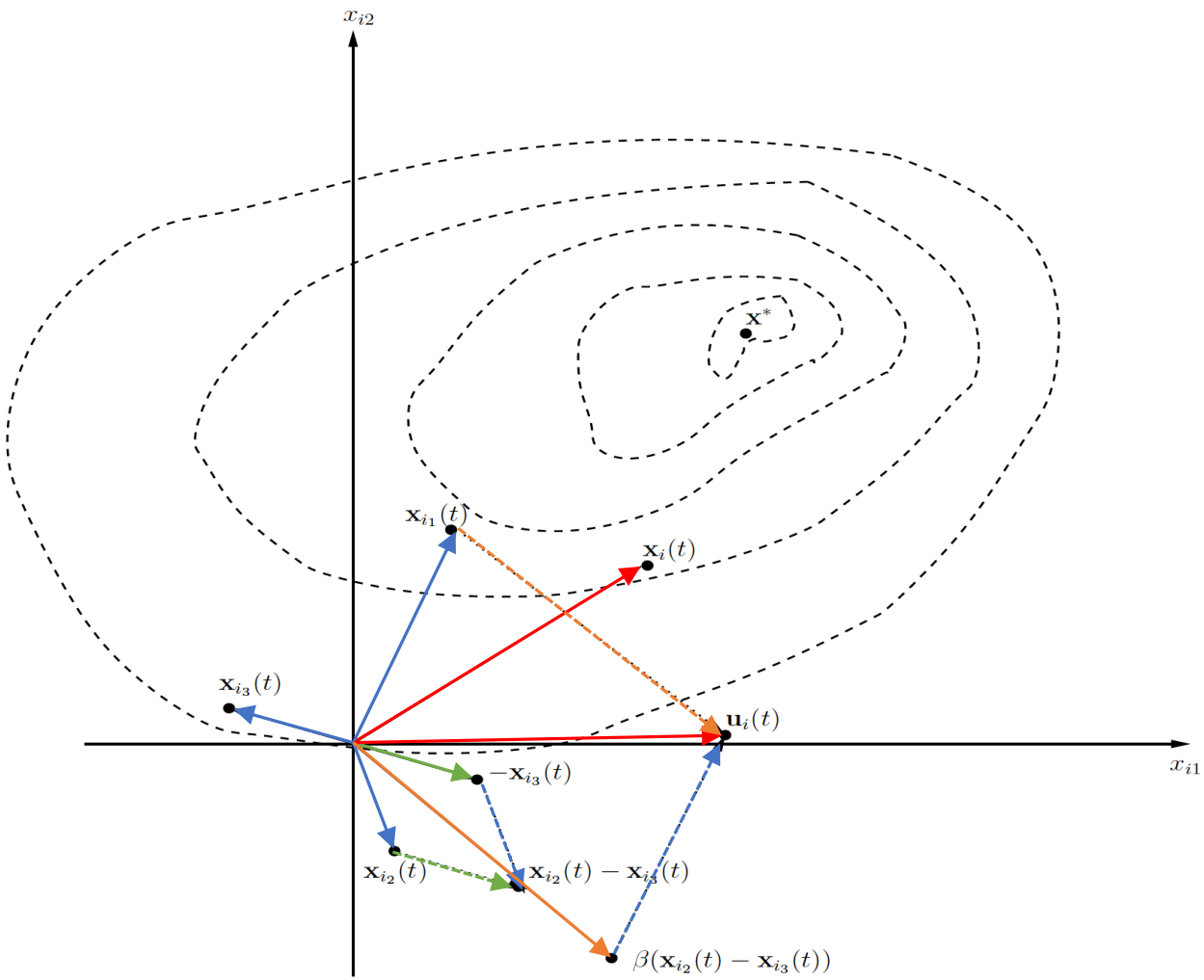
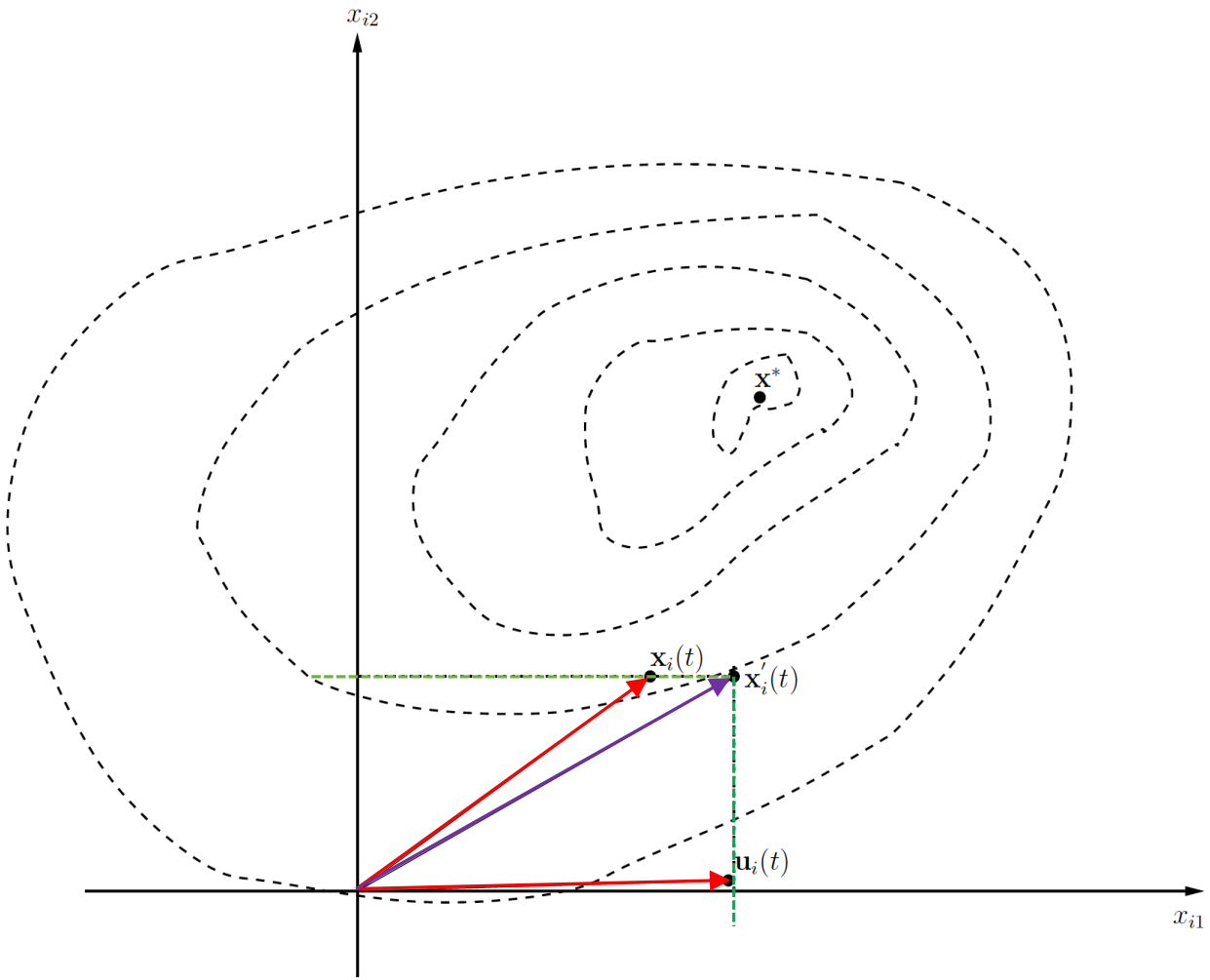
Algorithm 7.1 Prototypical Genetic Algorithm

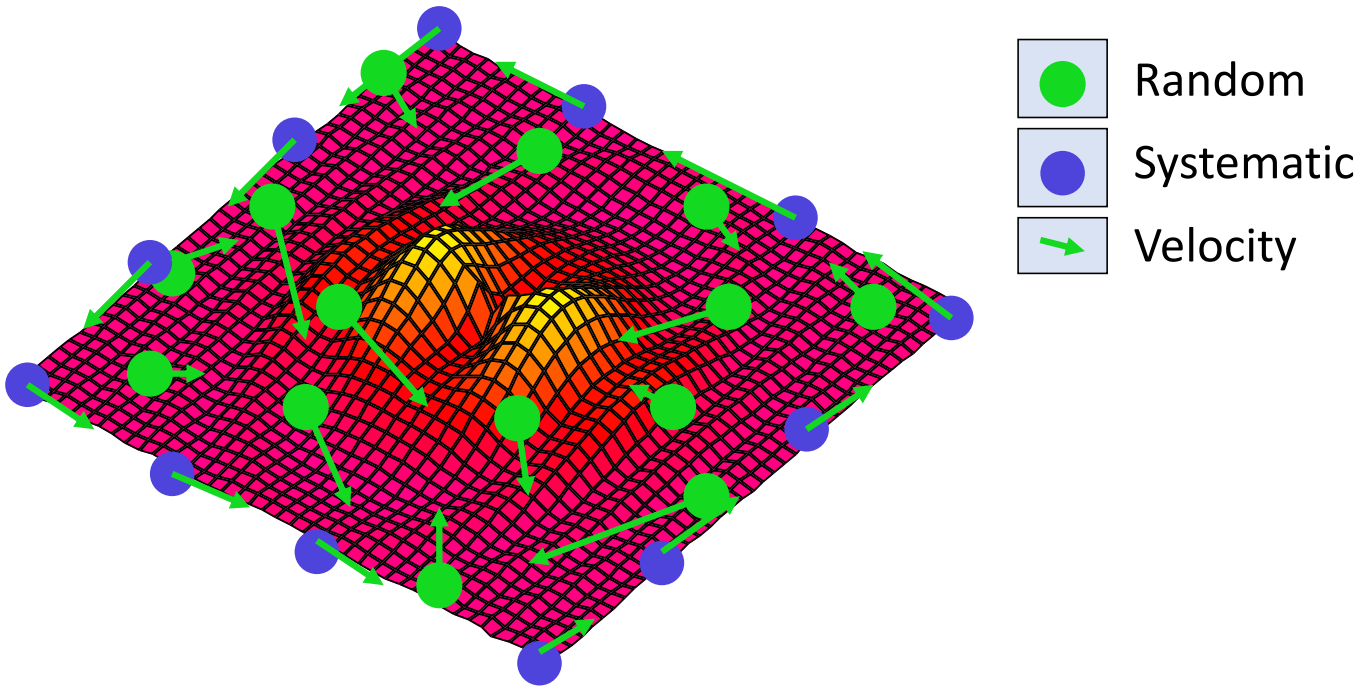
```
1  GeneticAlgorithm() {
2      t ← 0
3      initialize(P(t))
4      f(t) ← evaluateFitness(P(t))
5      while not(terminatep()) do {
6          t ← t + 1
7          C(t) ← select(P(t − 1))
8          C′(t) ← recombine(C(t))
9          C″(t) ← mutate(C′(t))
10         f(t) ← evaluateFitness(C″(t))
11         P(t) ← replace(P(t − 1), C″(t), f(t))
12     }
13     return P(t)
14 }
```

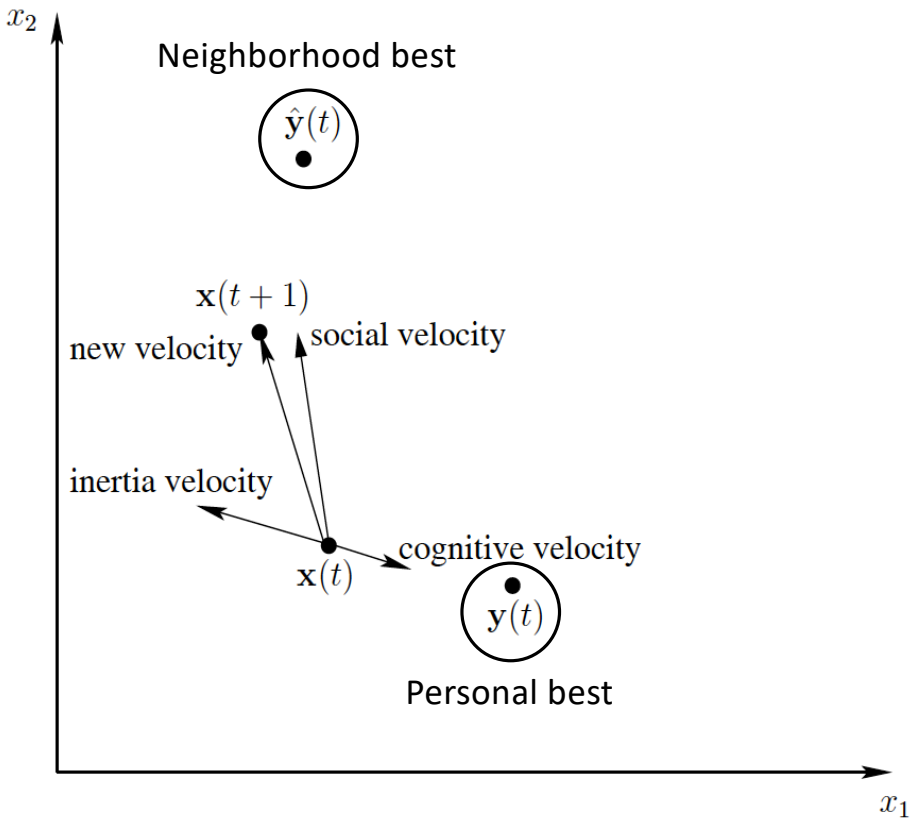
Algorithm 13.3 General Differential Evolution Algorithm

Set the generation counter, $t = 0$;
Initialize the control parameters, β and p_r;
Create and initialize the population, $\mathcal{C}(0)$, of n_s individuals;
while *stopping condition(s) not true* **do**
    for *each individual,* $\mathbf{x}_i(t) \in \mathcal{C}(t)$ **do**
        Evaluate the fitness, $f(\mathbf{x}_i(t))$;
        Create the trial vector, $\mathbf{u}_i(t)$ by applying the mutation operator;
        Create an offspring, $\mathbf{x}_i^{'}(t)$, by applying the crossover operator;
        if $f(\mathbf{x}_i^{'}(t))$ *is better than* $f(\mathbf{x}_i(t))$ **then**
            Add $\mathbf{x}_i^{'}(t)$ to $\mathcal{C}(t+1)$;
        end
        else
            Add $\mathbf{x}_i(t)$ to $\mathcal{C}(t+1)$;
        end
    end
end
Return the individual with the best fitness as the solution;

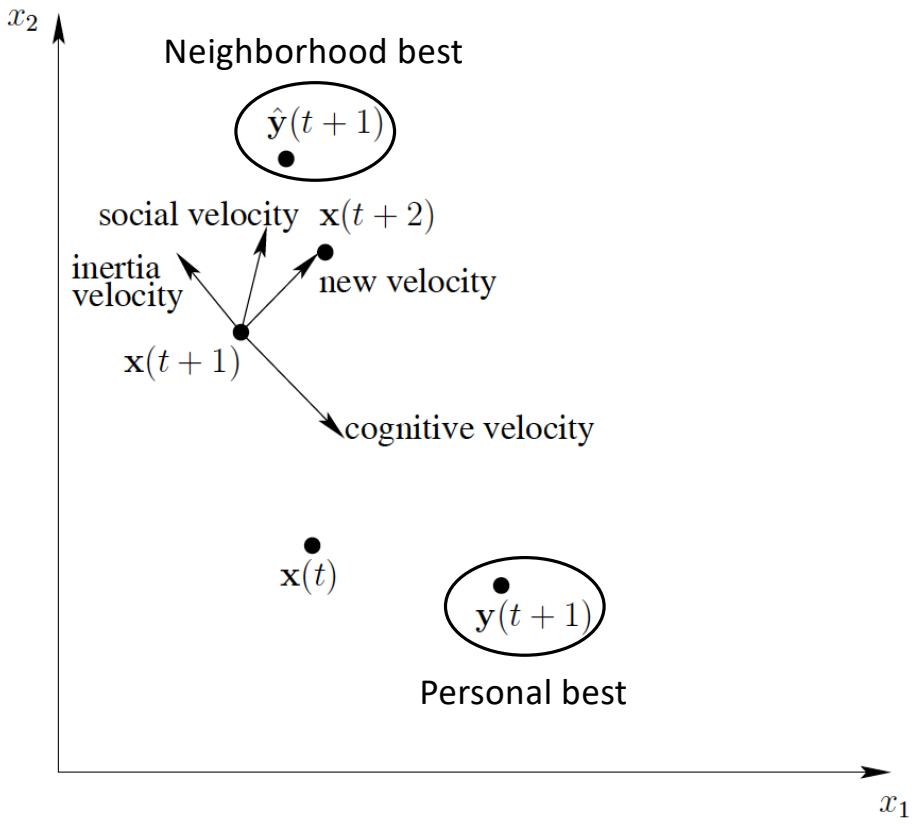

x_{i2}
$\mathbf{x}^*$
$\mathbf{x}_{i_1}(t)$
$\mathbf{x}_i(t)$
$\mathbf{x}_{i_3}(t)$
$\mathbf{u}_i(t)$
x_{i1}
$-\mathbf{x}_{i_3}(t)$
$\mathbf{x}_{i_2}(t)$
$\mathbf{x}_{i_2}(t) - \mathbf{x}_{i_3}(t)$
$\beta(\mathbf{x}_{i_2}(t) - \mathbf{x}_{i_3}(t))$


x_{i2}
$\mathbf{x}^*$
$\mathbf{x}_i(t)$
$\mathbf{x}_i'(t)$
$\mathbf{u}_i(t)$
x_{i1}


Random
Systematic
Velocity



(a) Time Step t



(b) Time Step $t+1$

Algorithm 16.1 *gbest* PSO

Create and initialize an n_x-dimensional swarm;
repeat
  for *each particle* $i = 1, \ldots, n_s$ **do**
    //set the personal best position
    if $f(\mathbf{x}_i) < f(\mathbf{y}_i)$ **then**
      $\mathbf{y}_i = \mathbf{x}_i$;
    end
    //set the global best position **if** $f(\mathbf{y}_i) < f(\hat{\mathbf{y}})$ **then**
      $\hat{\mathbf{y}} = \mathbf{y}_i$;
    end
  end
  for *each particle* $i = 1, \ldots, n_s$ **do**
    update the velocity using equation (16.2);
    update the position using equation (16.1);
  end
until *stopping condition is true*;

$$
\begin{aligned}
v_{ij}(t+1) &= v_{ij}(t) \\
&+ c_1 r_{1j}(t)[pb_{ij}(t) - x_{ij}(t)] \\
&+ c_2 r_{2j}(t)[gb_{ij}(t) - x_{ij}(t)]
\end{aligned}
$$

$$
x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)
$$

Algorithm 16.2 *lbest* PSO

Create and initialize an n_x-dimensional swarm;
repeat
  for *each particle* $i = 1, \ldots, n_s$ **do**
    //set the personal best position
    if $f(\mathbf{x}_i) < f(\mathbf{y}_i)$ **then**
      $\mathbf{y}_i = \mathbf{x}_i$;
    end
    //set the neighborhood best position
    if $f(\mathbf{y}_i) < f(\hat{\mathbf{y}}_i)$ **then**
      $\hat{\mathbf{y}} = \mathbf{y}_i$;
    end
  end
  for *each particle* $i = 1, \ldots, n_s$ **do**
    update the velocity using equation (16.6);
    update the position using equation (16.1);
  end
until *stopping condition is true*;

$$
\begin{aligned}
v_{ij}(t+1) &= v_{ij}(t) \\
&+ c_1 r_{1j}(t)[pb_{ij}(t) - x_{ij}(t)] \\
&+ c_2 r_{2j}(t)[lb_{ij}(t) - x_{ij}(t)]
\end{aligned}
$$

$$
x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)
$$