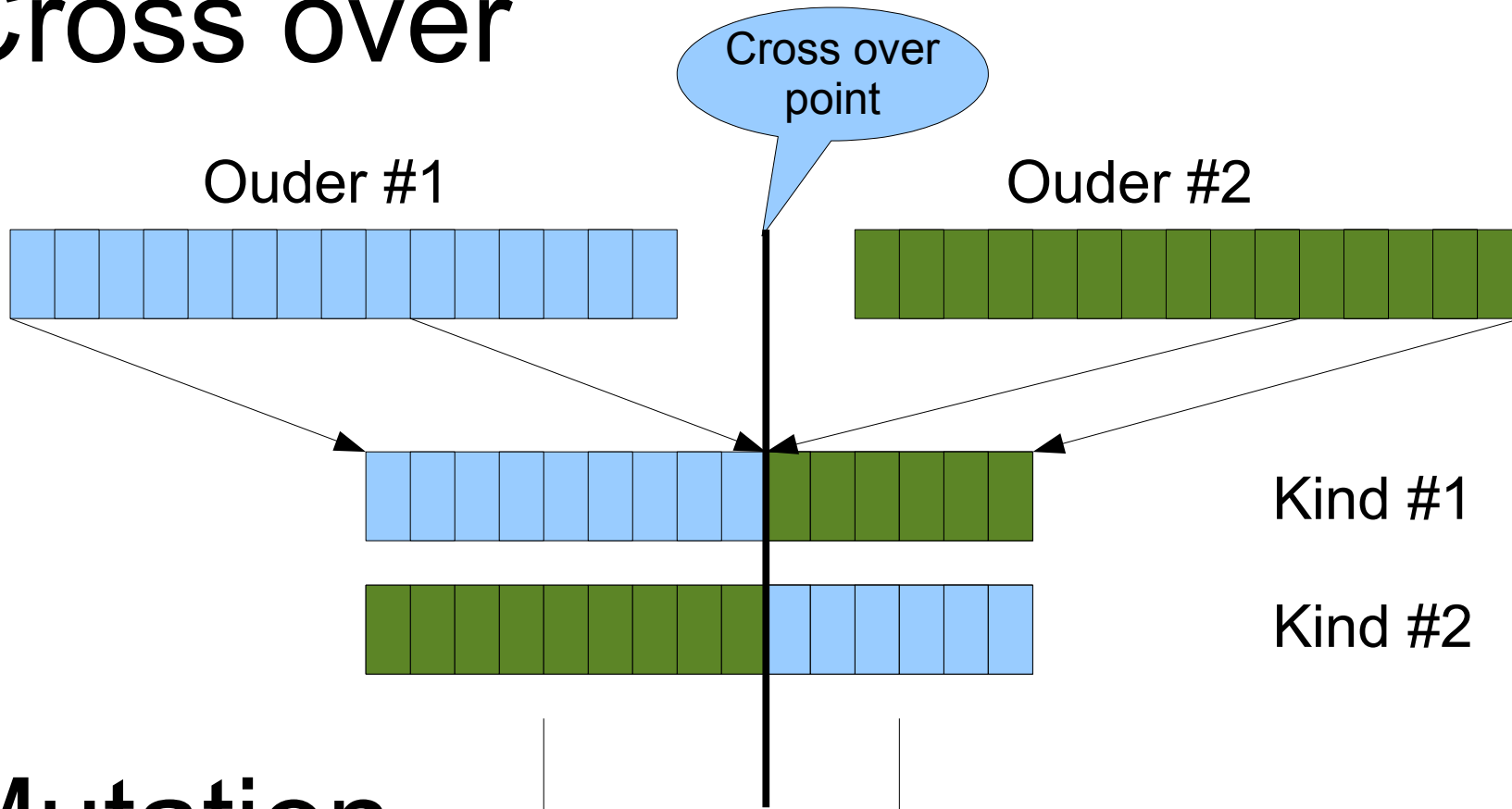


Doe dit zo lang stop criteria niet bereikt is.

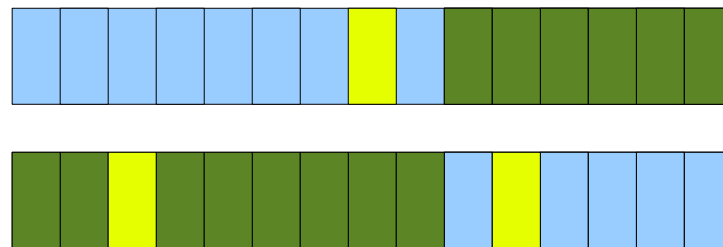


# GA – Genetic Algorithm

## Cross over

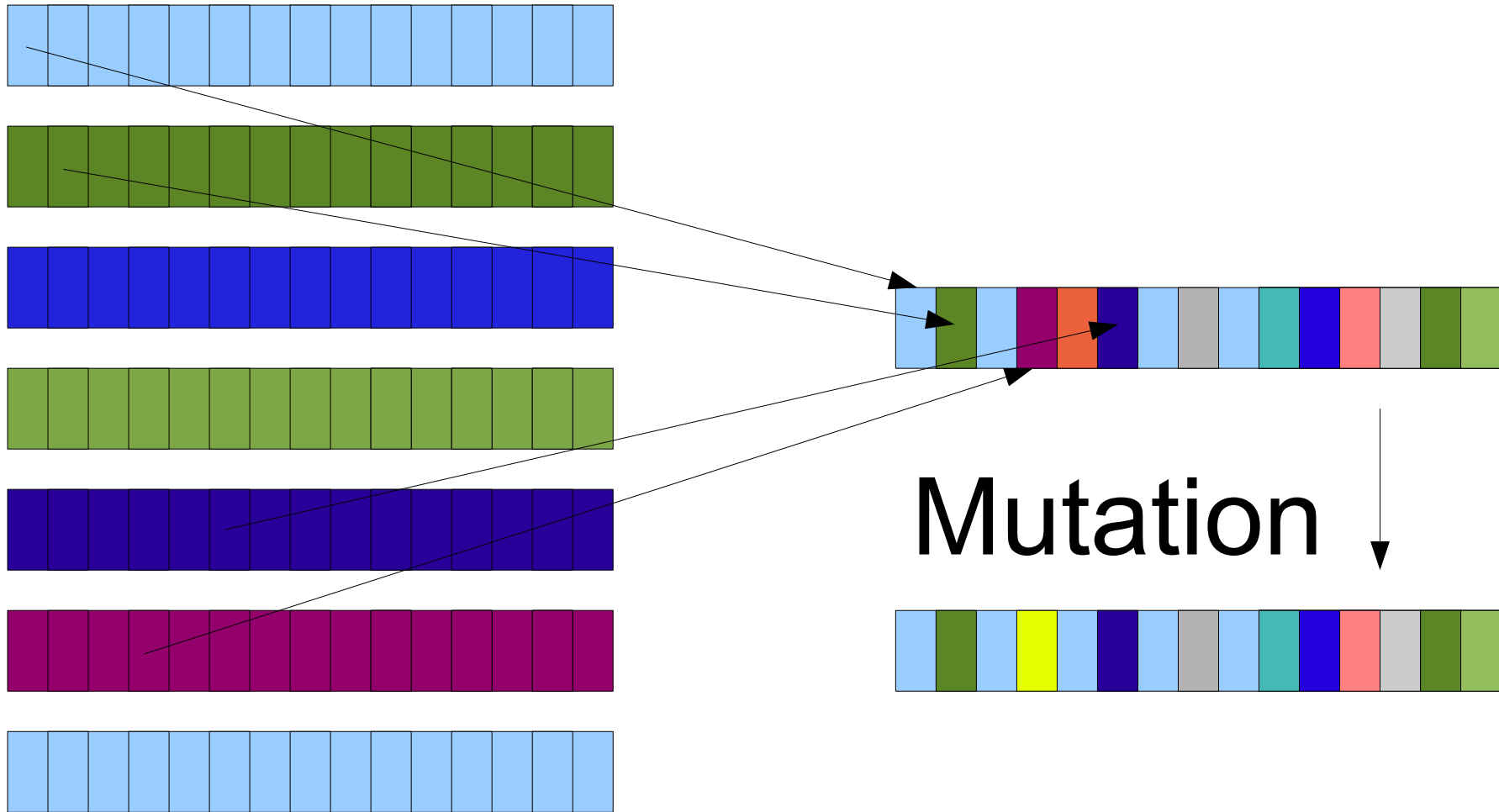


## Mutation



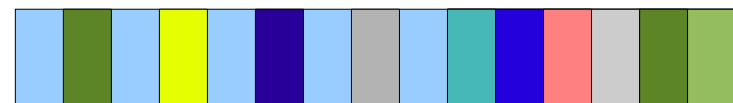
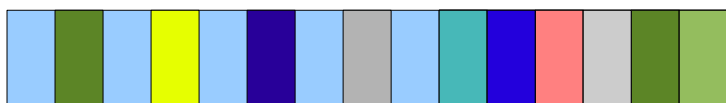
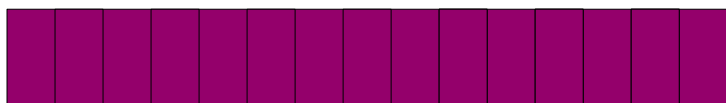
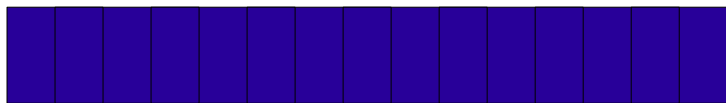
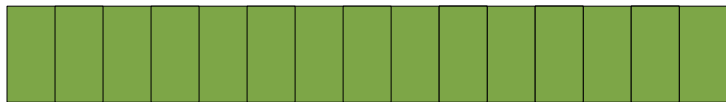
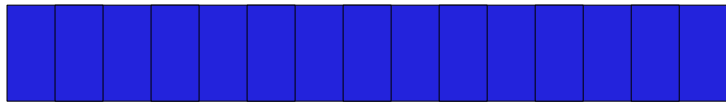
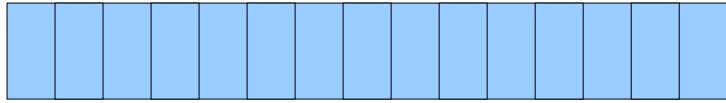
# Harmony search (grappige variant)

Ouders



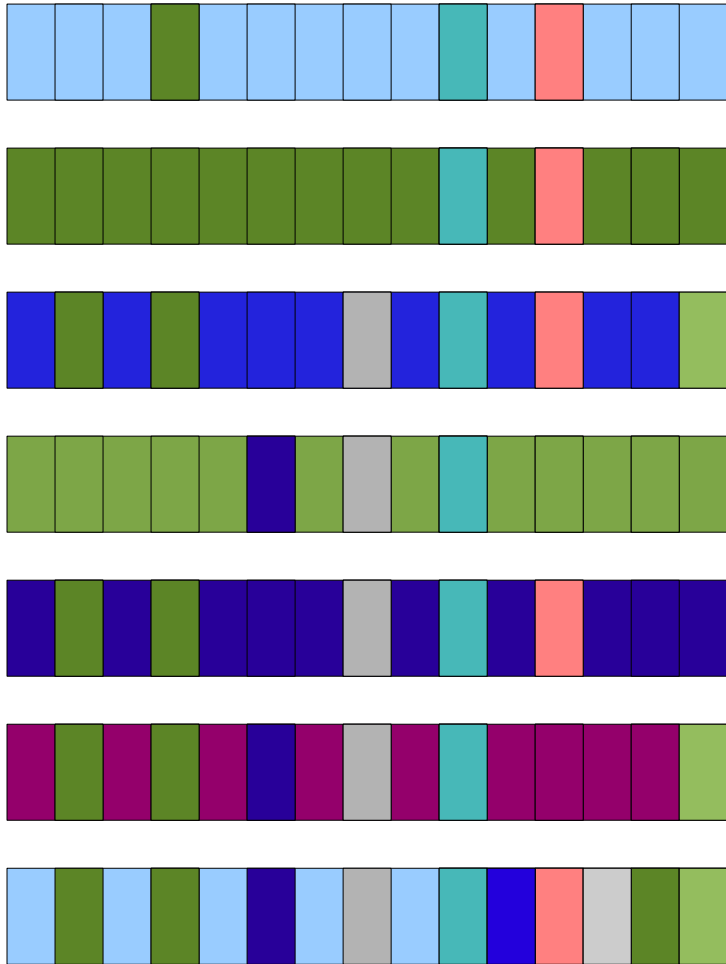
# Harmony search (grappige variant)

Ouders



# Harmony search

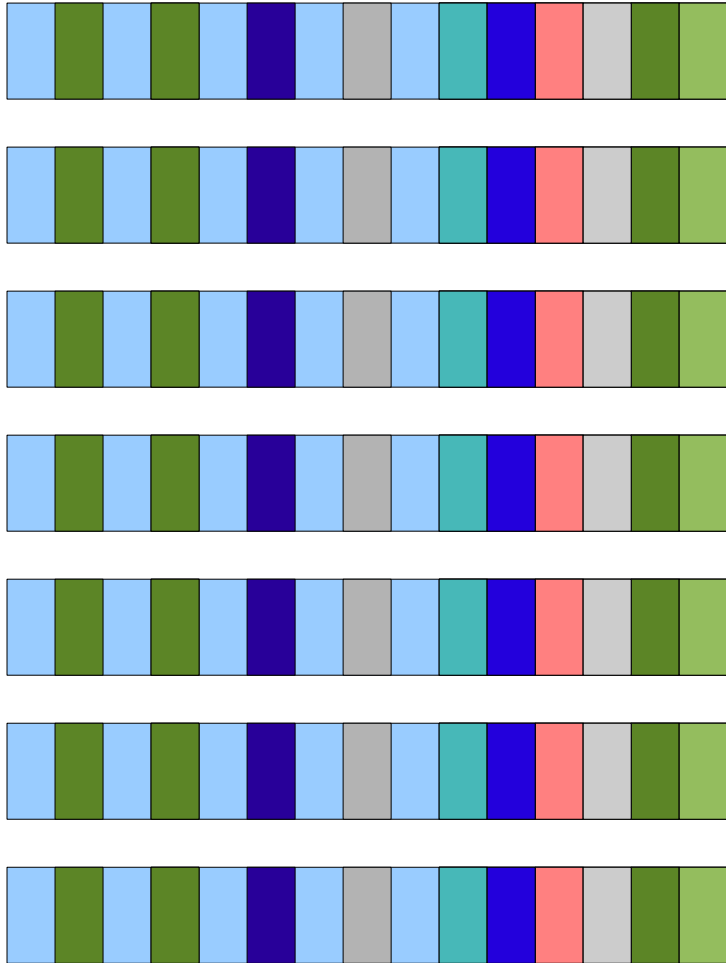
Ouders



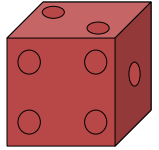


# Harmony search

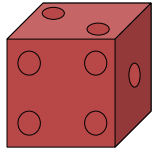
Ouders



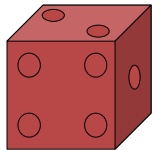
# Overzicht



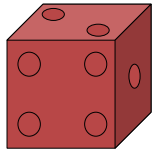
Zoeken?  
Parameter spaces. Local Optima



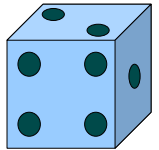
Hill climbing (berg beklimmen)



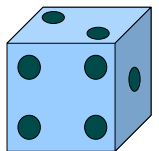
Simulated Annealing



GA – Genetic Algorithm  
EA – Evolutionary Algorithm

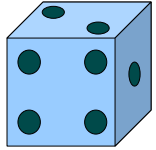


GP – Genetic Programming



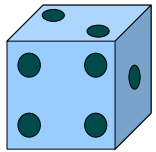
Swarms  
Multi Agent systems

# GP – Genetic Programming

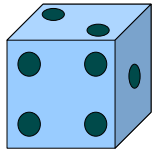


Parameters vormen programma or circuit

- Operatoren
- Constanten
- Variabelen
- (subroutines)

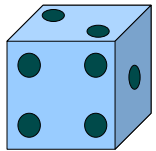


Resultaat is boom (tree) of structuur (architectuur, circuit).

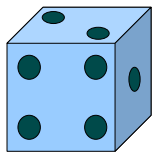


Evalulatie kan heel kostbaar zijn omdat iedere boom (Tree) getest moet worden voor iedere test vector.

Evaluatie is som van alle subtests



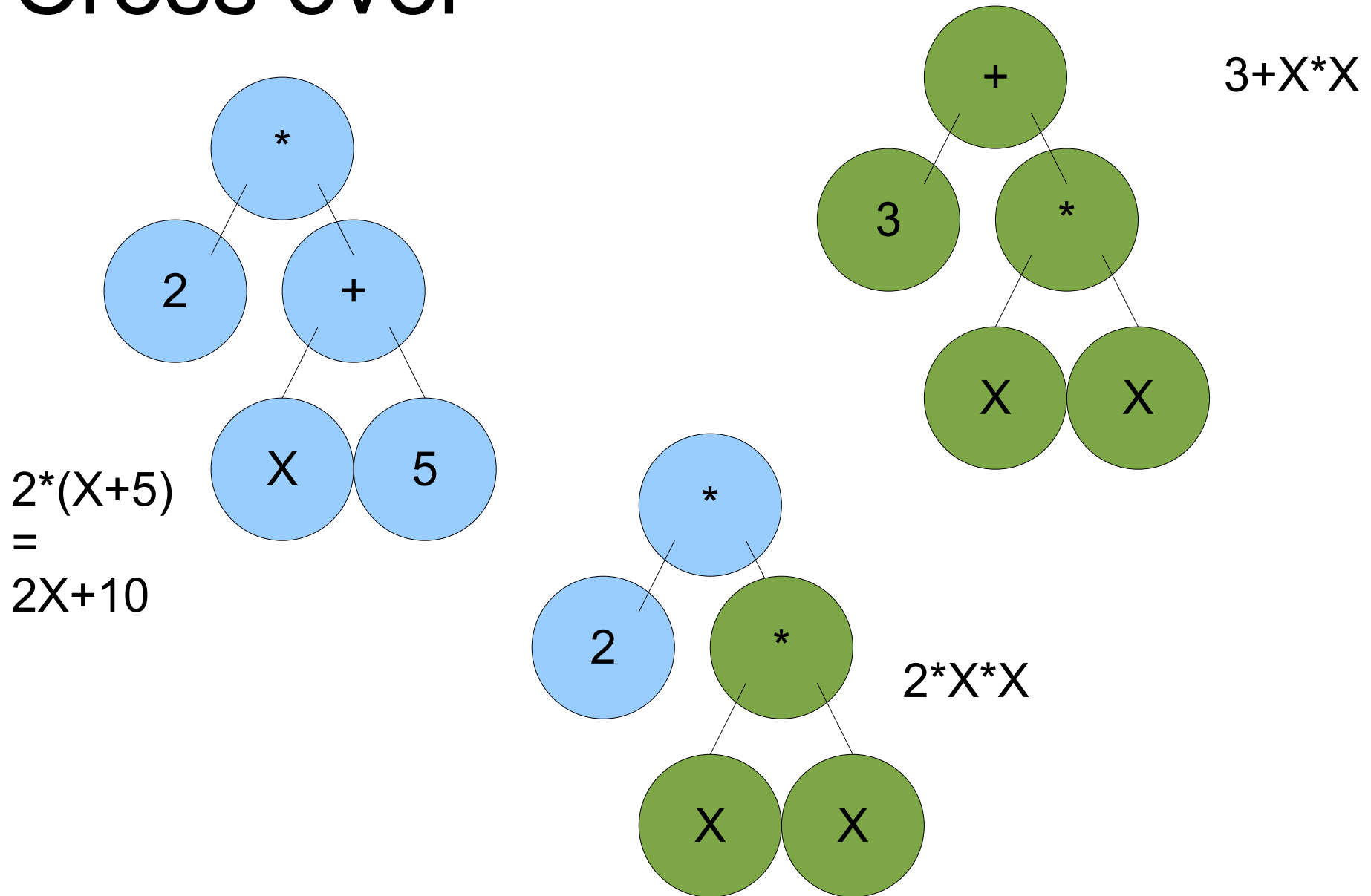
Geen beperkingen in oplossingen. Kan architecturen bouwen. Electronische circuits. Robot brein. 3D structuren.



Door grote van zoek gebied is de methode HEEL CPU intensief. SA, GA of neural networks hebben voorkeur indien mogelijk.

# GP – Genetic Programming

## Cross over



# GP – Genetic Programming

## Resultaten bereikt met GP

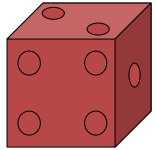
Temperatuur regelaar voor chemische processen

Antennes

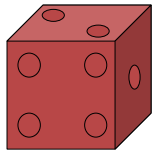
Oscillator circuits

FPGA designs (waarvan niemand snap hoe ze precies werken)

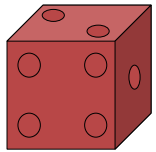
# Overzicht



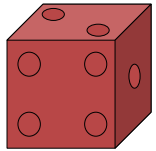
Zoeken?  
Parameter spaces. Local Optima



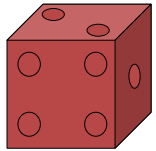
Hill climbing



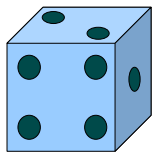
Simulated Annealing



GA – Genetic Algorithm  
EA – Evolutionary Algorithm



GP – Genetic Programming

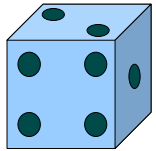


Swarms  
Multi Agent systems

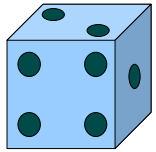
Hoe staat het  
met de demo?

# Swarms.

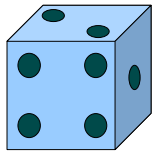
## Doe iets slims met simpele units



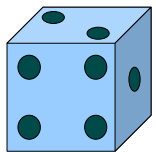
Voorbeelden: Mieren, bijen en vogels in formatie  
Ook bekend als “Emergent behavior”



Kan verbluffend goed werken voor weinig inspanning.  
Oplossing niet CPU efficiënt, maar wel heel robuust!

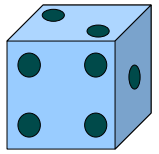


Goed voor Image Classificatie  
(Onze blikvanger is een mooi voorbeeld!)

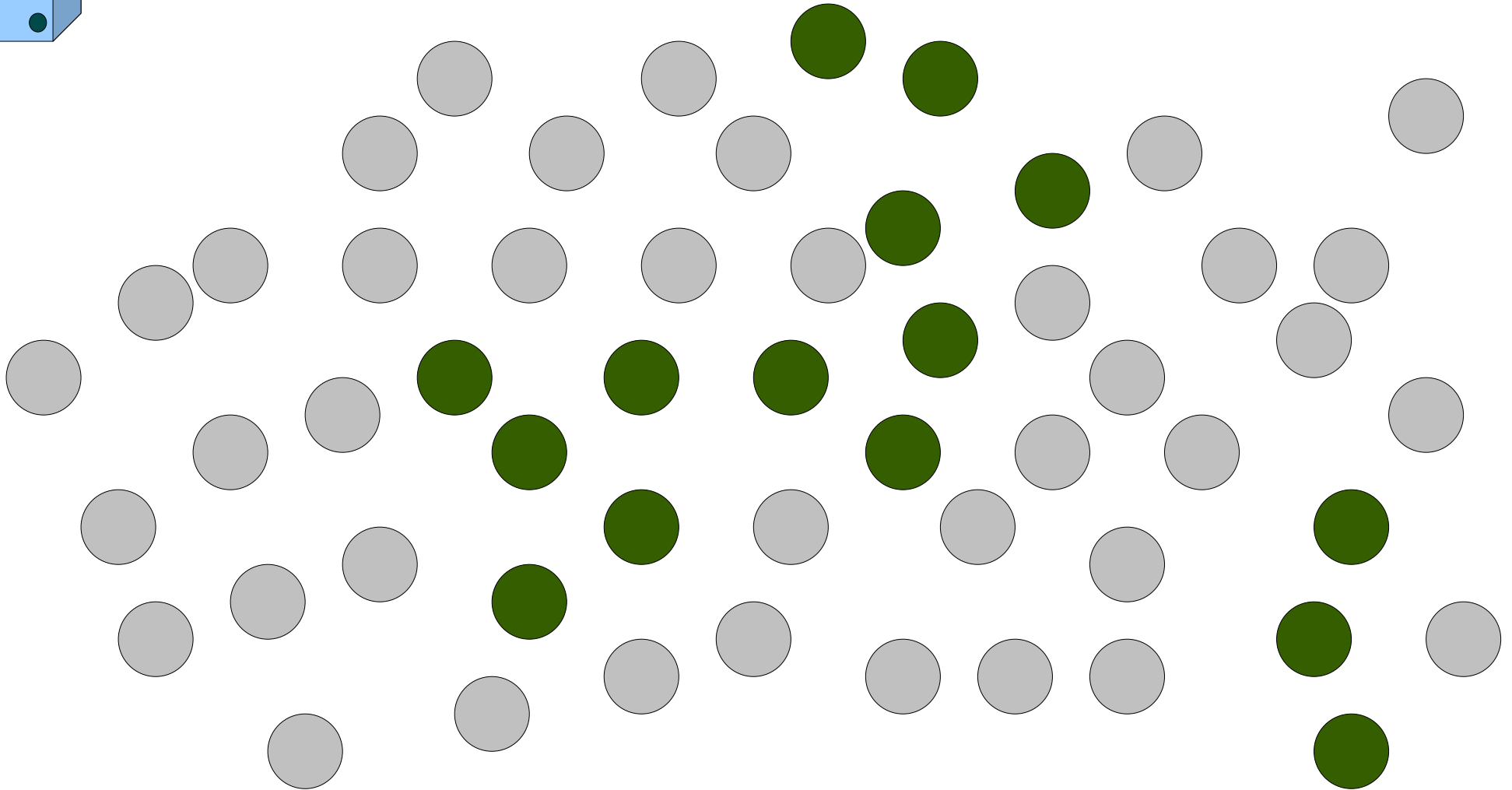


Toepassingen: Computer Vision. Machine code generatie.  
Planning en scheduling. Robot navigatie.

# Swarms.

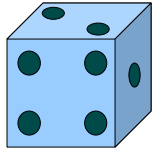


Wanneer is een bal groen?

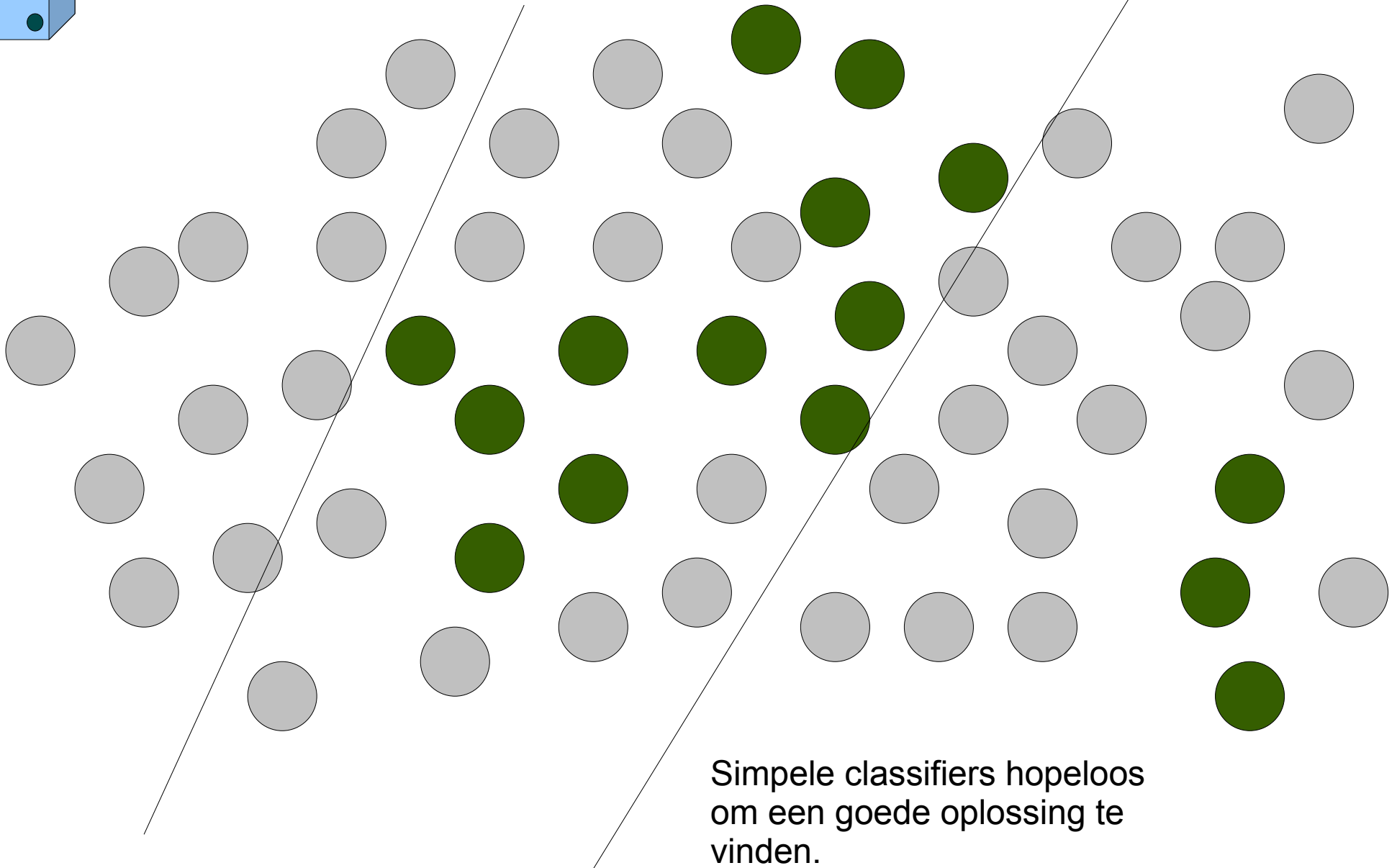




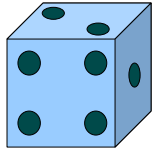
# Swarms.



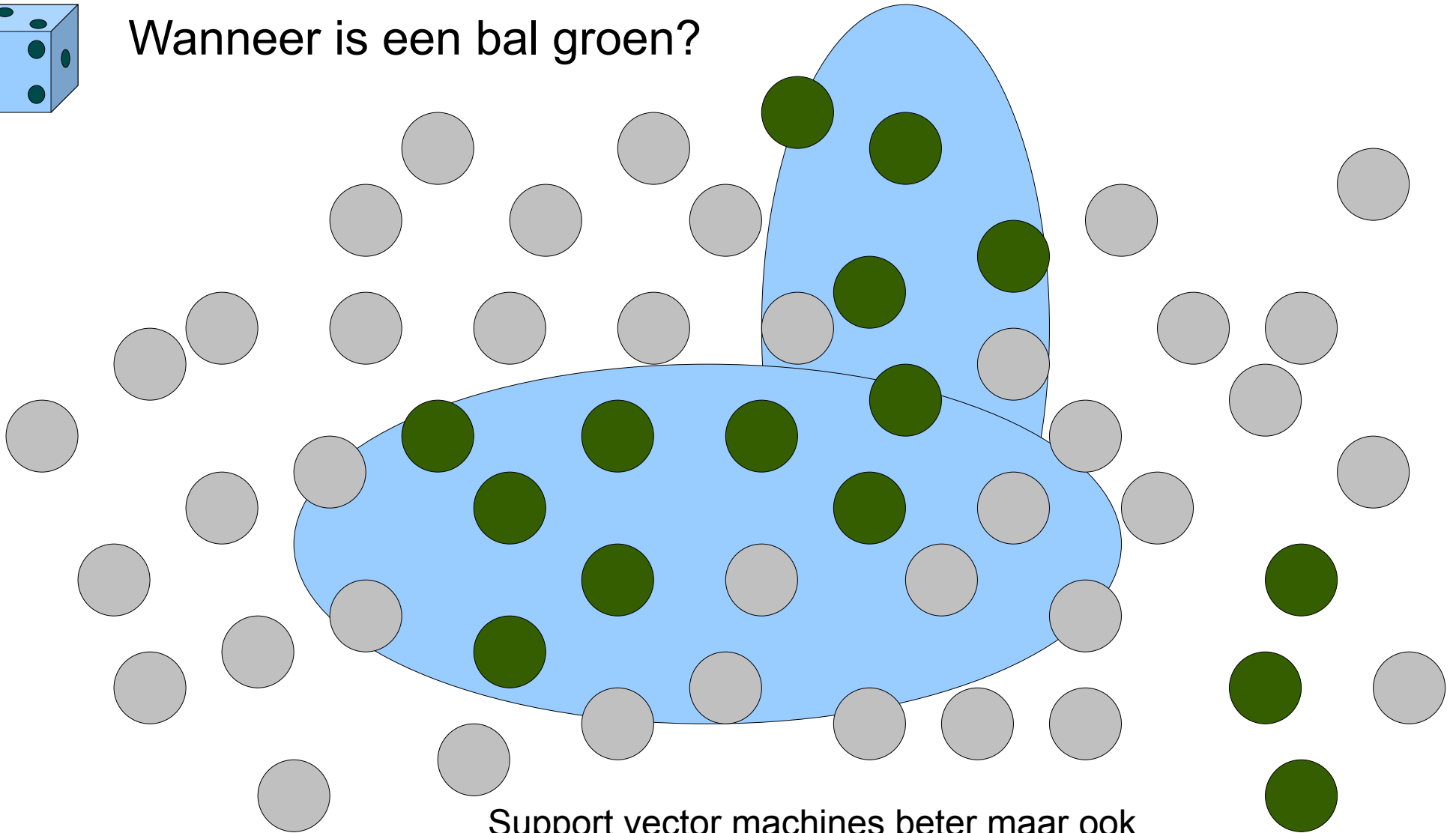
Wanneer is een bal groen?



# Swarms.

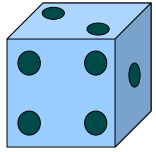


Wanneer is een bal groen?

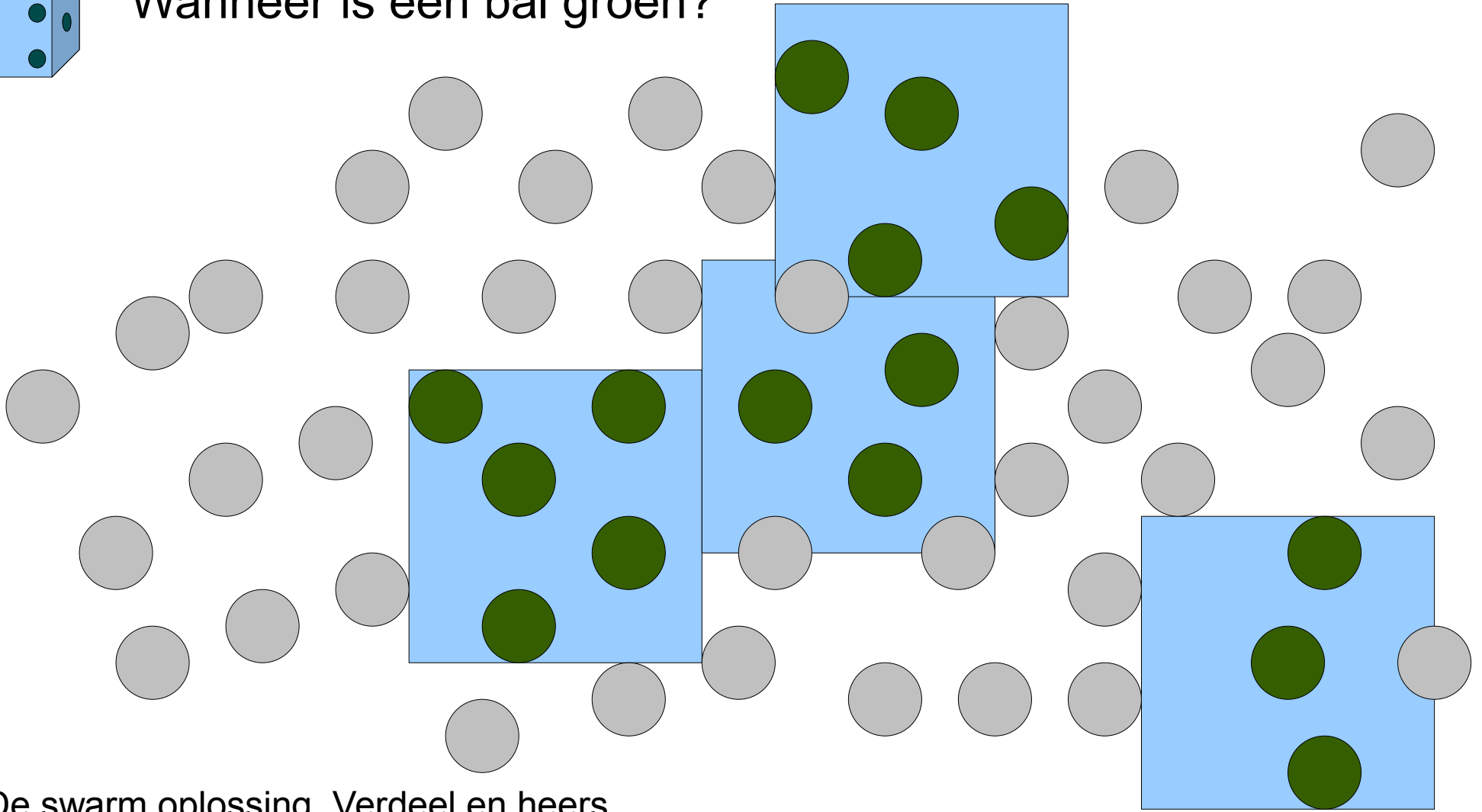


Support vector machines beter maar ook valse hits. Zal veel moeite kosten en veel vectors om een goed resultaat te bereiken.

# Swarms.



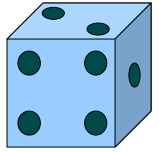
Wanneer is een bal groen?



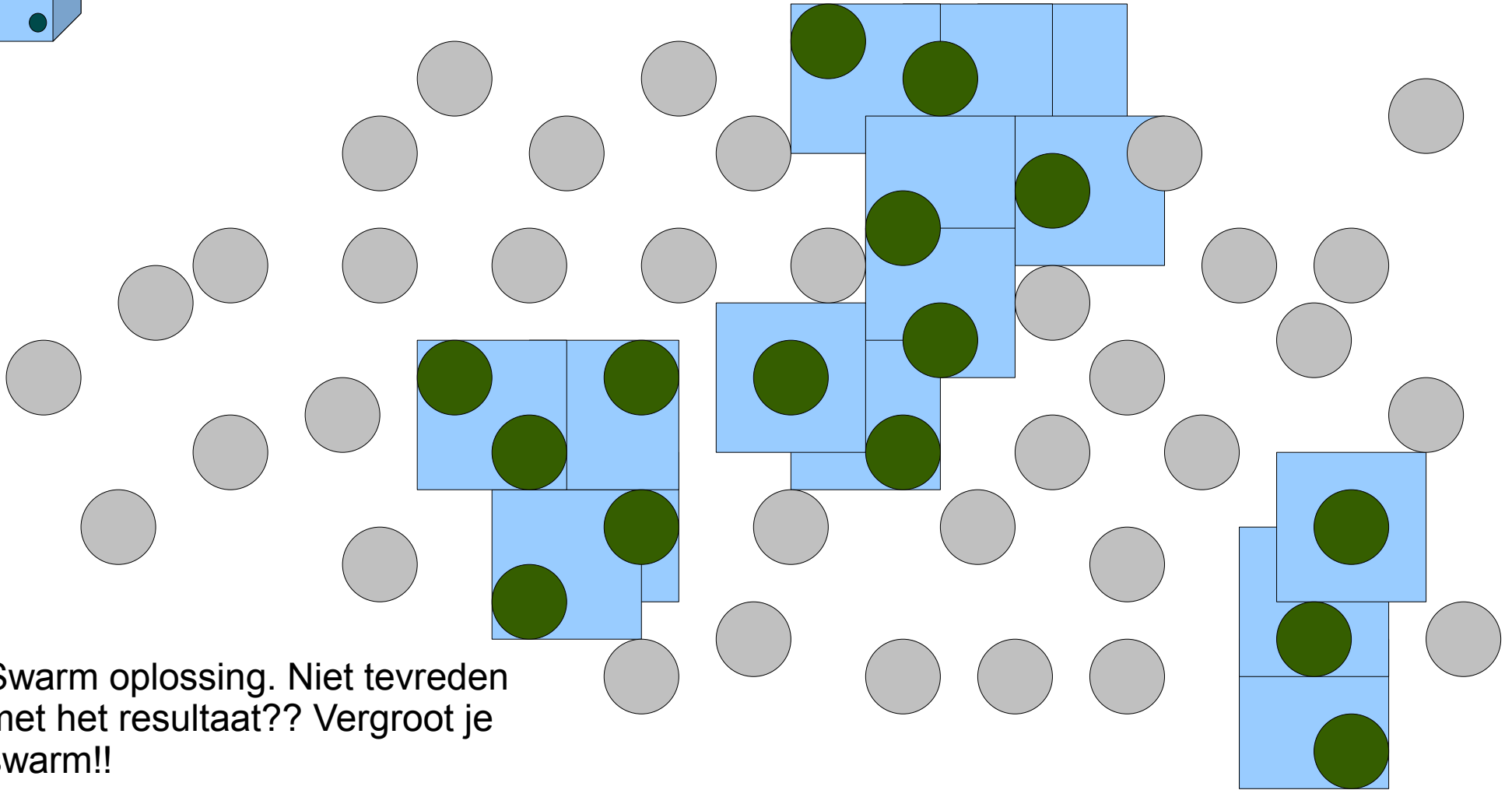
De swarm oplossing. Verdeel en heers.  
Splits probleem op in kleine brokjes en  
iedere klassifier is simpel (rechthoekje) en  
heeft nauwelijks false hits.

Recht hoekjes willekeurig gegenereerd.  
Daarna getest op “groenigheid”!

# Swarms.



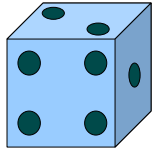
Wanneer is een bal groen?



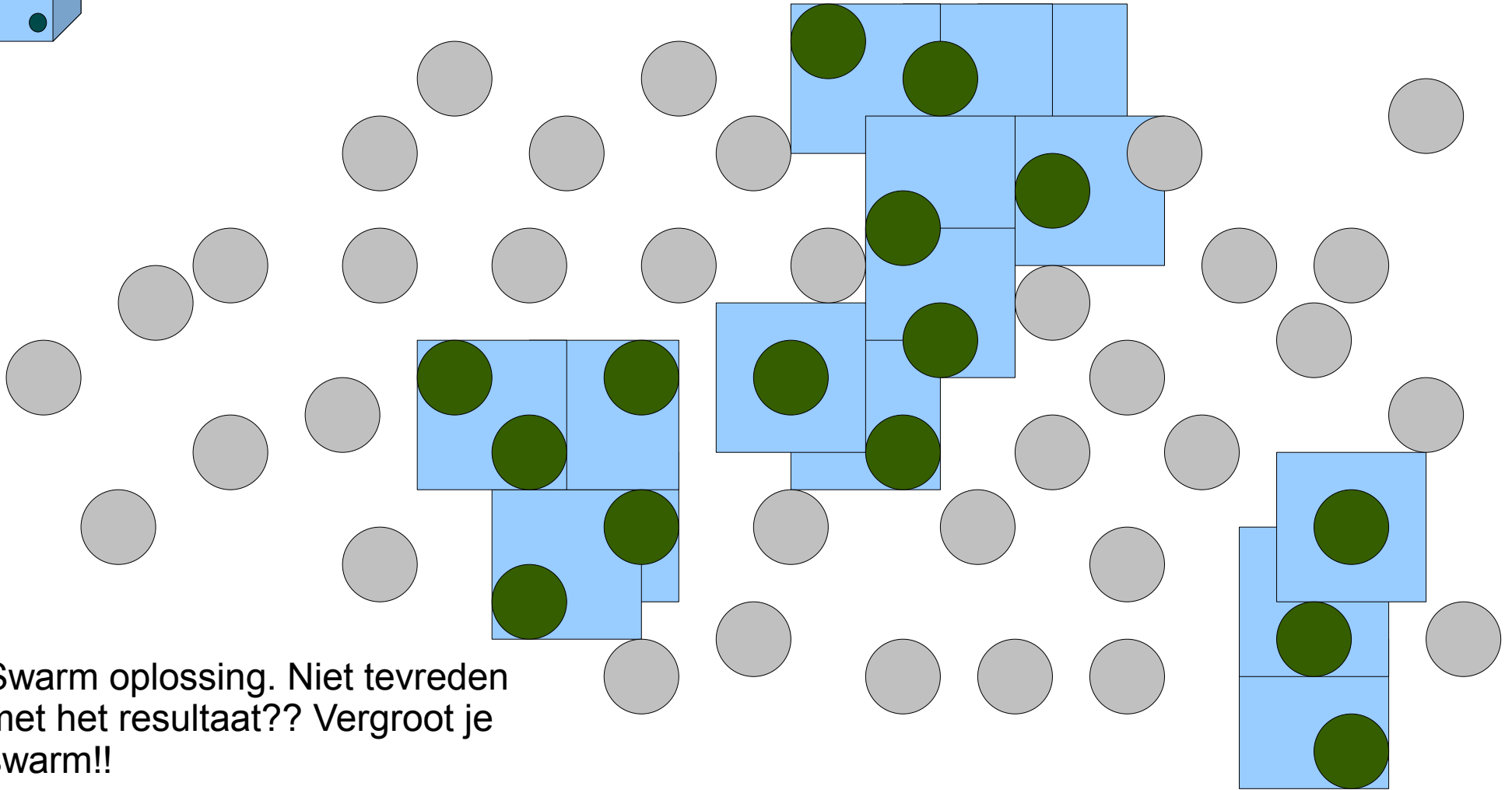
Swarm oplossing. Niet tevreden met het resultaat?? Vergroot je swarm!!

Overlap geen probleem. Meer rechthoekjes voor een groene punt betekent betere herkenning! Robuuste oplossing. Valt een detector (rechthoekje) uit dan blijft het toch goed werken.

# Swarms.



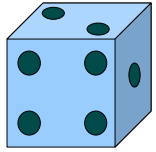
Wanneer is een bal groen?



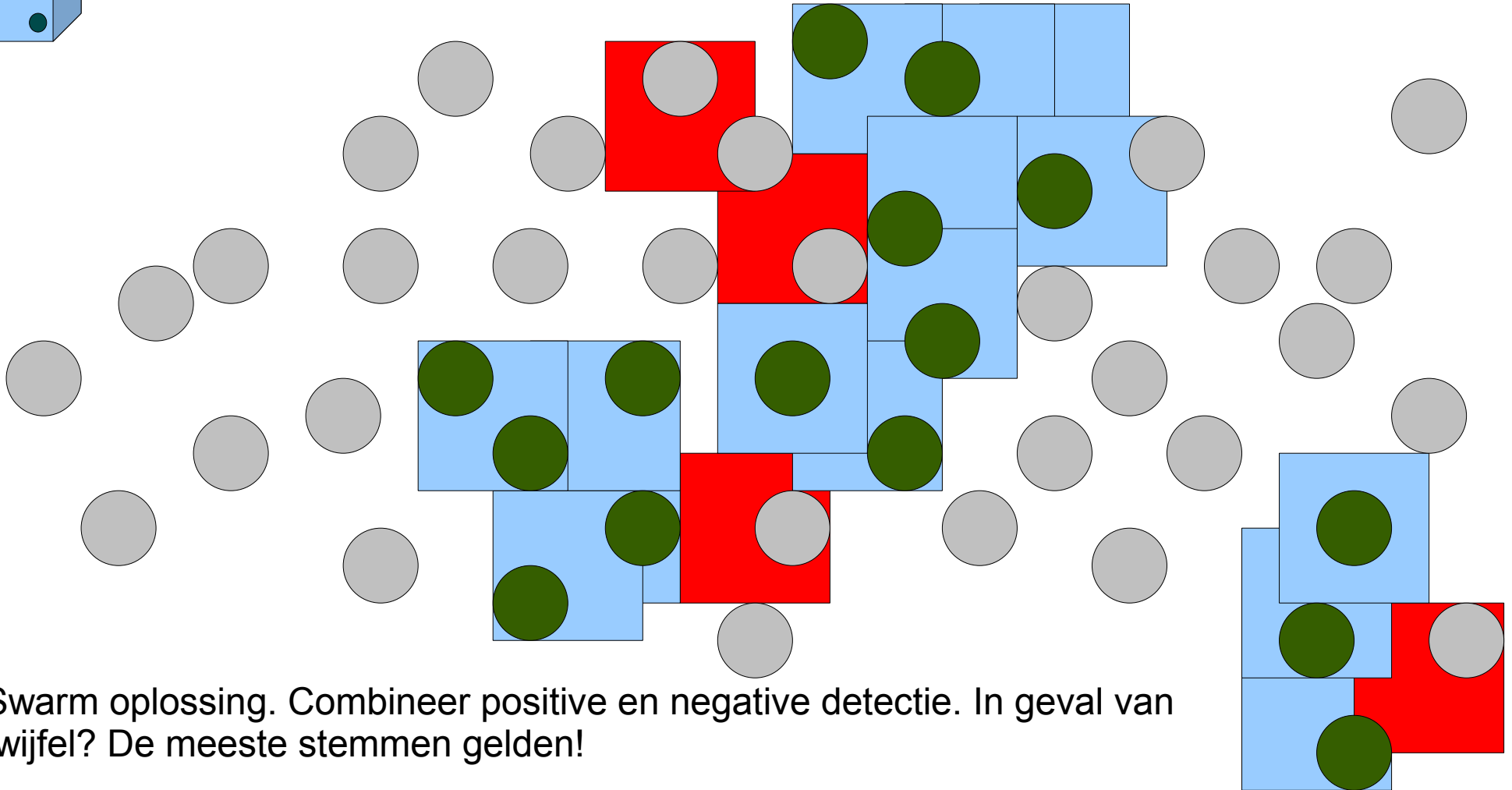
Swarm oplossing. Niet tevreden met het resultaat?? Vergroot je swarm!!

Overlap geen probleem. Meer rechthoekjes voor een groene punt betekent betere herkenning! Robuuste oplossing. Valt een detector (rechthoekje) uit dan blijft het toch goed werken.

# Swarms.



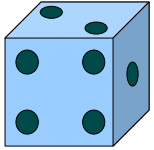
Wanneer is een bal groen?



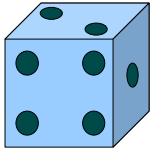
Swarm oplossing. Combineer positive en negative detectie. In geval van twijfel? De meeste stemmen gelden!

Door de verhouding van positief en negatief te meten heb je ook feedback over de kwaliteit van de herkenning. 99% positief dan bal bijna zeker groen. 51% positief? Waarschijnlijk groen, maar zeker is het niet.

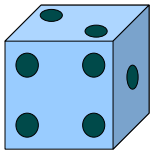
# Swarms. Makkelijk te maken



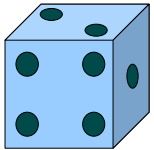
Genereer een collectie classifiers.



Selecteer de beste classifiers uit de groep.



Diversiteit zorgt ervoor dat zelfs slechte classifiers bruikbaar zijn. Meeste stemmen gelden.



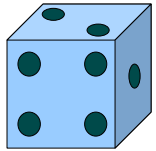
Voor binaire classificatie (2 uitkomsten, ja/nee) is slechts 10% goed gelijk aan 90% goed als je het antwoord inverteerd! Simpele boolean classifiers zijn goed genoeg en goedkoop in CPU cycles.



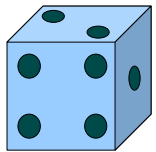
Onderschat de kracht van de swarm niet.  
Robuust en simpel te implementeren.  
CPU power is tegenwoordig relatief “goedkoop”.

# Multi-agents.

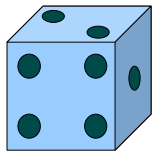
## Stel je vraag aan de juiste instantie



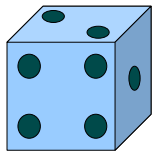
Ook bekend als “Network of experts”  
Staat specialisatie van agents toe.



Voorbeelden: expert systemen, biologisch Brein



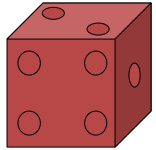
- Genereer meerdere oplossingen of antwoorden.
- Kies juiste antwoord uit gegeven antwoorden.
- Of kies degene die het meest zekere is van zijn antwoord.



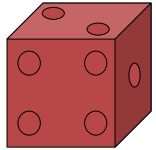
Kan in een hierarchie werken. Zodat de keuze wie antwoord geeft komt van een andere groep agents.



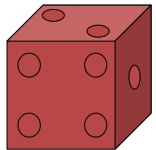
# Overzicht



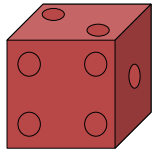
Zoeken?  
Parameter spaces. Local Optima



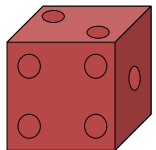
Hill climbing



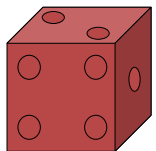
Simulated Annealing




GA – Genetic Algorithm  
EA – Evolutionary Algorithm



GP – Genetic Programming



Swarms  
Multi Agent systems



Hoe staat het  
met de demo?

