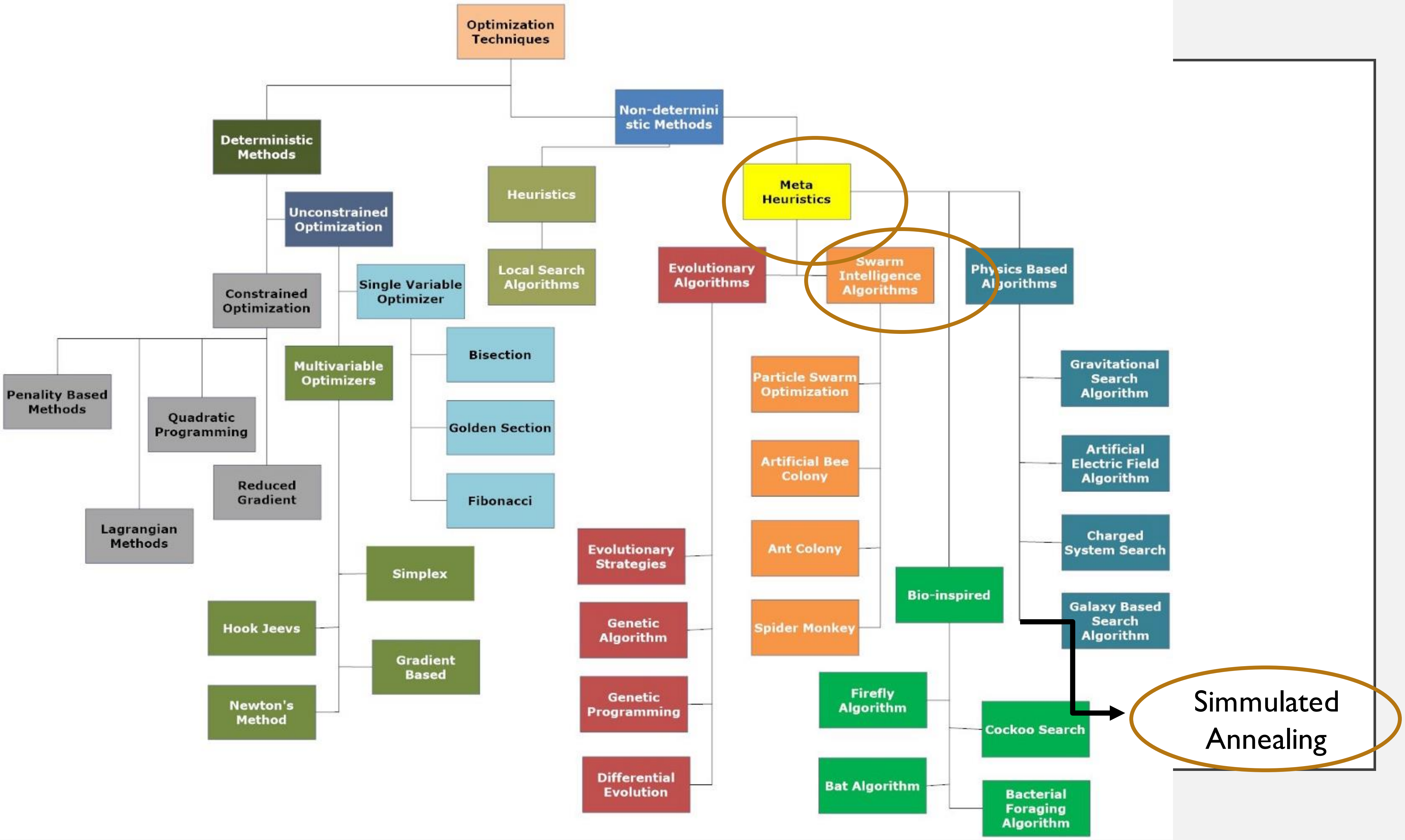
SIMULATED ANNEALING DAN SIMPLE PARTICLE SWARM

Classification of Optimization Techniques



SIMULATED ANNEALING DAN SIMPLE PARTICLE SWARM

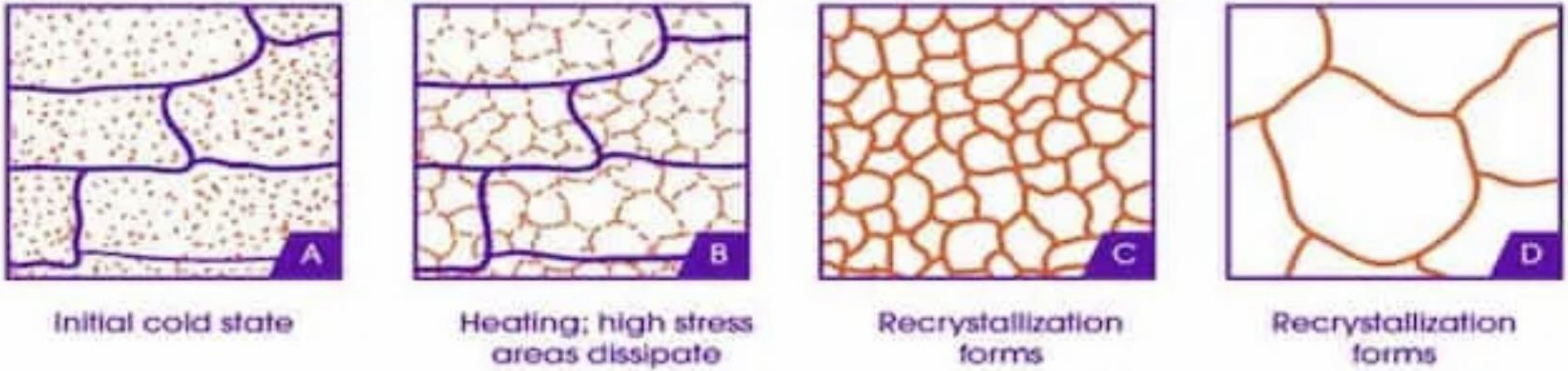
- Both methods are in the branch of Meta-heuristic optimization method.
- Simulated annealing was inspired from a thermal process for obtaining low energy states of a solid in a heat bath (Physic based algorithm)
- The process of simulated annealing contains two steps:
  1. Increase the temperature of the heat bath to a maximum value at which the solid melts.
  2. Decrease carefully the temperature of the heat bath until the particles arrange themselves in the ground state of the solid. Ground state is a minimum energy state of the solid.
- The ground state of the solid is obtained only if the maximum temperature is high enough and the cooling is done slowly.

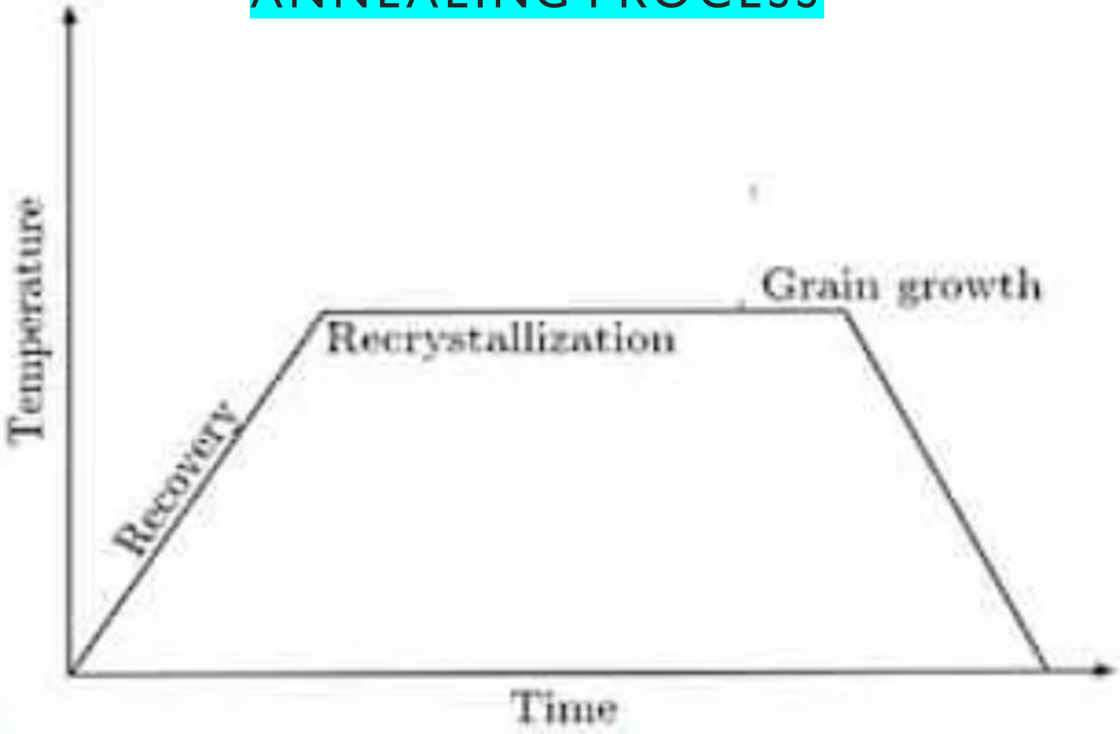
ANALOGY OF PHYSICAL ANNEALING

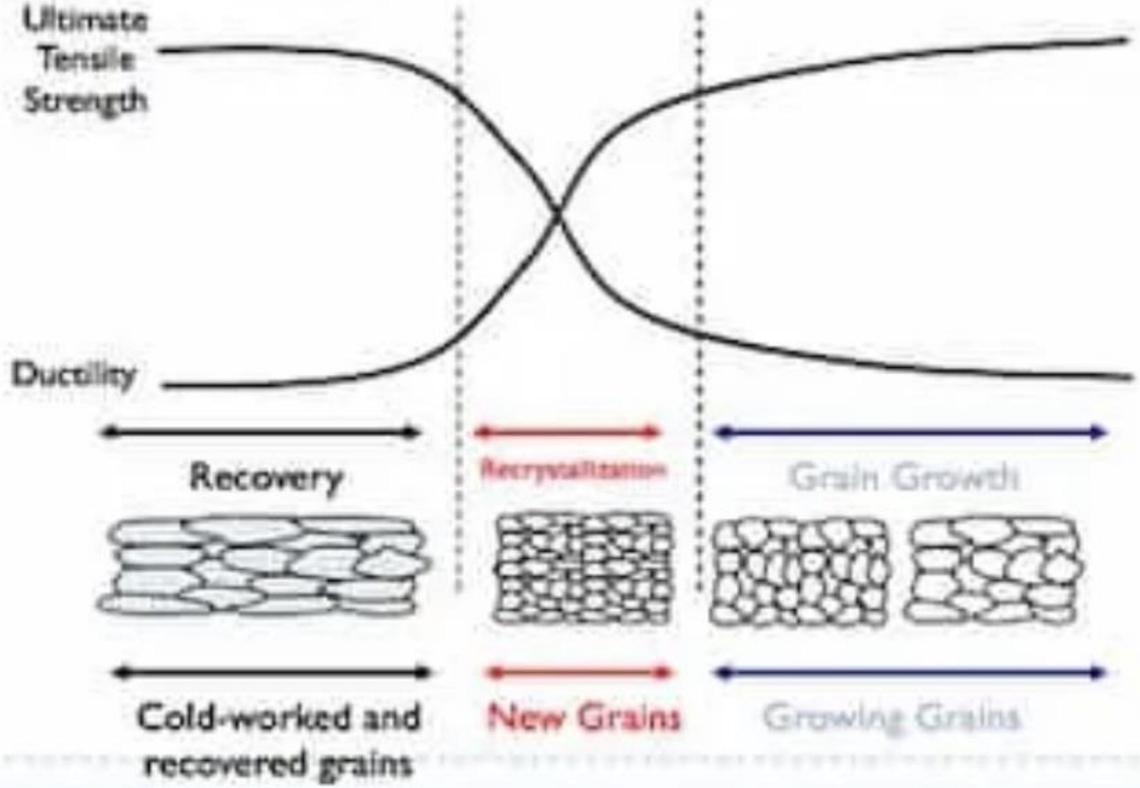
- Based on the cooling of material in a bath heat
- Returning a local search but still have a probability to move to another solution that probably better.

	Physical annealing process	Simulated annealing method
Problem involved	Metal in the heat bath	min $f(x)$ s.t. $x \in S$
Variable	Energy	Cost function
Purpose	Low energy state	Minimal value
Strategy	Cooling schedule	Algorithm

ANNEALING PROCESS

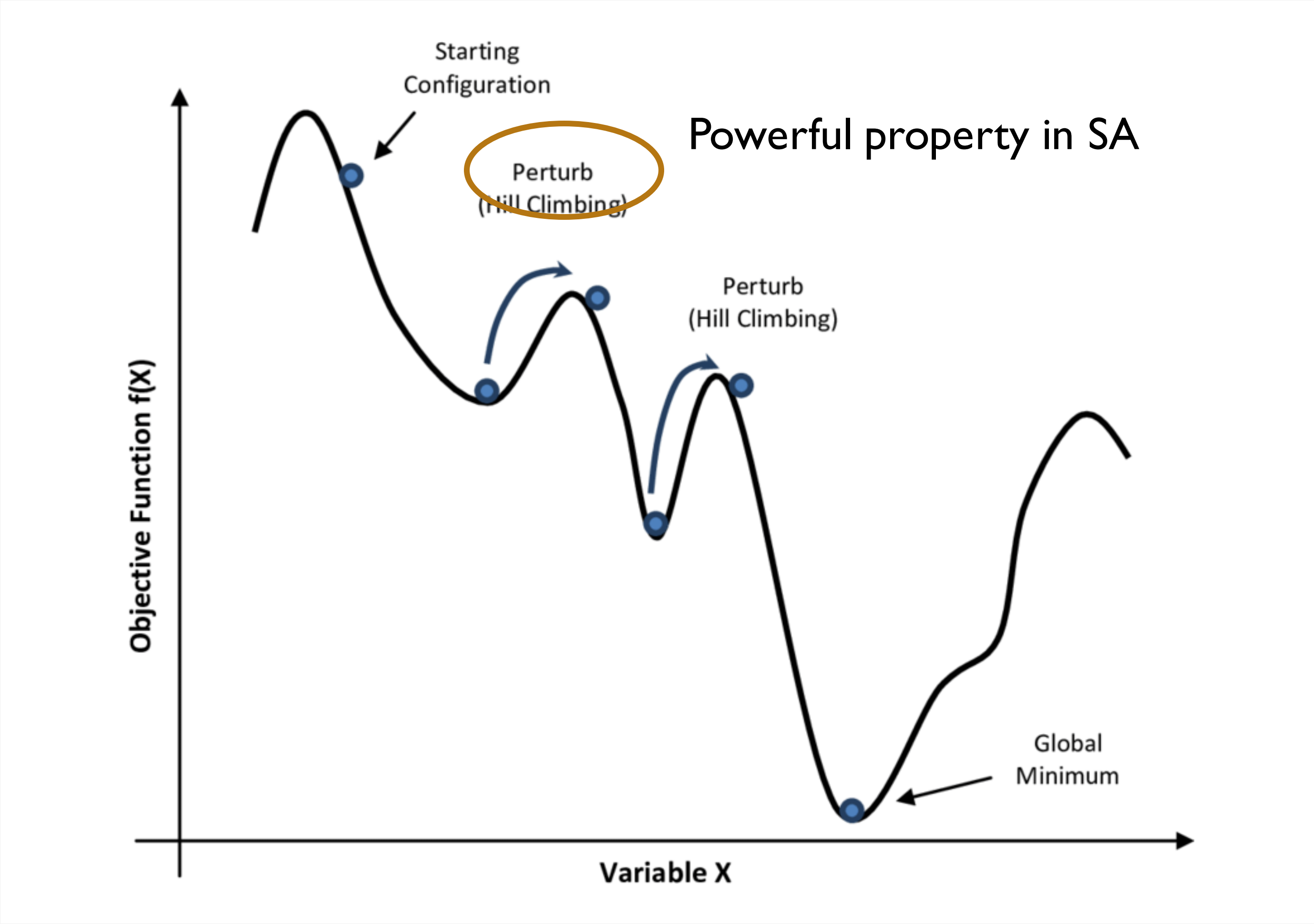






SIMULATED ANNEALING

- The goal is to find global solution obtained the minimum value of objective function



SIMULATED ANNEALING

- To apply simulated annealing with optimization purposes we require the following:
  - A successor function that returns a "close" neighboring solution given the actual one. This will work as the "disturbance" for the particles of the system.
  - A target function to optimize that depends on the current state of the system. This function will work as the energy of the system.
- The search is started with a randomized state. In a polling loop we will move to neighboring states always accepting the moves that decrease the energy while only accepting bad moves accordingly to a probability distribution dependent on the "temperature" of the system.

SIMULATED ANNEALING

- Decrease the temperature slowly, accepting less bad moves at each temperature level until at very low temperatures the algorithm becomes a greedy hill-climbing algorithm.

  - The distribution used to decide if we accept a bad movement is know as Boltzman distribution.

$$P(\gamma) = \frac{e^{-E_\gamma/T}}{Z(T)},$$

$$Z(T) = \sum_{\gamma'} e^{-E_{\gamma'}/T},$$

- This distribution is very well known is in solid physics and plays a central role in simulated annealing. Where γ is the current configuration of the system, E_γ is the energy related with it, and Z is a normalization constant.

ALGORITHM

```
Create random initial solution γ
E_old=cost(γ);
for(temp=temp_max; temp>=temp_min;temp=next_temp(temp) ) {
     for(i=0;i<i_max; i++ ) {
              succesor_func(γ); //this is a randomized function
              E_new=cost(γ);
              delta=E_new-E_old;
              if(delta>0)
                       if(random() >= exp(-delta/(K_b*temp));
                                undo_func(γ); //rejected bad move
                       else
                                E_old=E_new  //accepted bad move
              else
                       E_old=E_new; //always accept good moves
     }
}
```

K_b= **5,6704×10^{-5}**

EXAMPLE

- Travelling sales Problem (TSP)
- Given a set of location, find the shortest path!
- What is the cost function/energy?
  - Total travel distance through all cities.
- What is the variable/solution candidates?
  - The path or sequence of cities
- Is there a constraint must be satisfied?
  - Must it start and end in the same or a different city?

https://colab.research.google.com/drive/1kBHHamo_G-phsjApFYTwFDGQOsVEjdo2?usp=sharing

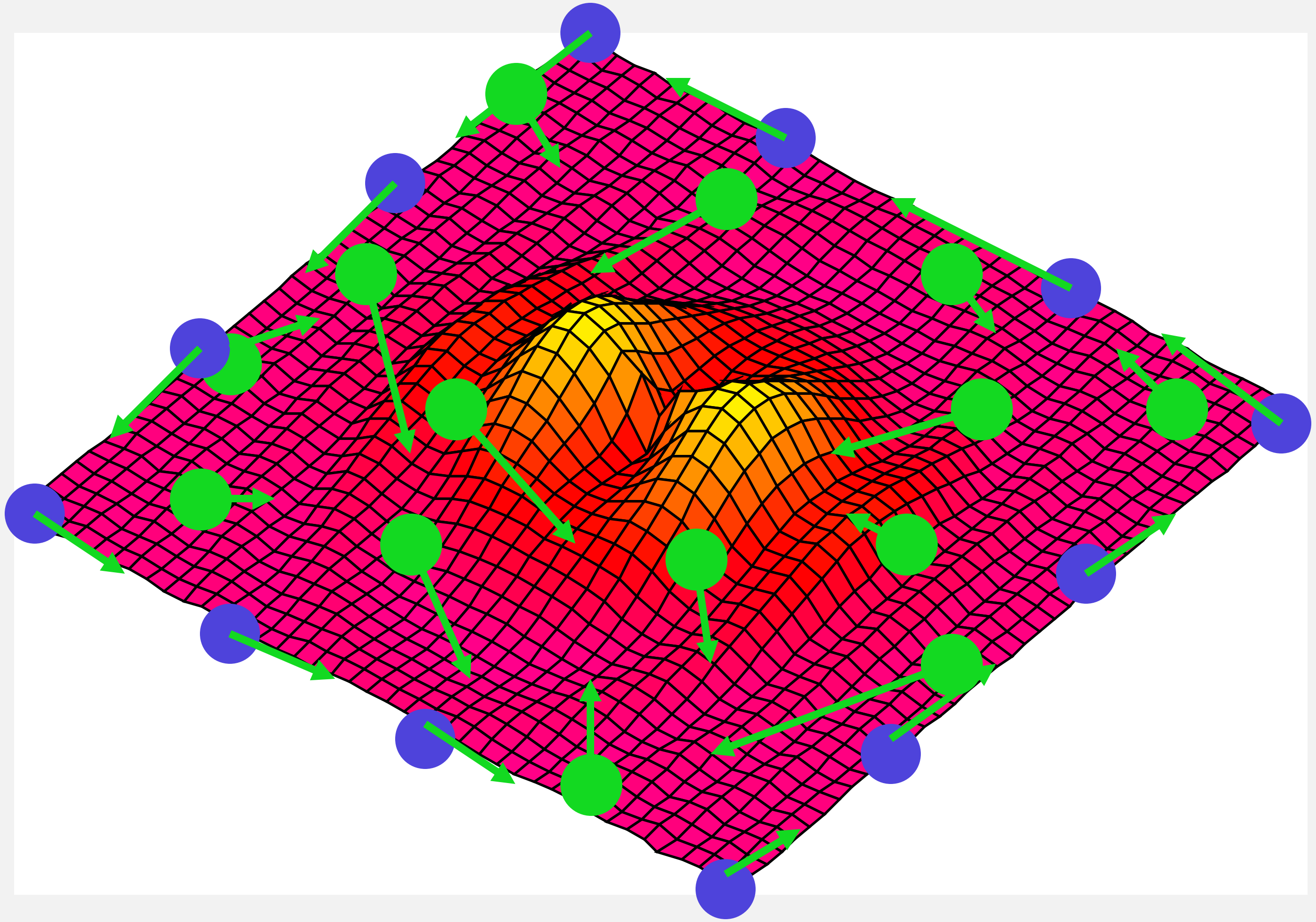
PARTICLE SWARM OPTIMIZATION

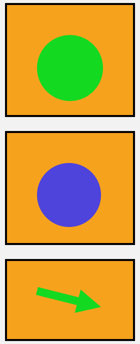
- Each particle is searching for the optimum
- Each particle is moving and hence has a velocity.
- Each particle remembers the position it was in where it had its best result so far (its personal best)

PARTICLE SWARM OPTIMIZATION

- The particles in the swarm co-operate. They exchange information about what they've discovered in the places they have visited
- The co-operation is very simple. In basic PSO it is like this:
  - A particle has a neighbourhood associated with it.
  - A particle knows the fitnesses of those in its neighborhood, and uses the position of the one with best fitness.
  - This position is simply used to adjust the particle's velocity

INITIALIZATION. POSITIONS AND VELOCITIES





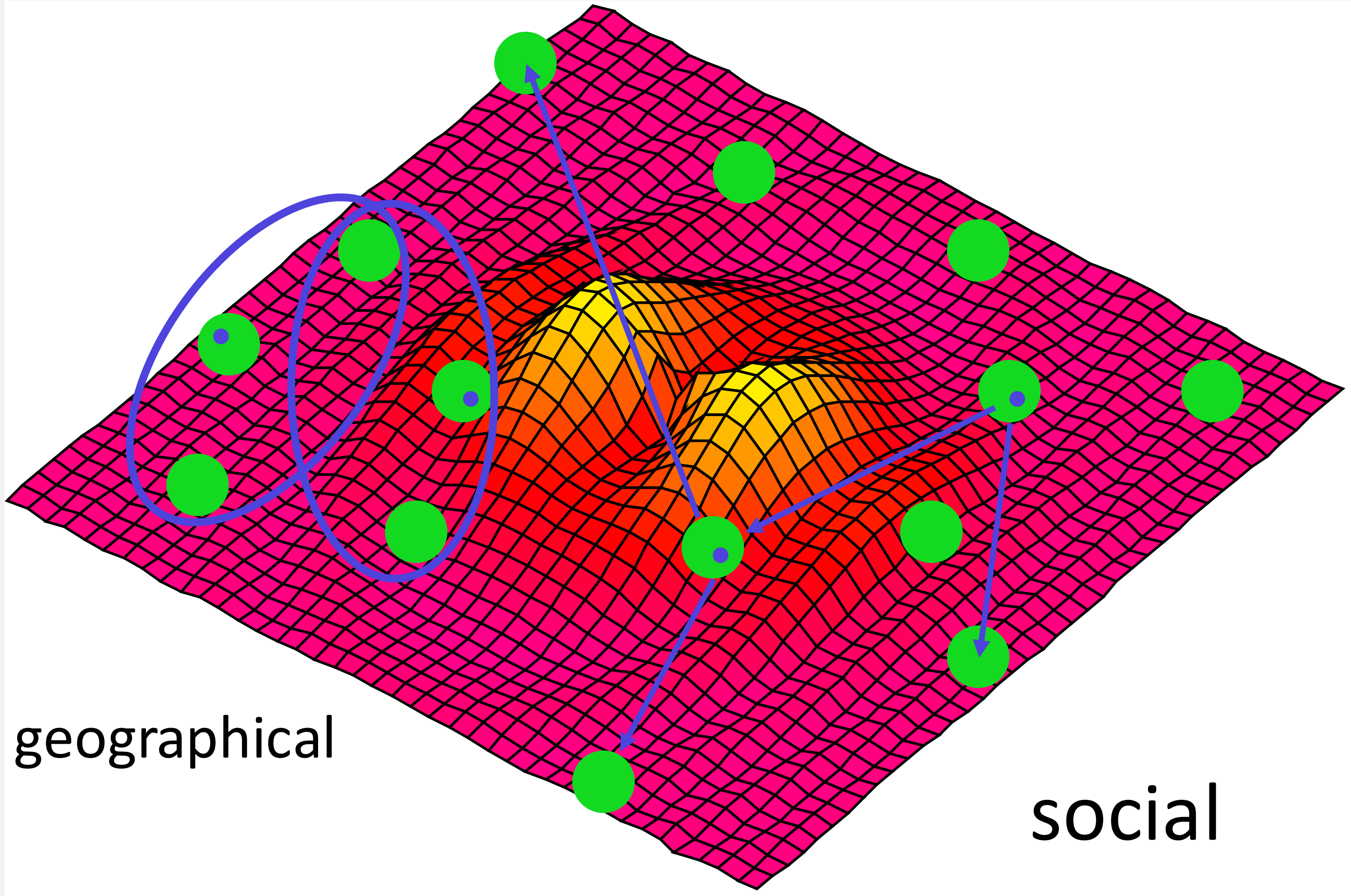
WHAT A PARTICLE DOES

In each timestep, a particle has to move to a new position. It does this by adjusting its *velocity*.

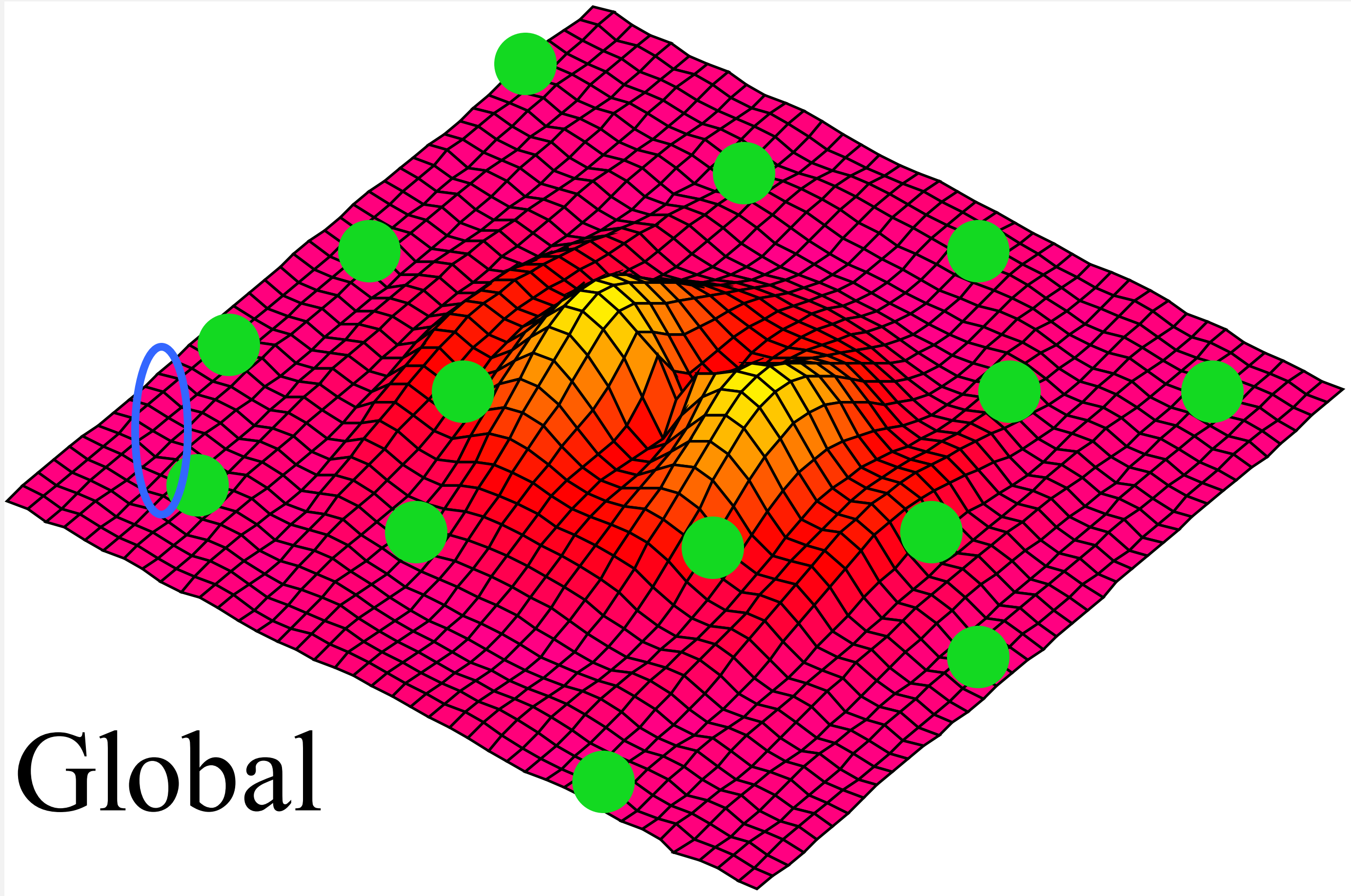
- *The adjustment is essentially this:*
- *The current velocity PLUS*
- *A weighted random portion in the direction of its personal best PLUS*
- *A weighted random portion in the direction of the neighbourhood best.*

Having worked out a new velocity, its position is simply its old position plus the new velocity.

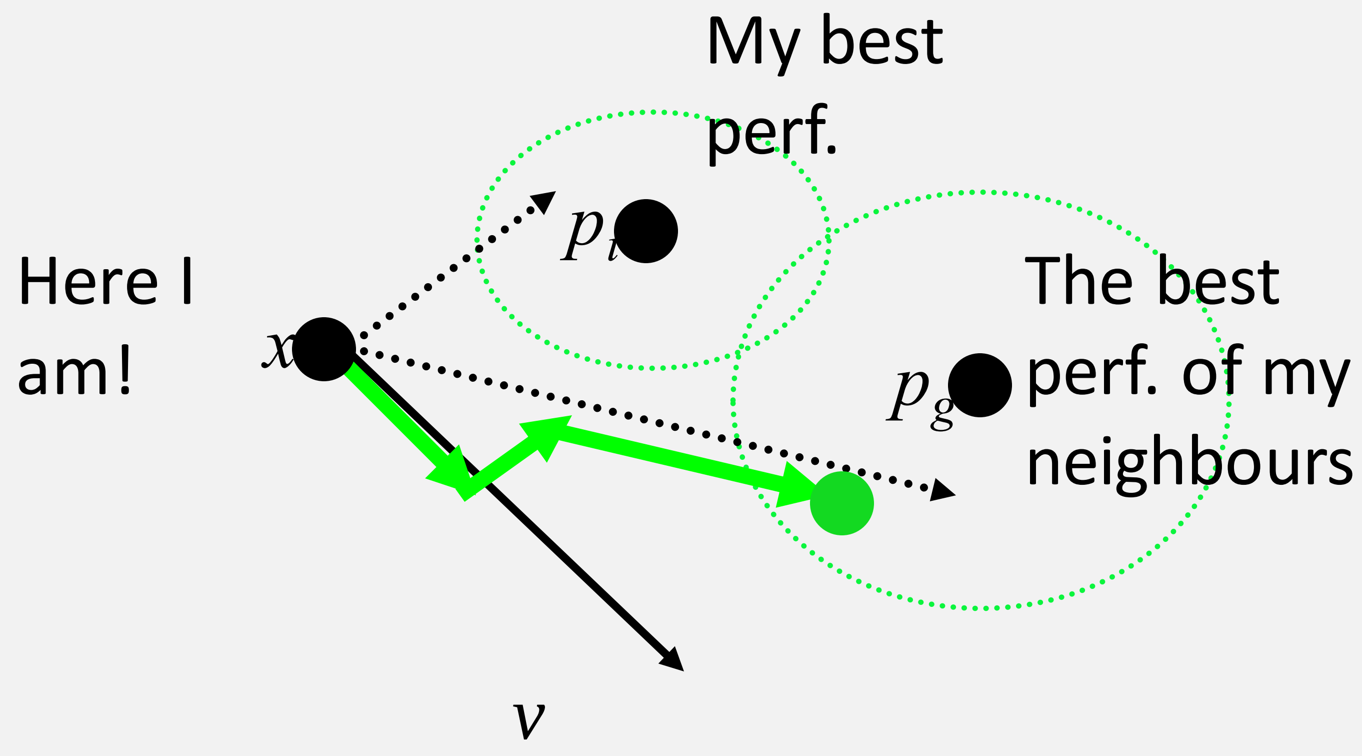
NEIGHBOURHOODS



NEIGHBOURHOODS



PARTICLES ADJUST THEIR POSITIONS ACCORDING TO A ``PSYCHOSOCIAL COMPROMISE'' BETWEEN WHAT AN INDIVIDUAL IS COMFORTABLE WITH, AND WHAT SOCIETY RECKONS



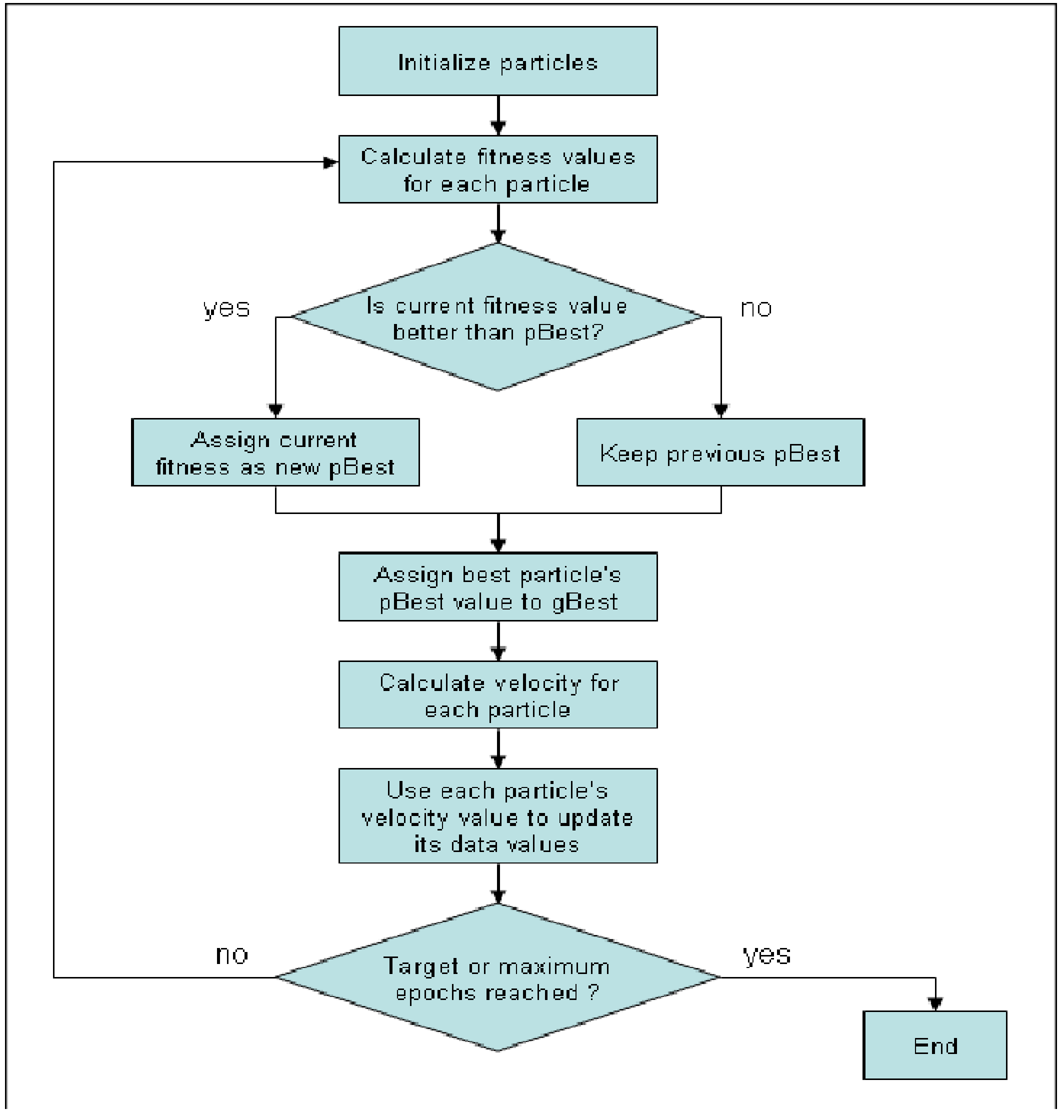
ALGORITHM

```
 For each particle
   Initialize particle
END

Do
   For each particle
       Calculate fitness value
       If the fitness value is better than its personal best
         set current value as the new pBest
   End

   Choose the particle with the best fitness value of all as gBest
   For each particle
       Calculate particle velocity according equation (a)
       Update particle position according equation (b)
   End
While maximum iterations or minimum error criteria is not attained
```

PSO ALGORITHM



PARAMETERS

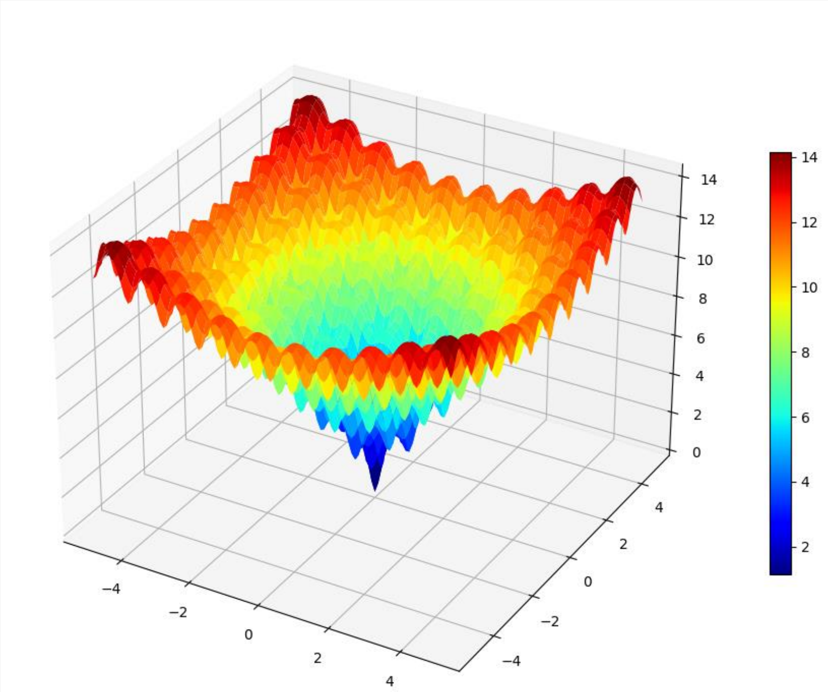
- Number of particles

  (10—50) are reported as usually sufficient.
- C1 (importance of personal best)
- C2 (importance of neighbourhood best)
- Usually C1+C2 = 4. No good reason other than empiricism
- Vmax – too low, too slow; too high, too unstable.

EXAMPLE PSO FOR MULTIMODAL FUNCTION IN R^2

The example to use PSO to find the minimum of this following function.

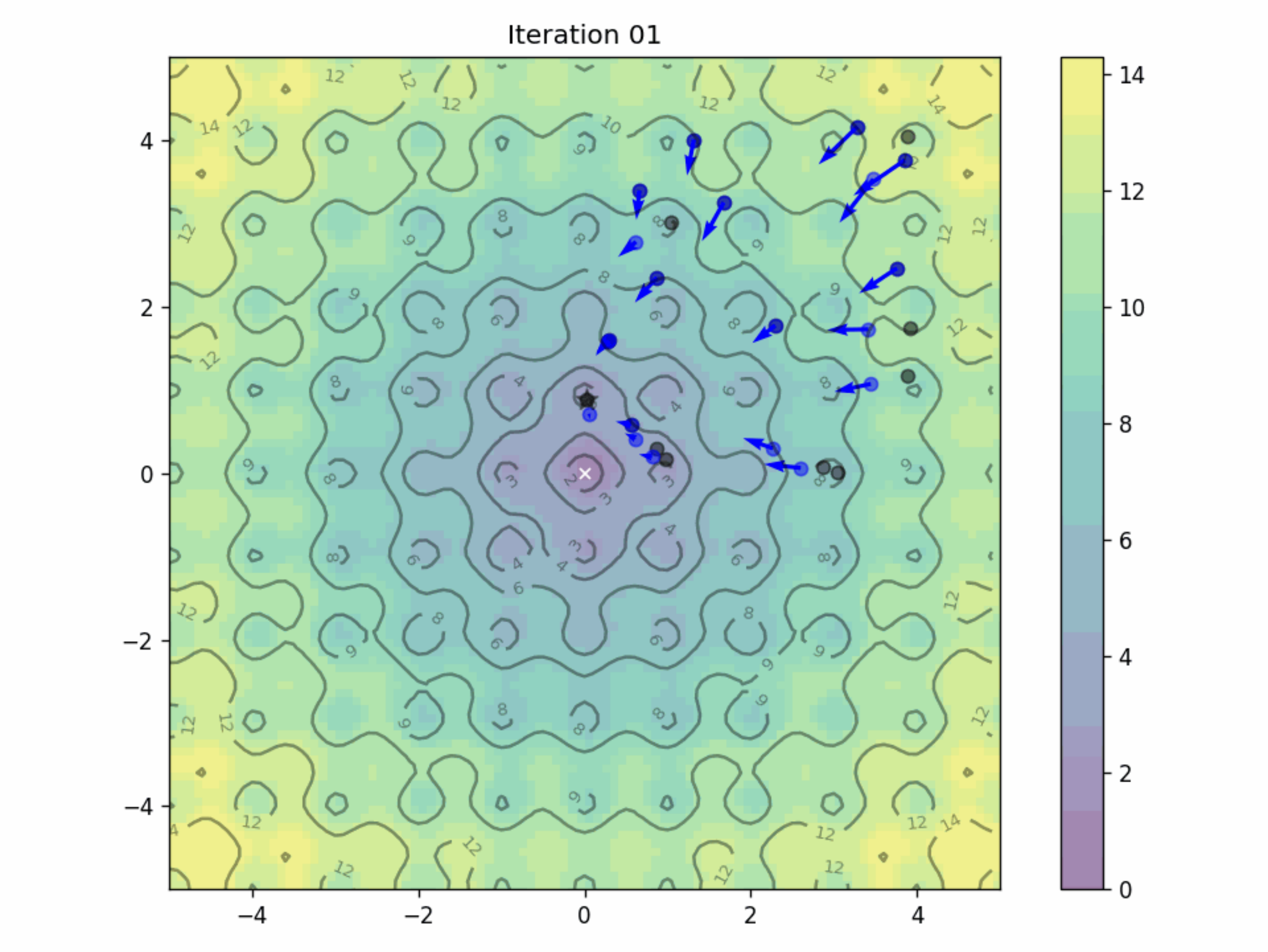
$$f(x,y) = -20e^{-0.2\sqrt{\frac{1}{2}(x^2+y^2)}} - e^{\frac{1}{2}(\cos 2\pi x+\cos 2\pi y)} + e + 20$$



Surface plot of z=f(x,y)

EXAMPLE PSO

Contour plot of f(x,y) and showing how the particles and the solution move to finally find the global minimum solution of z=f(x,y).



GROUP DISCUSSION
SOLVING TSP USING BOTH METHOD

1. Given a list of cities and their locations (usually specified as Cartesian co-ordinates on a plane), what is the shortest itinerary which will visit every city exactly once and return to the point of origin?
2. Generate **n** artificial locations as cities that we want to find the optimal itinerary of visiting those cities.
3. Use cartesian distance as the distance between two cities. Write the formula!
4. Determine how many possible paths can be generated to visit all cities starting and ending at the same location?
5. Decide and write the objective function to be minimized.
6. Use swarm particle optimization method and Simulated annealing to solve the optimization problem.
7. Write the algorithm using flowchart
8. Write the code equipped with several interesting plots: evolution of path during iteration, how the objective function decreasing during iteration and reach the optimum.
9. Write the report.

0. Judul dan tabel kontribusi setiap anggota.

1. Identifikasi masalah yang jelas.

2. Tujuan model dan simulasi.

3. Metode yang dijabarkan dengan jelas, berikut algoritma yang digunakan.

4. Hasil komputasi/simulasi model dan visulisasinya.

5. Analisis hasil / evaluasi.

6. Kesimpulan

7. Link lampiran program yang dapat diakses dan diperiksa dosen. Jika tidak bisa diakses dosen, maka laporan dianggap tidak lengkap yang tentunya akan mengurangi nilai.

Nama	NIM	Bobot kontribusi per jobdesk (%)								Total (max.100%)
		Poin-1	Poin-2	Poin-3	Poin-4	Poin-5	Poin-6	Poin-7	Tambahan (sebutkan)	

REFERENCES

- https://mat.uab.cat/~alseda/MasterOpt/IntroHO.pdf