Chapter 2 Heuristics and Metaheuristics

吕志鹏

教授，博士生导师

人工智能与优化研究所　所长

华中科技大学　计算机科学与技术学院

Heuristics and Metaheuristics

Heuristics

* Definition - educated guess, it’s originated in the old Greek word *heuriskein:* art of discovering new strategies(rules) to solve problems

* When you use a heuristic to solve a problem, you have a gut feeling that it is a pretty good solution, but can not prove it mathematically

* You can not prove that there is not a better solution out there.

Metaheuristics

* The suffix *meta* also a Greek word: upper level methodology.

* Introduced by Dr. **Fred Glover** in 1986

* **Metaheuristic**: Upper level general methodology (templates) that can be used as guiding strategies in designing underlying heuristics to solve specific optimization problems.

* Relationship between heuristics and metaheuristics.

Types of Metaheuristics (1)

* **Single Solution Algorithms (S):**

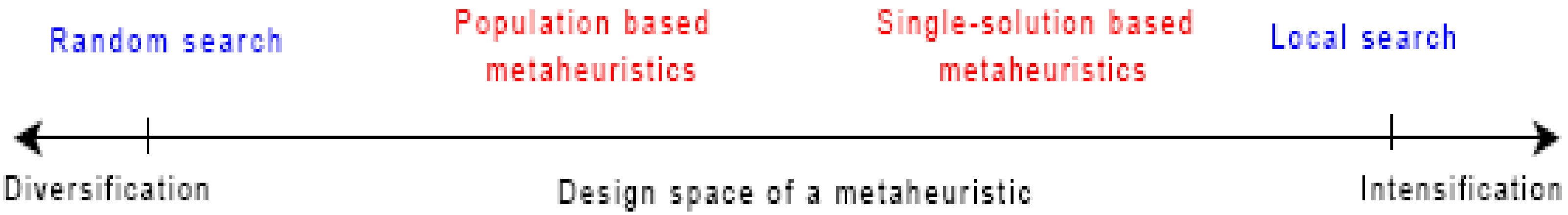
    * Common concepts for S-metaheuristics
    * Local search
    * Advanced local search

Types of Metaheuristics (2)

- **Population Based Algorithms (P):**
  - Common concepts for P-metaheuristics
  - Evolutionary algorithms
- **Hybrid Algorithms (H):**
  - Memetic algorithm: LS + GA

Metaheuristic Design

* No "Super Method" for all problems (NFL Theorem)
  * Exploration / Exploitation
  * Intensification / Diversification



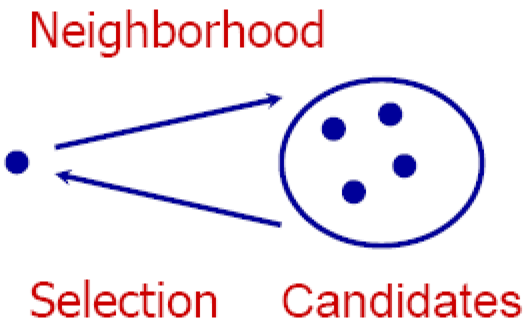
Classification of Metaheuristic

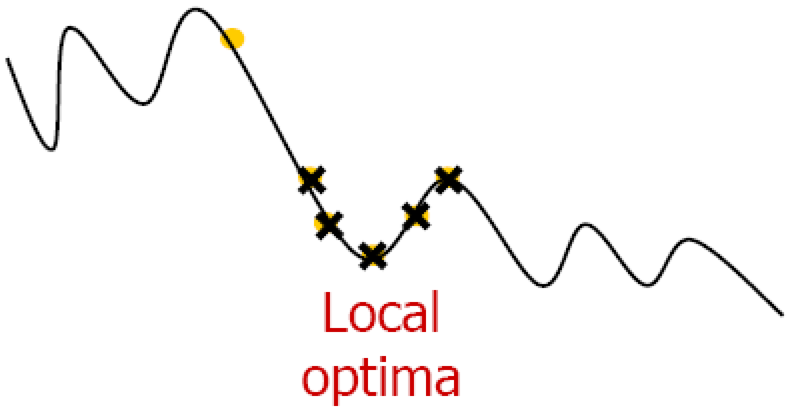
* Nature inspired/ Non nature inspired
* Memory usage / Memory less
* Deterministic/ Stochastic
* Population based/ Single-solution based
* Iterative/ Greedy
* Dynamic vs. static objective function
* One vs. various neighborhood structures

S-Metaheuristics

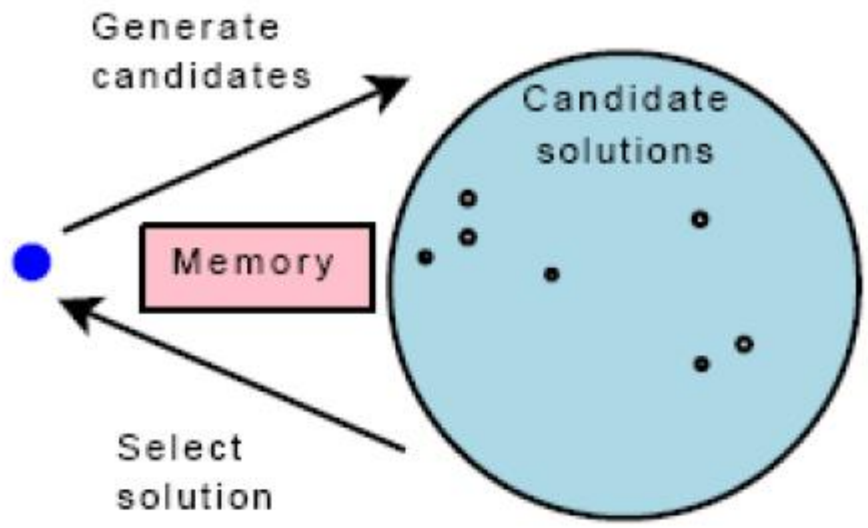
Single solution-based metaheuristics

- "Improvement" of a single solution; Walks through neighborhoods or search trajectories in the landscape
- Iterative exploration of the neighborhood. (intensification)
- All LS algorithms differ from each other in two aspects:
  - neighborhood definition → where **can** we go?
  - search strategy → where **do** we go?





Main Principle of S-metaheuristics



Main principles of single solution-based metaheuristics

Template of LS Algorithms

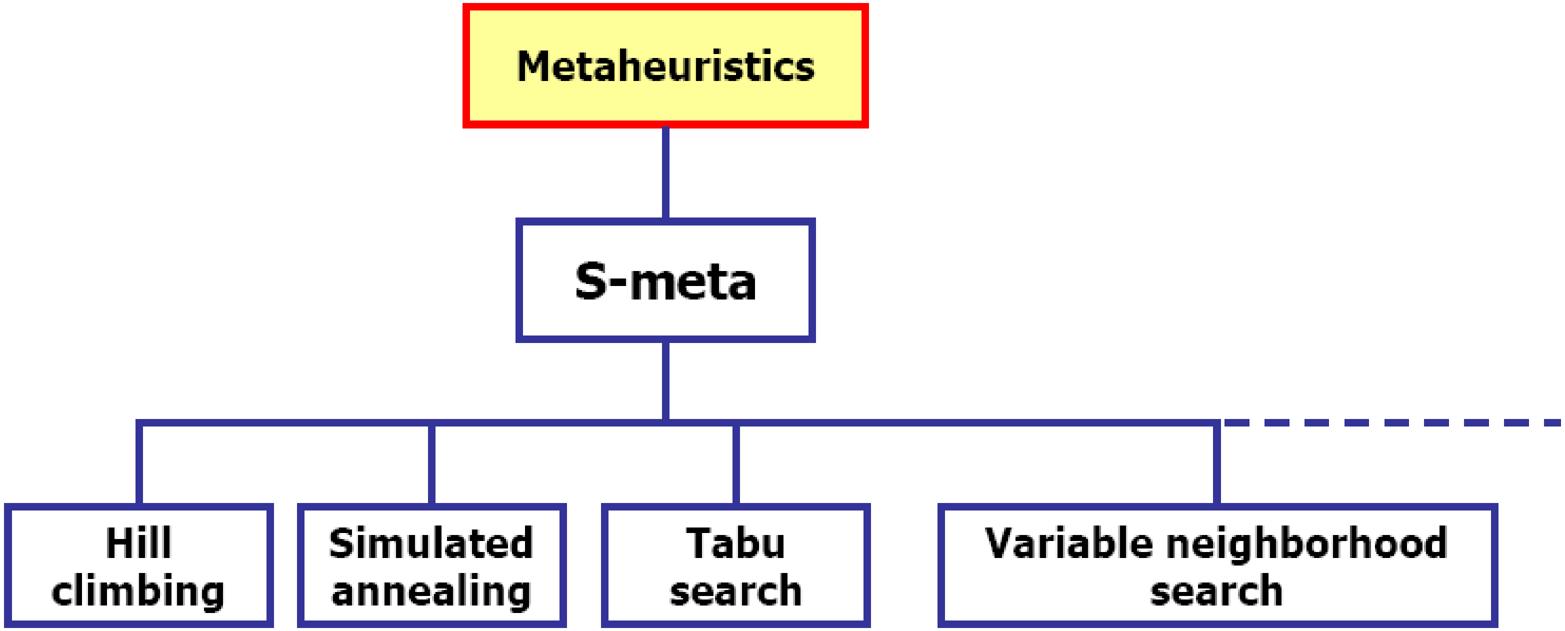
Generate initial solution x0

Repeat

* Construct the neighborhood of x0, denoted by N(x0)
* Select the best candidate solution x1 in N(x0)
* x0 ← x1

Until (stopping criterion is met)

Taxonomy of S-metaheuristics



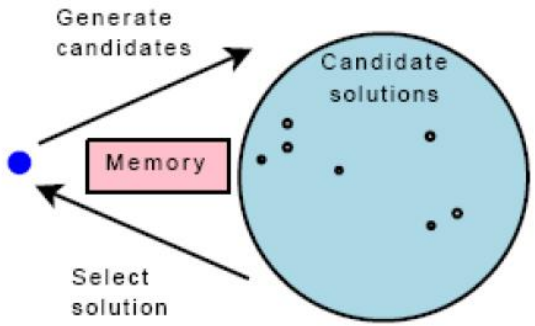
Outline of S-metaheuristics

* Common concepts for S-metaheuristics
  * Neighborhood
  * Initial solution
  * Incremental evaluation of neighbors
* Advanced Local Search
  * Iterative local search
  * Tabu Search
  * Simulated Annealing
  * Variable Neighborhood search

Neighborhood

Neighborhood

* **Neighborhood:** is defined by all the candidate solutions incurred by one single move on the current solution:



* **Principle of Neighborhood Definition:**
  * Binary Problems: one-flip (simplest is the best)
  * Grouping Problems: one-move, two-swap, Kemp chain
  * Scheduling Problems: permutation represents a priority queue. The relative order in the sequence is important.
  * Routing Problems: adjacency of the element is important.

Permutation Neighborhood

* Insertion, exchange and inversion neighborhoods:

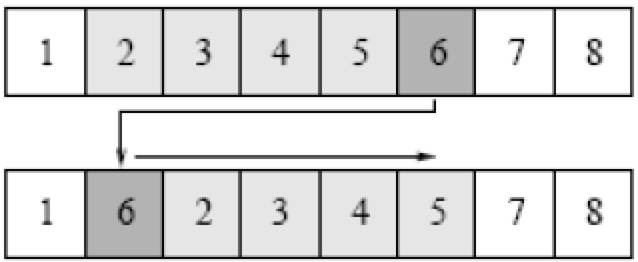


Fig. 2.7 Insertion operator.

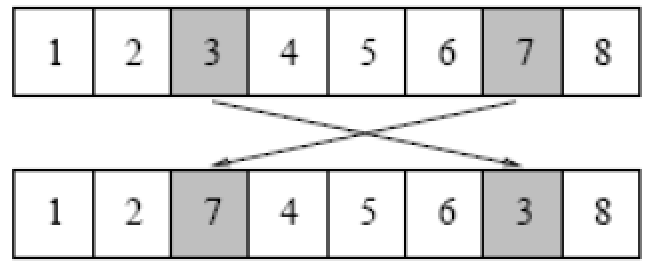


Fig. 2.8 Exchange operator.

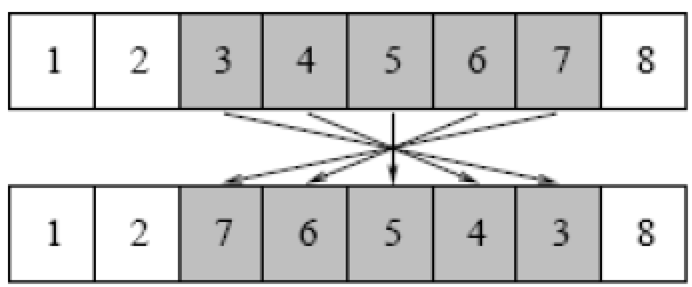


Fig. 2.9 Inversion operator.

Initial solution

- Two main strategies:
  - Random solution
  - Heuristic solution (e.g. Greedy Constructive Heuristic)
- Tradeoff: quality–computational time
- Using better initial solutions will not **always** lead to better local optima solution
- Generally to speak, its importance depends on the search power of local search algorithm

Incremental evaluation of neighborhood

* **Very Important!**

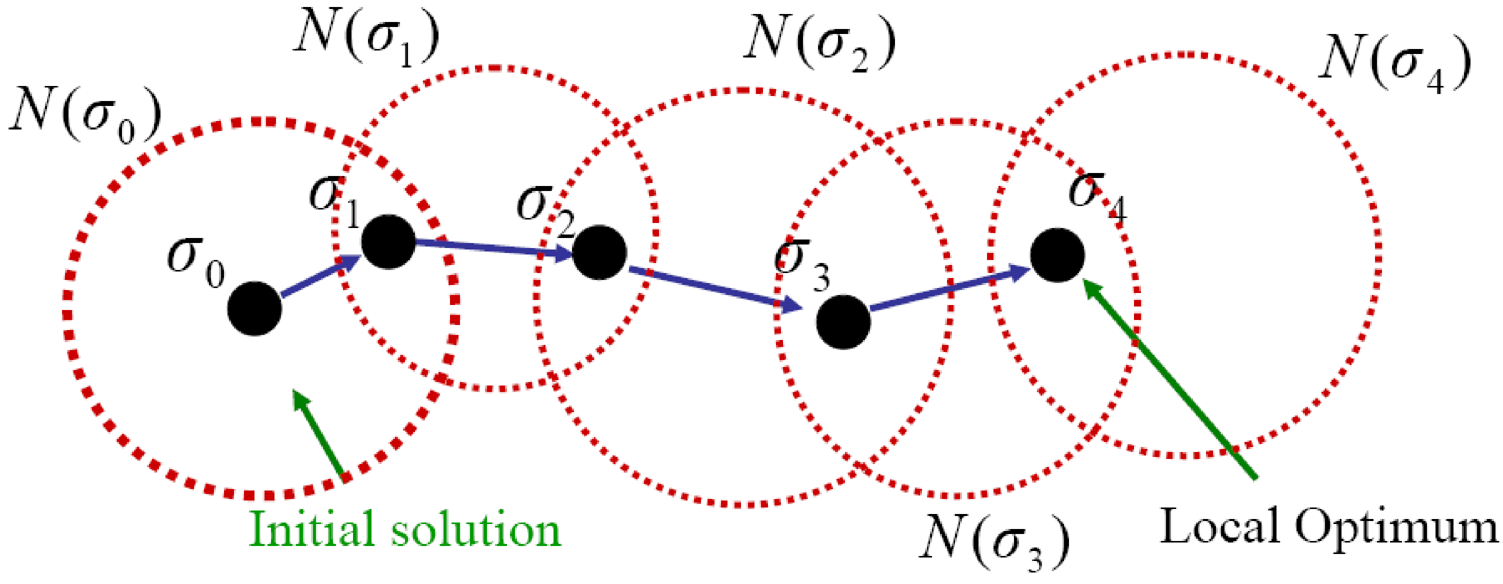
* Evaluation of a solution: Most expensive part of a metaheuristic. One should quickly determine the incremental value of the objective function for each candidate solution.

* Naïve evaluation: complete evaluation of every solution of the neighborhood

Local Search

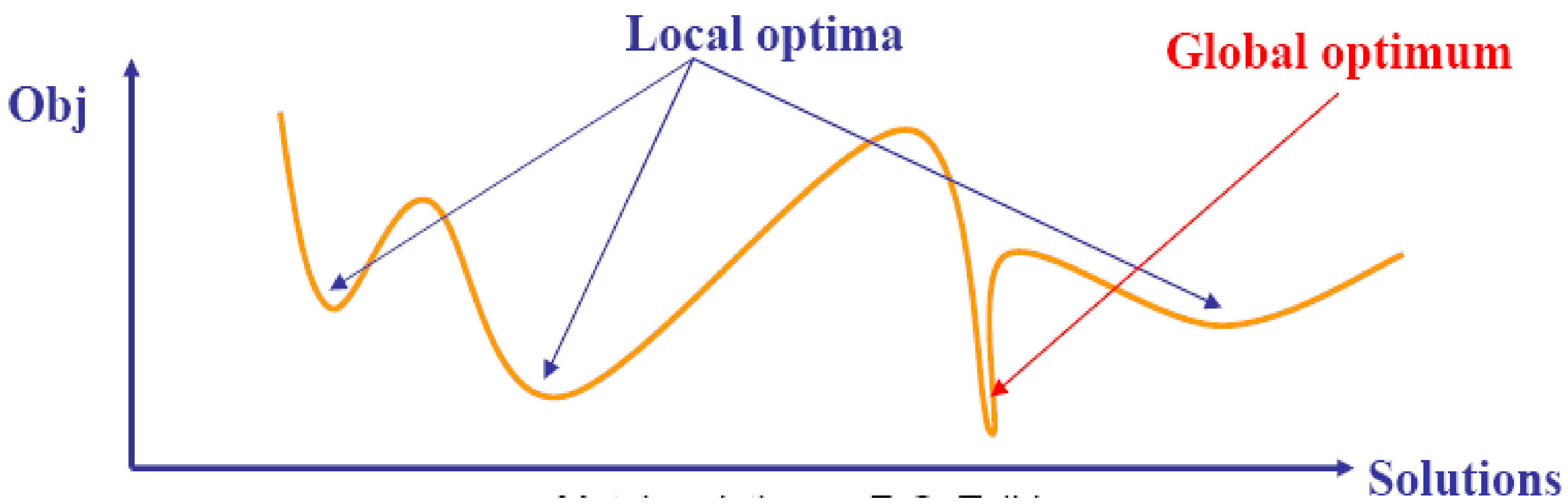
Local Search-Hill Climbing

* Oldest and simplest S-metaheuristic method
* Hill-climbing, descent, iterative improvement, and so on
* Replaces the current solution with an improving one



Local Search

* Easy to implement.
* It only leads to local optima.
* The found optima highly depends on the initial solution.



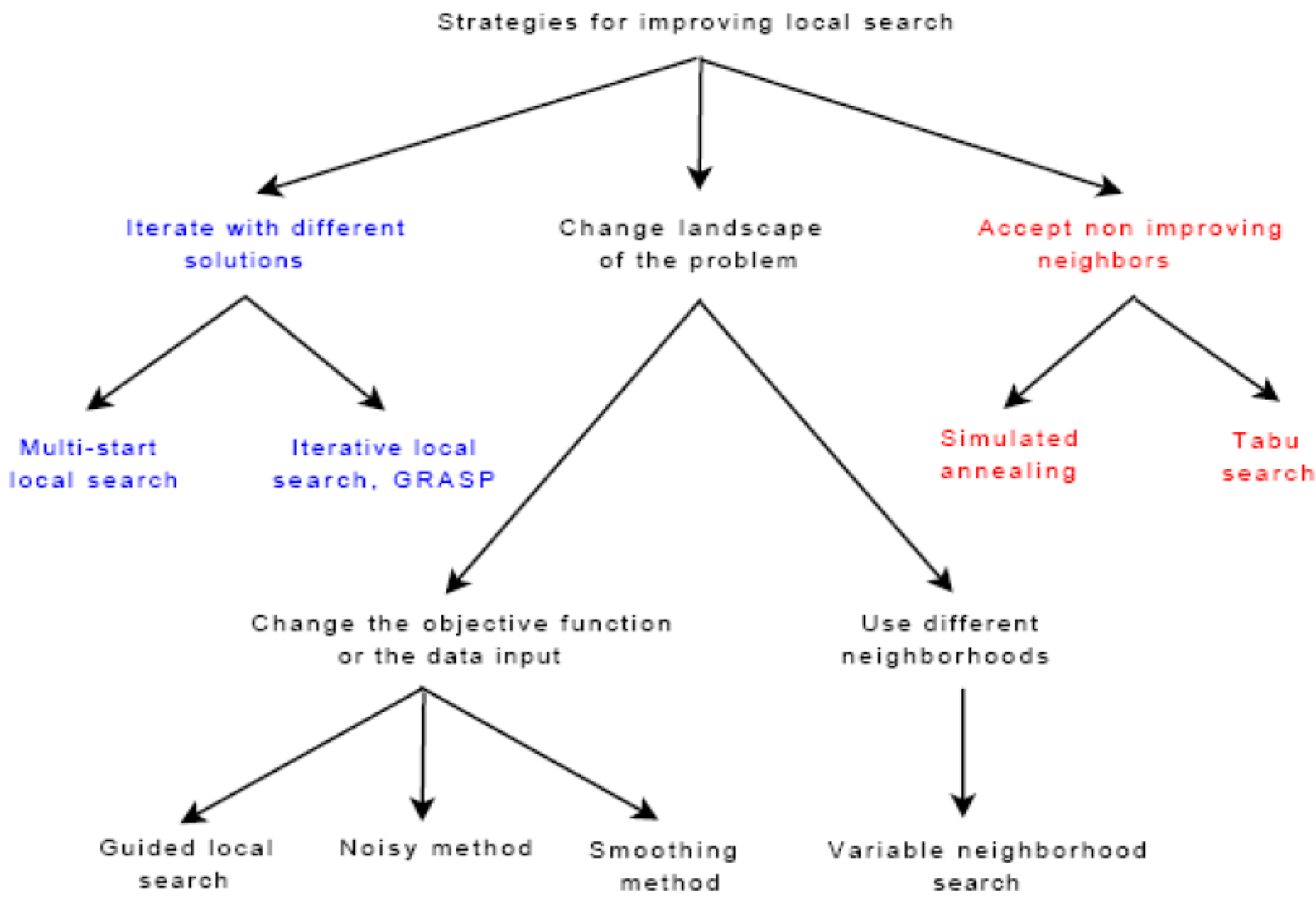
Selection of the neighbor

* **Best improvement:** Deterministic/full -choosing the best neighbor (i.e. that improves the most the objective function).

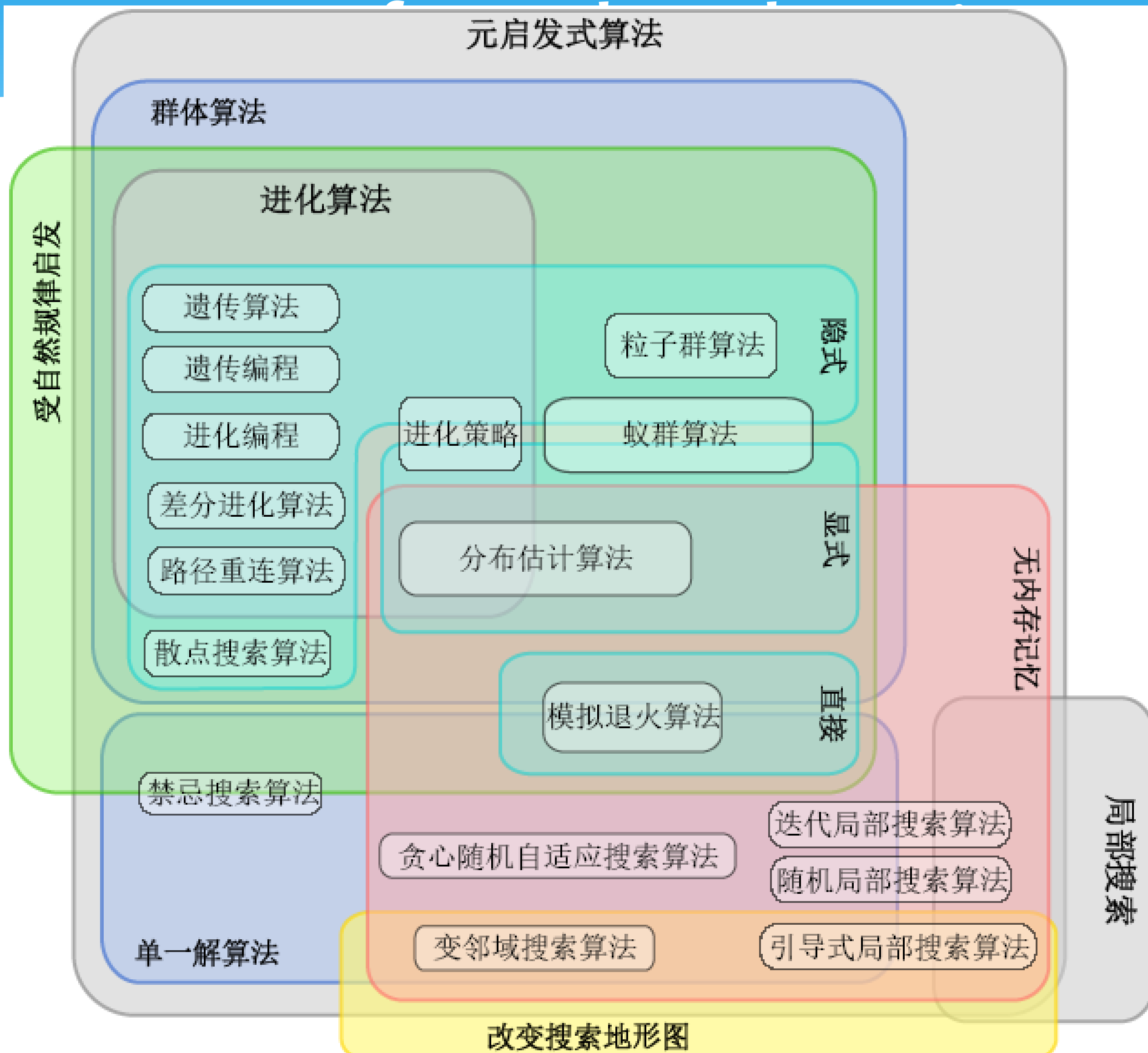
* **First improvement:** Deterministic/partial -choosing the first processed neighbor that is better than the current solution.

Advanced Local Search

Advanced local search
Escape from local optima

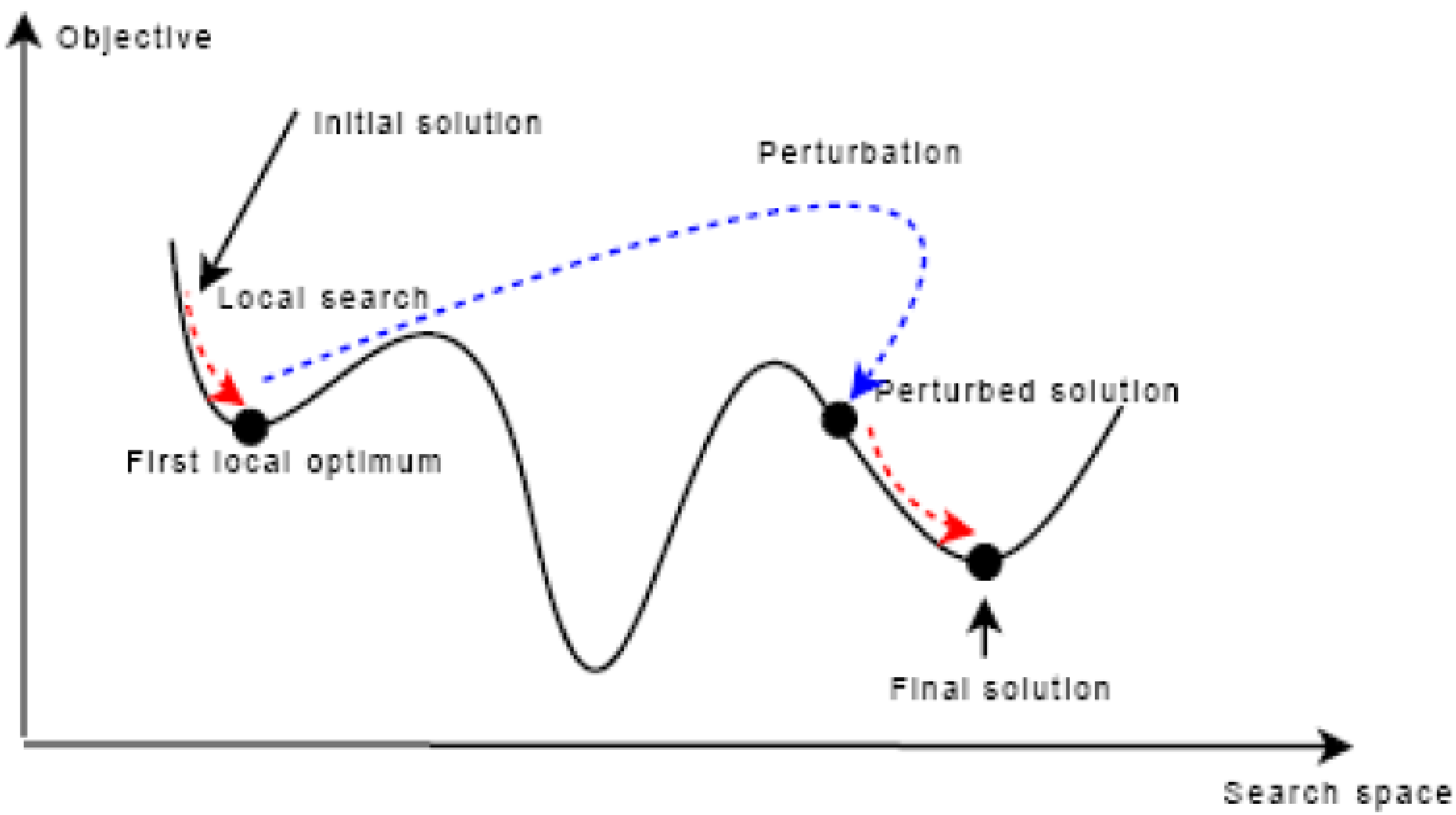


Advanced local search



Iterated Local Search

Iterated Local Search



Template of ILS Algorithms

TABLE I: Iterated Local Search Algorithm

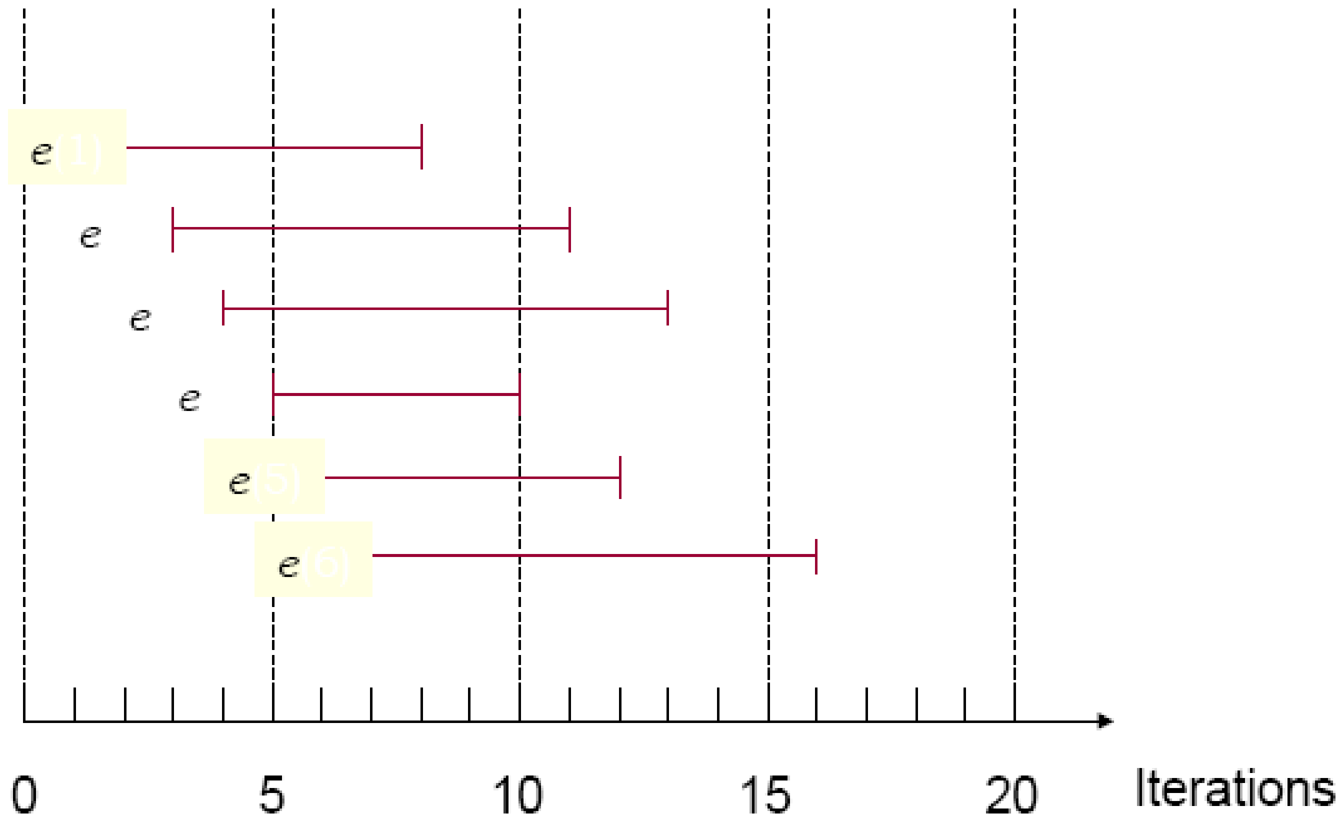
1:	$s_0 \leftarrow$ Initial Solution
2:	$s' \leftarrow$ Local Search(s_0)
3:	**repeat**
4:	$\quad s^* \leftarrow$ Perturbation Operator(s')
5:	$\quad s^{*'} \leftarrow$ Local Search(s^*)
6:	$\quad s' \leftarrow$ Acceptance Criterion$(s^{*'}, s')$
7:	**until** stop condition met

Tabu Search

Tabu Search

- Proposed by Dr. **Fred Glover** (1986)
- It behaves like Hill Climbing algorithm
- But it accepts non-improving solutions in order to escape from local optima (where all the neighboring solutions are non-improving)
- Deterministic algorithm

Tabu Search Illustration



Template of TS Algorithms

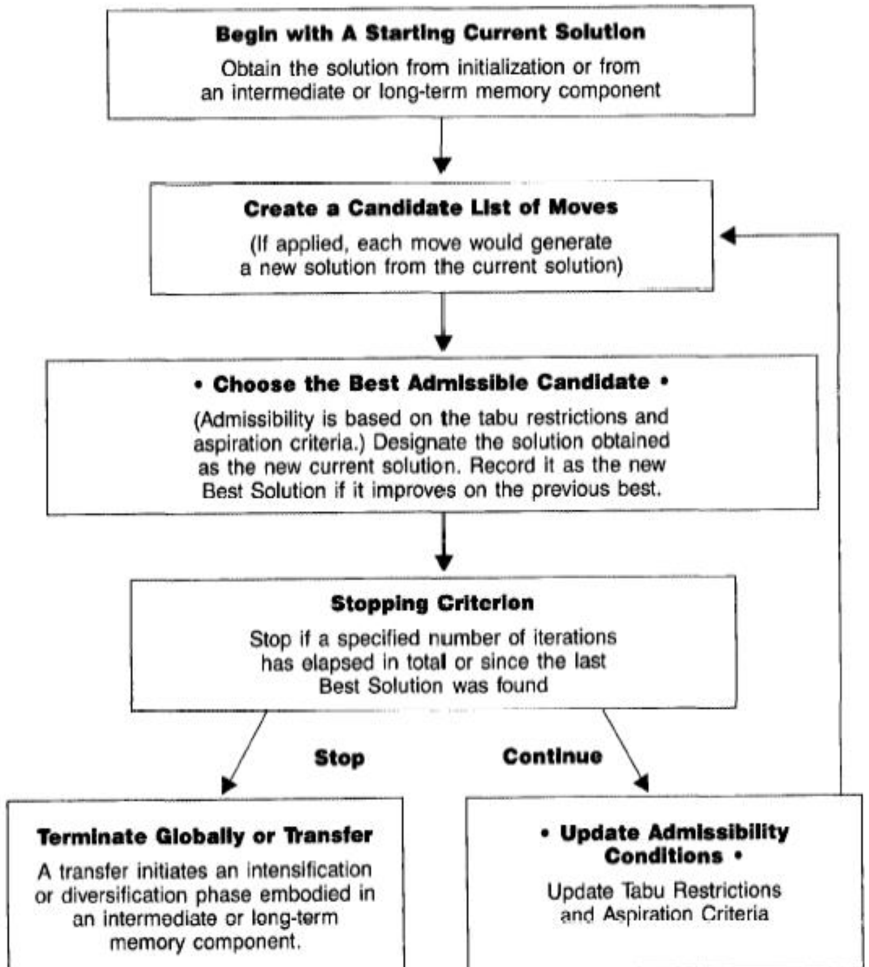


Figure 1: Tabu search short-term memory component.

Design questions

* Tabu list (Short Term Memory)
  * Attributive, not solution (expensive)
  * Recency based
  * Multiple tabu lists
  * Tabutenure (length of tabu list)
* Aspiration criterion: accepting tabu moves if it can override the best found solution found so far.
* Medium term memory (Intensification): giving priority to attributes of a set of elite solutions
* Long Term Memory (Diversification)
  * Frequency-based

Simulated Annealing

Simulated Annealing

* Mimics the physical annealing process (statistical mechanics).
* Material is heated and slowly cooled towards a strong crystalline structure (instead of meta stable states)
* The first SA algorithm was developed in 1953 (Metropolis).
* Kirkpatrick, S , Gelatt, C.D., Vecchi, M.P. 1983. “Optimization by Simulated Annealing”. Science, vol 220, No. 4598, pp 671-680.

Simulated Annealing

* SA allows downwards steps.
* A move is selected at random and its acceptation is conditional (stochastic Boltzmann distribution)

$$P(\Delta E) = e^{\frac{eval(v_n) - eval(v_c)}{T}} = e^{\frac{-\Delta E}{T}}$$

* Role of the temperature:
  * T small : local search (end of the search)
  * T large : random search (beginning of the search)

Template of SA Algorithms

Generate initial solution x0, T=T0

Repeat

* Construct the neighborhood of x0, denoted by N(x0)
* Randomly generate one candidate solution x1 in N(x0)
* x0 ← x1 with the probability $P(\Delta E) = e^{\frac{eval(v_n) - eval(v_c)}{T}} = e^{\frac{-\Delta E}{T}}$
* Update temperature T

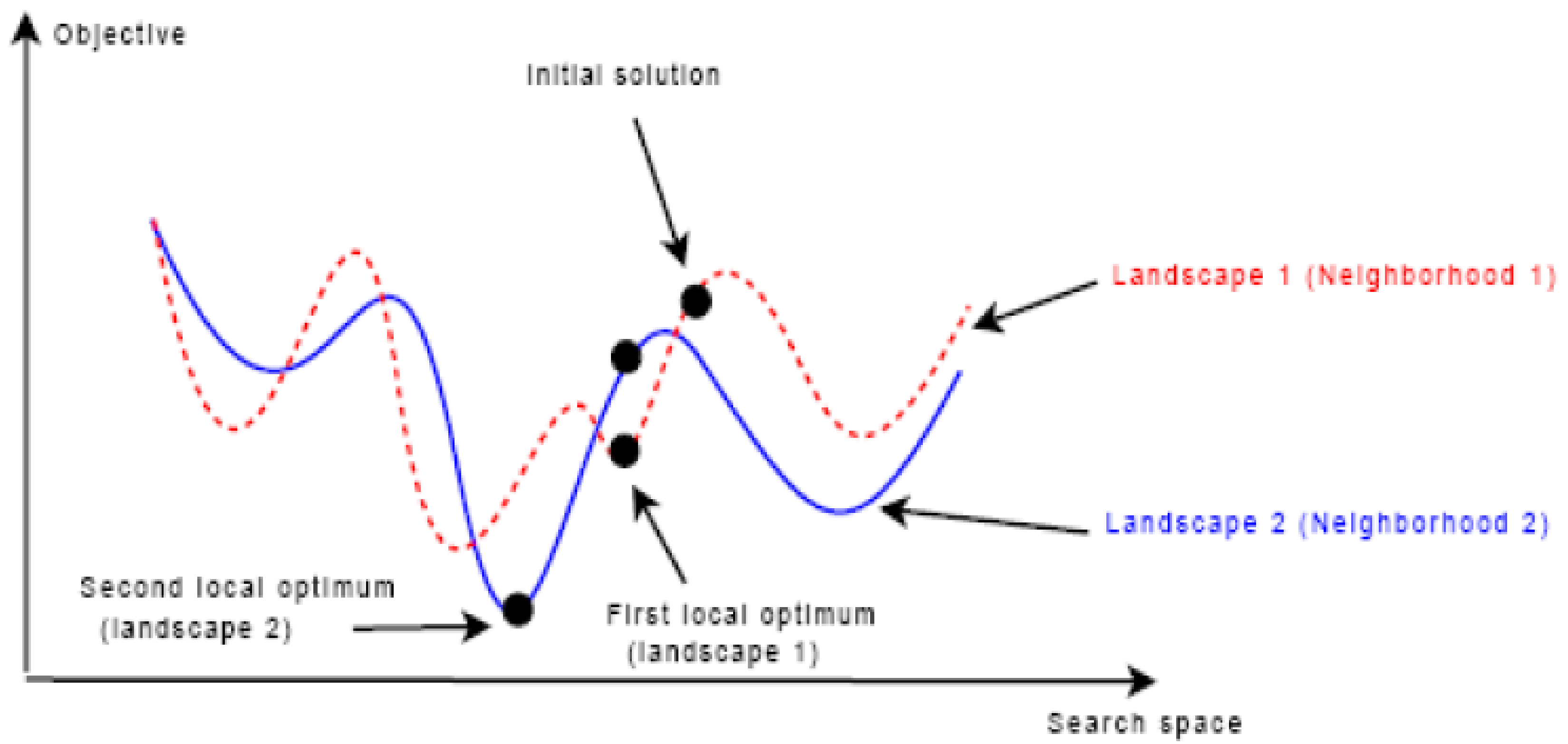
Until (stopping criterion is met)

Cooling Schedule

* Main design questions:
  * Initial temperature
    * *Not* be too hot-random walk
    * Hot enough-allow moves, else hill climbing
  * Equilibrium state
    * A sufficient number of moves at each temperature
  * Cooling
    * Compromise: quality / search time.
    * Different strategies: linear, geometric, logarithmic, adaptive…

Variable Neighborhood Search

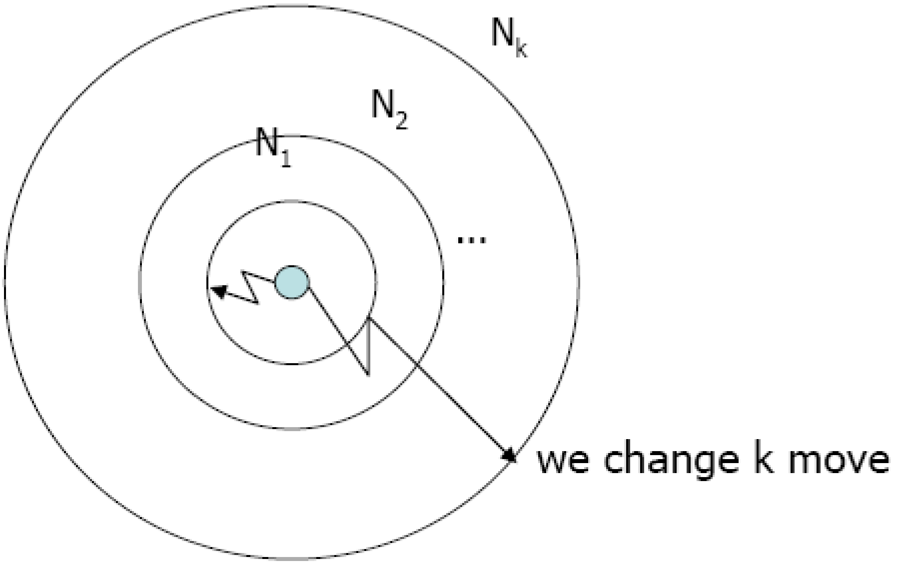
Variable Neighborhood Search



Variable Neighborhood Search

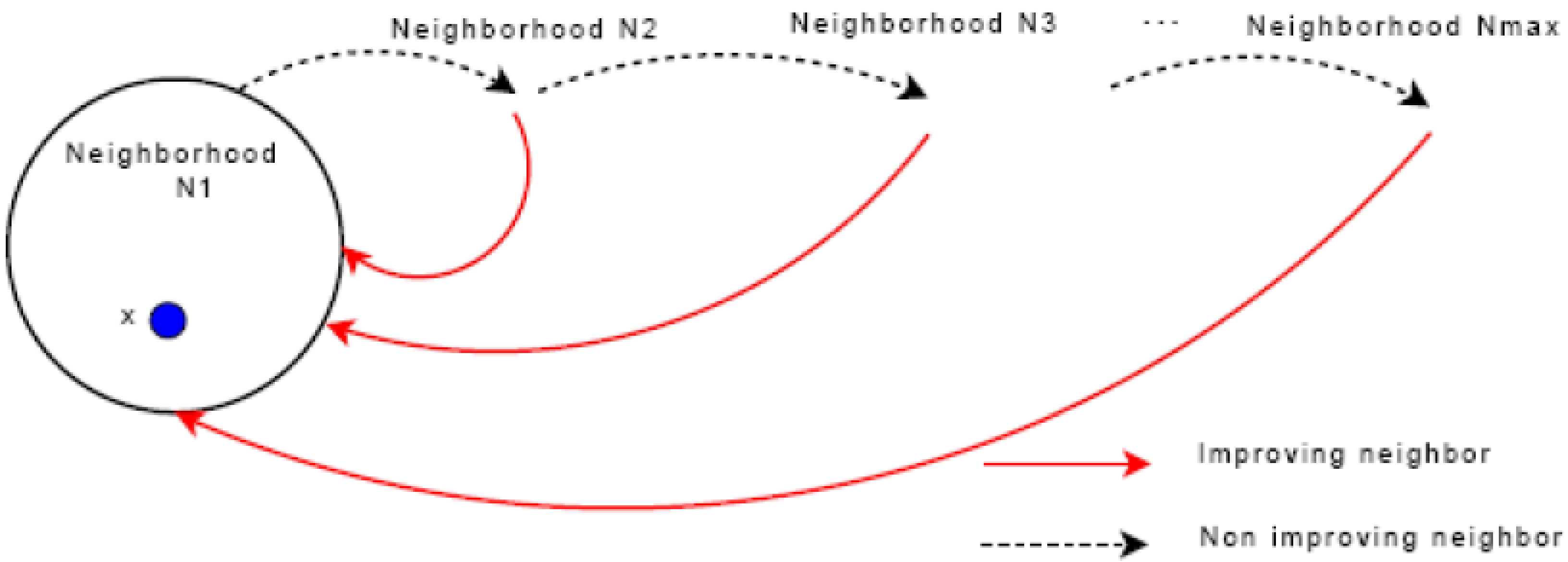
- Different neighborhoods
  - Ex: Nk → Neighborhood in k distance

- Order of exploration
  - Ex: Nk(x) → Set of solutions in the kth neighborhood of x.



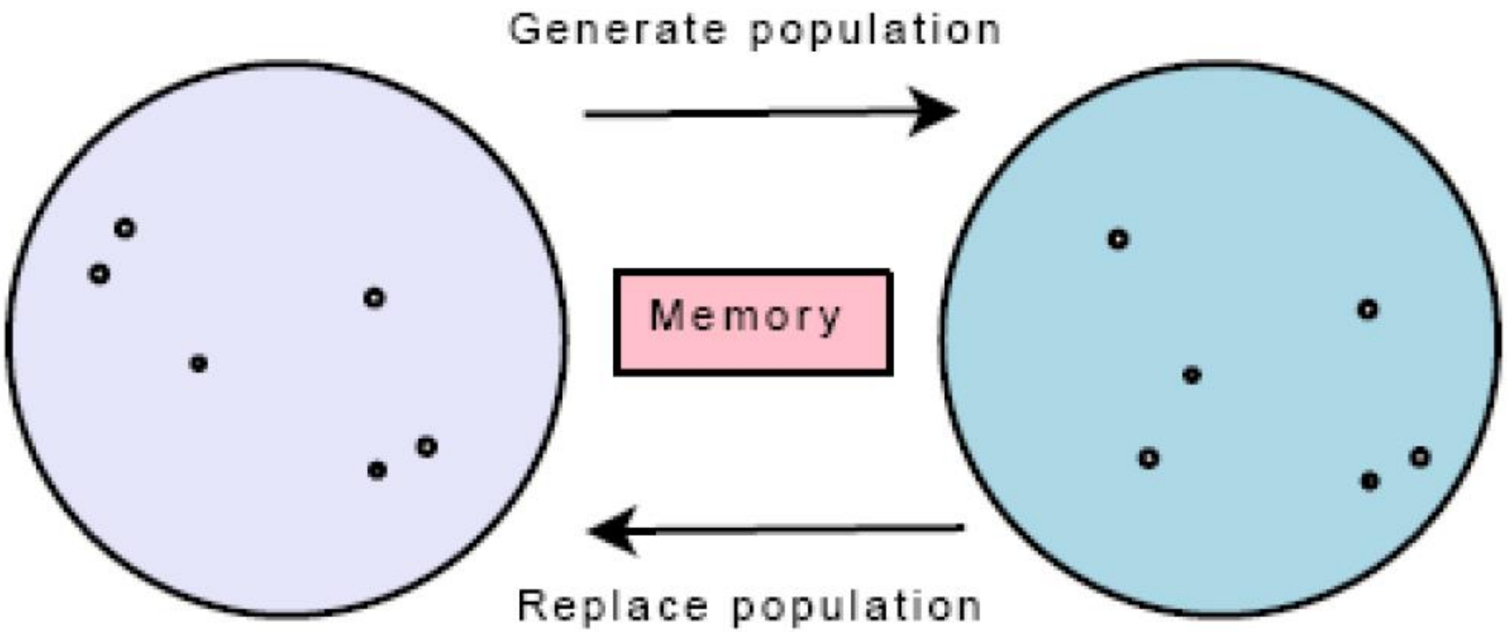
Variable Neighborhood Search

- VNS ends when there is no improvement with the all neighborhoods
- The final solution provided by the algorithm should be a local optimum with respect to all *kmax* neighbourhoods.

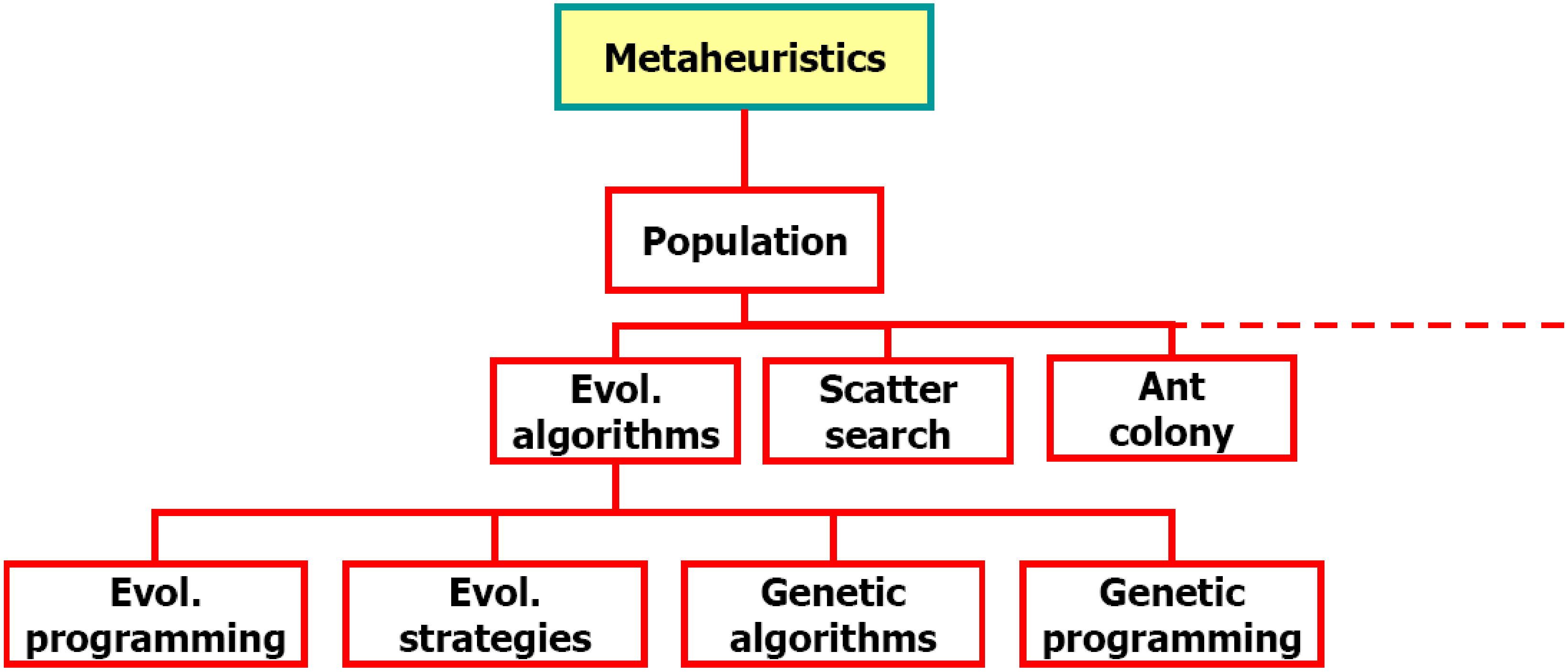


Population Based Metaheuristics

Template of P-metaheuristics

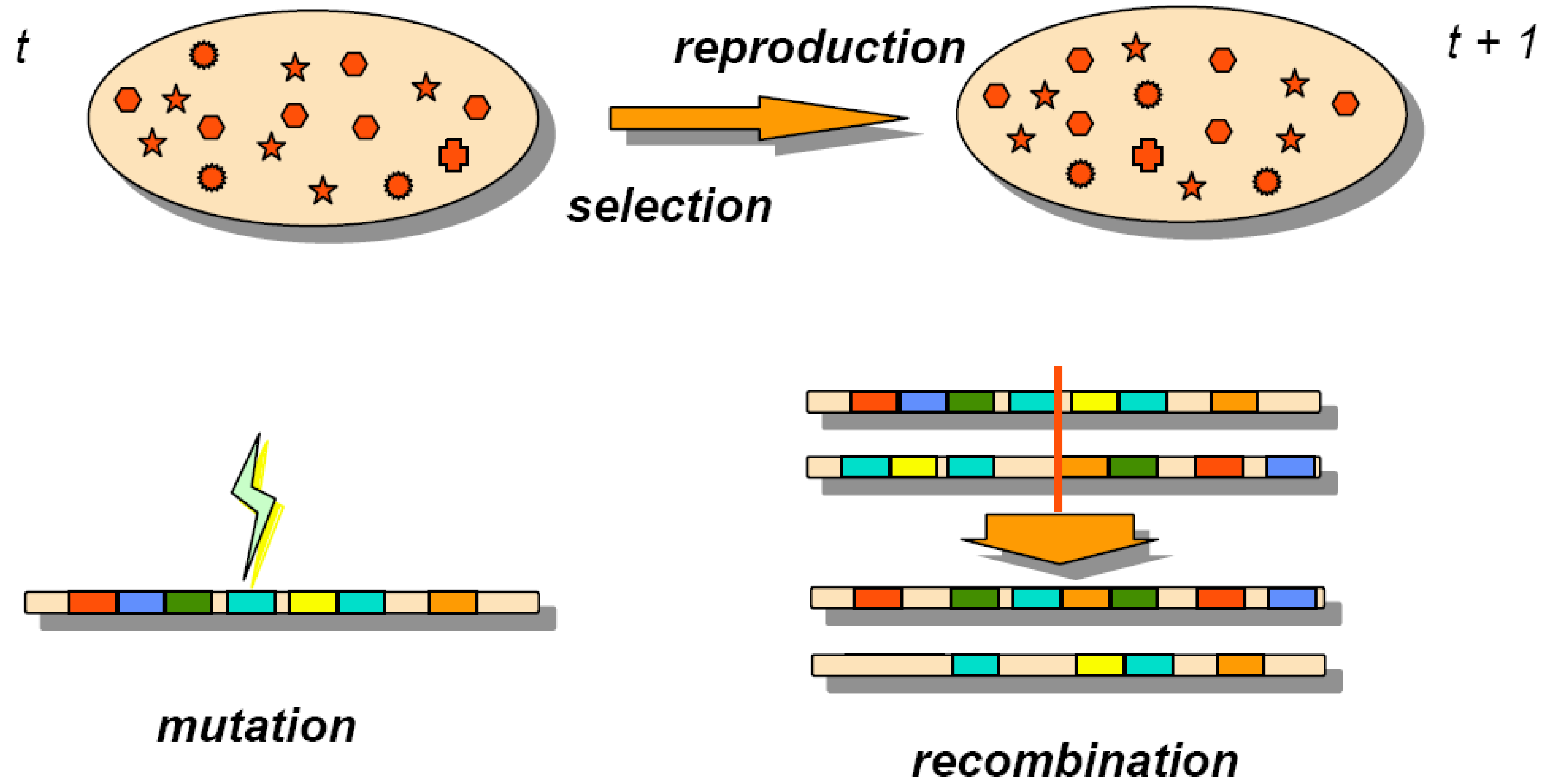


Taxonomy (Population-based Metaheuristics)



- Single solution metaheuristics are Intensification oriented
- Population-based metaheuristics are Diversification oriented

Main Components



Common search components for evolutionary algorithms

* Selection strategies: which parents are selected for reproduction.
* Reproduction strategies: Semantic crossover/mutation operators.---problem structure.
* Replacement strategies: how the current population is updated according to the generated offsprings. How to maintain a healthy population to enhance the search.

Genetic Algorithm

Genetic Algorithms (GA) OVERVIEW

- A class of probabilistic optimization algorithms
- Inspired by the biological evolution process
- Uses concepts of “Natural Selection” and “Genetic Inheritance” (Darwin 1859)
- Originally developed by John Holland (1975)

GA overview (cont)

- Particularly well suited for hard problems where little is known about the underlying search space
- Widely-used in business, science and engineering

GA overview (cont)

A genetic algorithm maintains a population of candidate solutions for the problem at hand, and makes it evolve by iteratively applying a set of stochastic operators

Stochastic operators

* ***Selection*** replicates the most successful solutions found in a population at a rate proportional to their relative quality
* ***Recombination*** decomposes two distinct solutions and then randomly mixes their parts to form novel solutions
* ***Mutation*** randomly perturbs a candidate solution

Simple Genetic Algorithm

produce an initial population of individuals

evaluate the fitness of all individuals

while termination condition not met **do**

    select fitter individuals for reproduction

    recombine between individuals

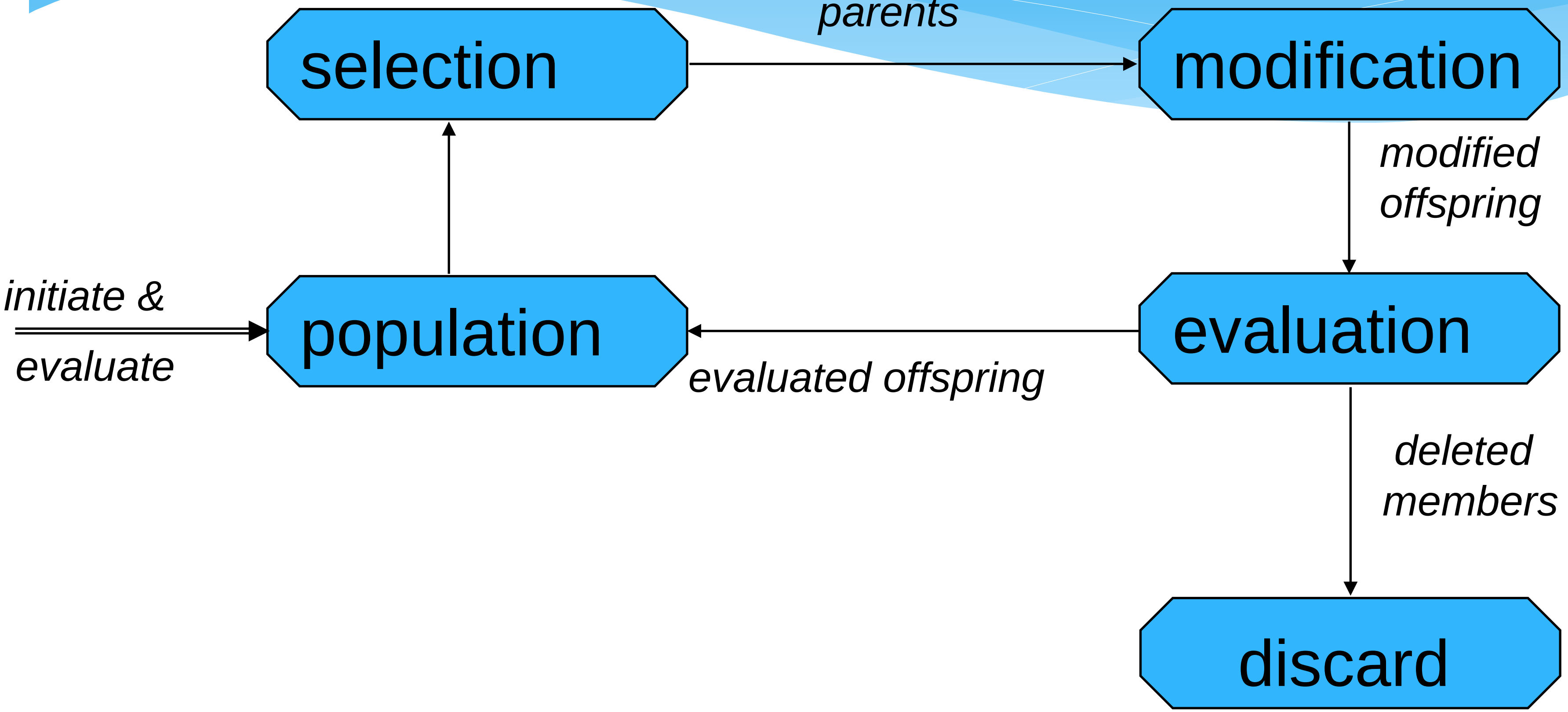
    mutate individuals

    evaluate the fitness of the modified individuals

    generate a new population

End while

The Evolutionary Cycle



Components of a GA

A problem definition as input, and

- Encoding principles (gene, chromosome)
- Initialization procedure (creation)
- Selection of parents (reproduction)
- Genetic operators (mutation, recombination)
- Evaluation function (environment)
- Termination condition

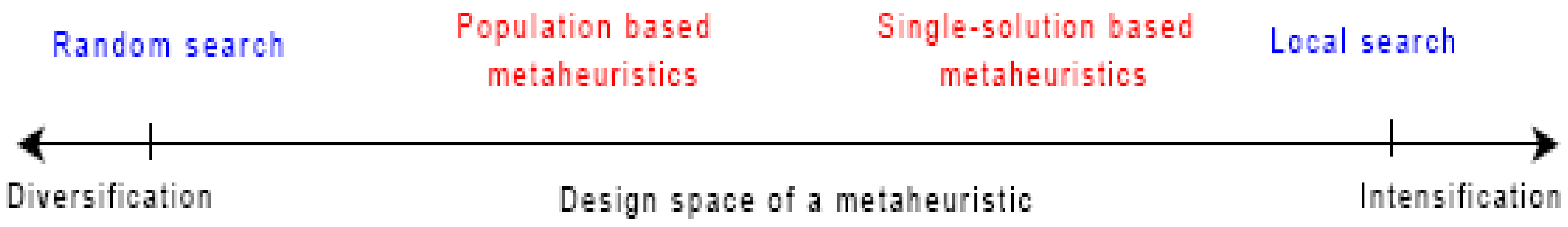
Other P-Metaheuristics

- **Ant Colony Optimization**
- **Particle Swarm Intelligence**
- **Bee Colony**
- **Artificial Immune Systems**

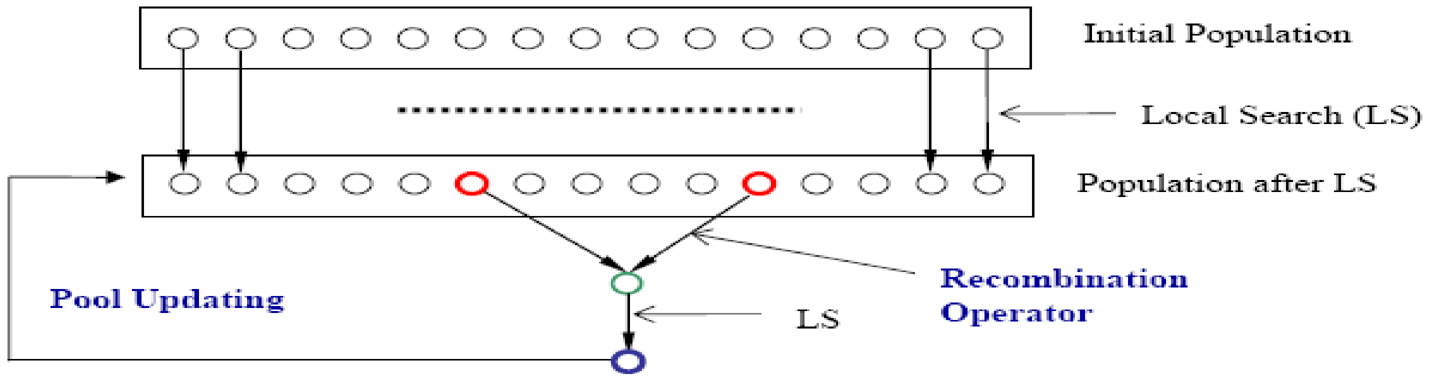
Hybrid Metaheuristics

Hybrid Metaheuristics

* **Local Search:** **I**ntensification oriented
* **Population-Based Algorithm:** **D**iversification oriented
* Their combination leads to a better tradeoff between **I&D**?



Memetic Algorithm



Hybrid Evolutionary Algorithm (LS+EA)

* Hybrid Evolutionary Algorithm = Memetic Algorithm
* Difference with EA algorithms:
  * Smaller population (20 to 30) with elite solutions
  * Semantic recombination operator ---- **big jump**
  * Pool Updating (population **diversity** is very important!)

Sum Up

- **Highlights to design a good metaheuristic algorithm:**
  - Select good neighborhood(s)
  - Fast evaluation of neighborhood moves
  - Tradeoff between Intensification and Diversification
    - Select appropriate search strategies
    - Combination of neighborhoods
    - Combination of search strategies
  - Semantic/problem specific operators
  - Using memory and history in the search
  - Always consider problem structure anywhere

Sex and Metaheuristic Metaphors

* In short, *Courtship Algorithms* provide a natural foundation for developing new metaheuristic processes. To inform such processes, it is possible to formulate a collection of Key Strategic Principles, as embodied in time-honored expressions that people have applied to courtship.

Sex and Metaheuristic Metaphors

- *Have we met before?* (Use memory to exploit recurrences.)
- *Variety is spice. (Employ diversification at all levels.)*
- *Try it, you'll like it. (Experiment and probe beyond the surface.)*
- *Opposites attract. (Join diversification (via opposites) and intensification (via attraction).)*
- *Familiarity breeds … (Watch for important affinities.)*

Sex and Metaheuristic Metaphors

- *When in Rome …* (Adapt to the terrain.)
- *Once is not enough*. (Iterate over good options.)
- *Play the field*. (Don’t be restricted to a single move or strategy.)
- *Two’s company, three’s a crowd.* (Sometimes paired combinations work well.)
- *Ménage à trois* (And sometimes it can be worthwhile to go beyond pairs.)
- *The more the merrier*. (And then sometimes …)

Sex and Metaheuristic Metaphors

* *Am I getting warmer?* (Test the setting to determine the next move.)
* *Don’t stop now!* (Take advantage of momentum.)
* *Be gentle.* (But do so prudently.)
* *Be prepared.* (Initial strategies can set the stage for later ones.)
* *Let’s compare our …* (Discover advantageous similarities and differences.)
* *Be selective.* (Sift through options.)

Sex and Metaheuristic Metaphors

- *Don't leave things to chance*. (Random behavior can sometimes be counterproductive.)
- *No need to be timid*. (Well-timed aggressive moves can pay off.)
- *Go with the moment*. (Include short term strategies for congenial terrain.)
- *Trust your instincts*. (Intuitive exploration can supplement strict analysis.)
- *We've got to stop meeting like this!* (Introduce variation to avoid cycling traps.)

Sex and Metaheuristic Metaphors

- *The bar is closing. Let's find someplace more interesting to go!* (Establish limits and follow through with exploration of new terrain.)
- If at first you don't succeed, try, try again … (Restarting in new areas to find better results.)
- May I offer you these flowers? (In case of dynamic or imprecise optimization: reevaluate)
- Don't know whether I can propose another one. Let us try: "Don't do _that_ again ... at least not immediately" (Tabu search)

Sex and Metaheuristic Metaphors

* I'm not sure I'm ready for this. (Alternate between intensification and diversification strategies.)
* ”Do you have a sister?” (Neighborhood search may pay dividends.)
* "Your brains and my beauty" (generate combinations based on different criteria, e.g. rigorous and esthetic, quantitative and qualitative).
* Things are moving too quickly … (Don’t be afraid to be thorough.)

Thanks !