Optimization, Artificial Intelligence and Computational Intelligence

劉振隆
義守大學資訊管理學系教授

2007. 10.12

Outline

- Classical Optimization Methodology
- Computational Intelligence
- Some Applications on the Optimization
- Swarm Intelligence
- Conclusion

「如果讓我重做一次研究生」—王汎森院士(中研院歷史所)

- 「恭喜你對人類的知識有所創新，因此授予你這個學位」
- Learn how to learn — self help
- 與老師之夥伴關係(Teamwork)不斷Research
- 大量閱讀，進入研究領域，並循序漸進練習寫作
- 形成你的知識樹主幹—精讀5、6本原典
- 突破學科間的界線
- 留下時間，精緻思考
- 用兩條腿走路—專業學術與興趣
- Better to best

Classical Optimization Methodology

Nonlinear Programming

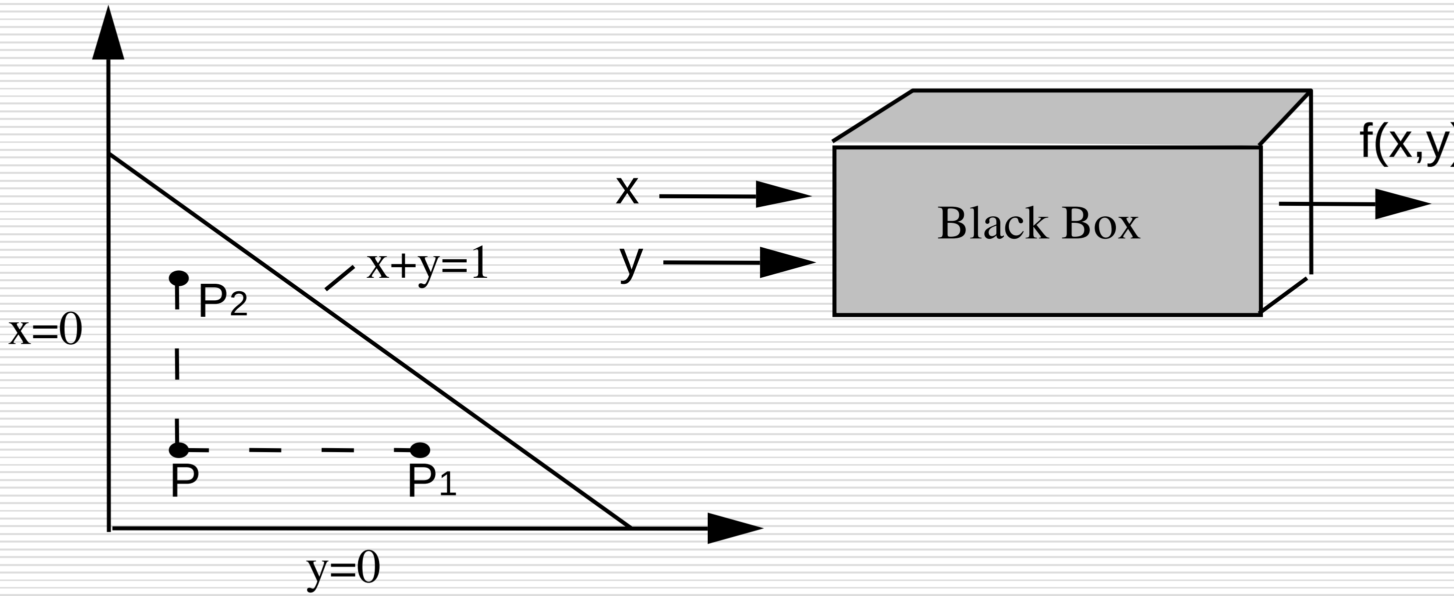
Minimize $f(\vec{x}), \vec{x} = \left\{ x_1, x_2, ..., x_N \right\} \in R^N$

Subject to $g_j(\vec{x}) \leq 0, \; j = 1, ..., p$

$h_j(\vec{x}) = 0, \; j = p + 1, ..., m$

where $f(\vec{x})$ represents the objective function , also named the fitness function, in the hyperplane of *N* dimensions.

Design theory and Operation Process



What combination of x and y will produce the lowest f(x,y) subject to the constraints $x > 0$, $y > 0$, $x + y < 1$?

Issues of Optimization

- Optimizing a function vs. optimization over time
- Discrete vs. continuous
- Do I have a model? Can I build a model?
- Feasible vs. optimal
- Globally vs. local optimal?
- Is computation expensive?
- What should be optimized, anyway?

Optimization Based Design and Analysis

Pose Design Problem As An Optimization Problem

- Can be single or multi-objective
- Systematically encode design constraints and objectives into fitness functions

Problem Properties Not Always Friendly

- Not differentiable
- Not convex
- Many local extrema
- No unique global optimum

Classical Optimization Methods

- Newton’s Method
- Gradient Methods
- Conjugate Direction Methods
- Quasi-Newton Methods
- Nedler-Mead Simplex Method
- ...

Gradient-Based Local Search

Lagrangian Operator:

$$L(\vec{x}, \lambda) = f(\vec{x}) + \sum_{i=1}^{p} \lambda_i g_i(\vec{x})$$

Kuhn-Tucker Condition :

$$\nabla f(\vec{x}^*) + \sum_{i=1}^{p} \lambda_i \nabla g_i(\vec{x}^*) = 0$$

Feasibility Requirement : $\nabla g_i(\vec{x}) \bullet \vec{S}^n \leq 0$

Usability Requirement : $\nabla f(\vec{x}) \bullet \vec{S}^n \leq 0$

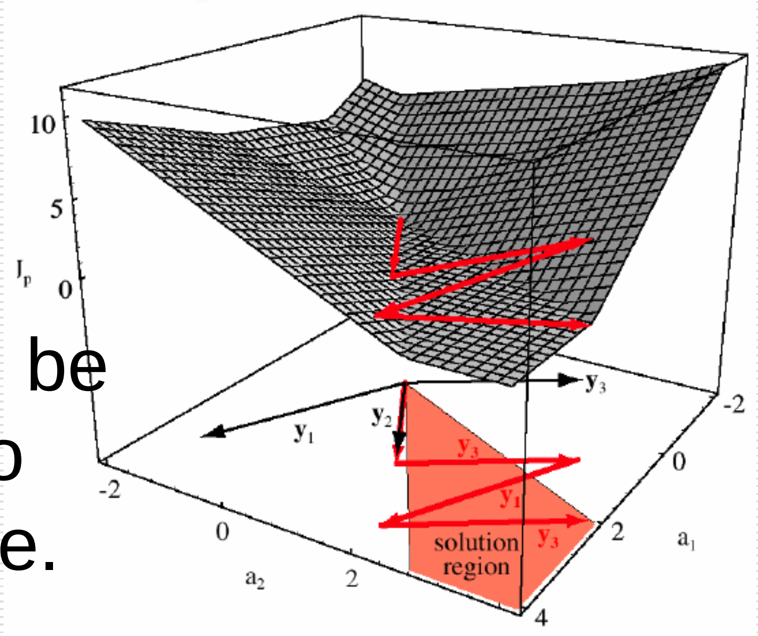
Iteration Procedure : $\vec{x}^{n+1} = \vec{x}^n + \beta \vec{S}^n$

(Golden Section Method)

Determining **S** is not easy

– Termed "learning rate" in AI

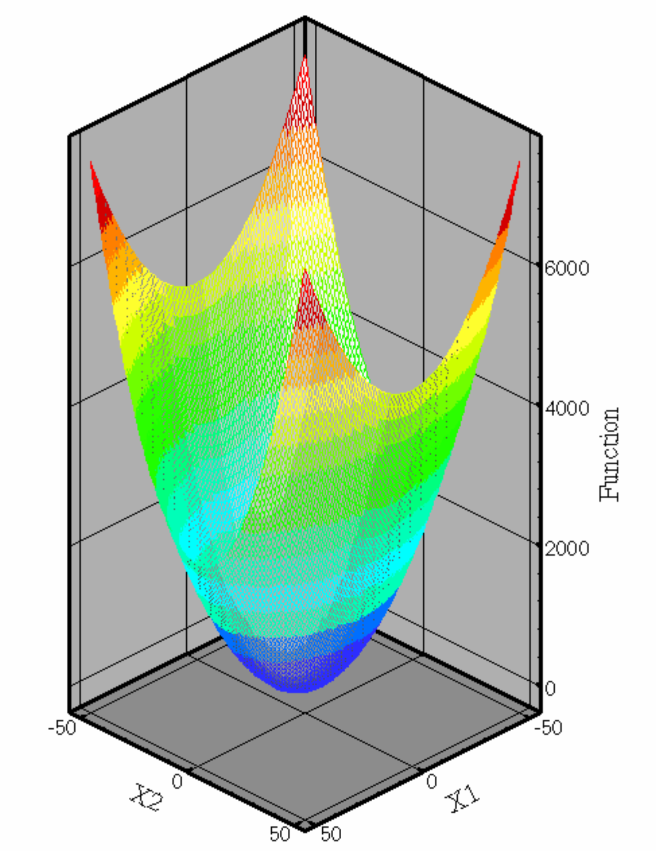
– Hard to determine **S**

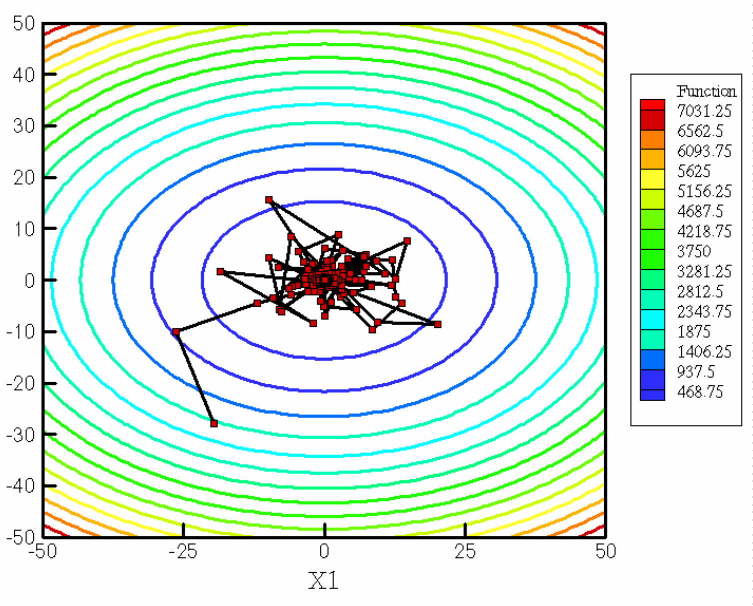


• If **S** is too small algorithm will be too slow to converge. If it is too large the procedure will diverge.

• Can select it using a line search or using a gold section method.

Unimodal Function --- work well

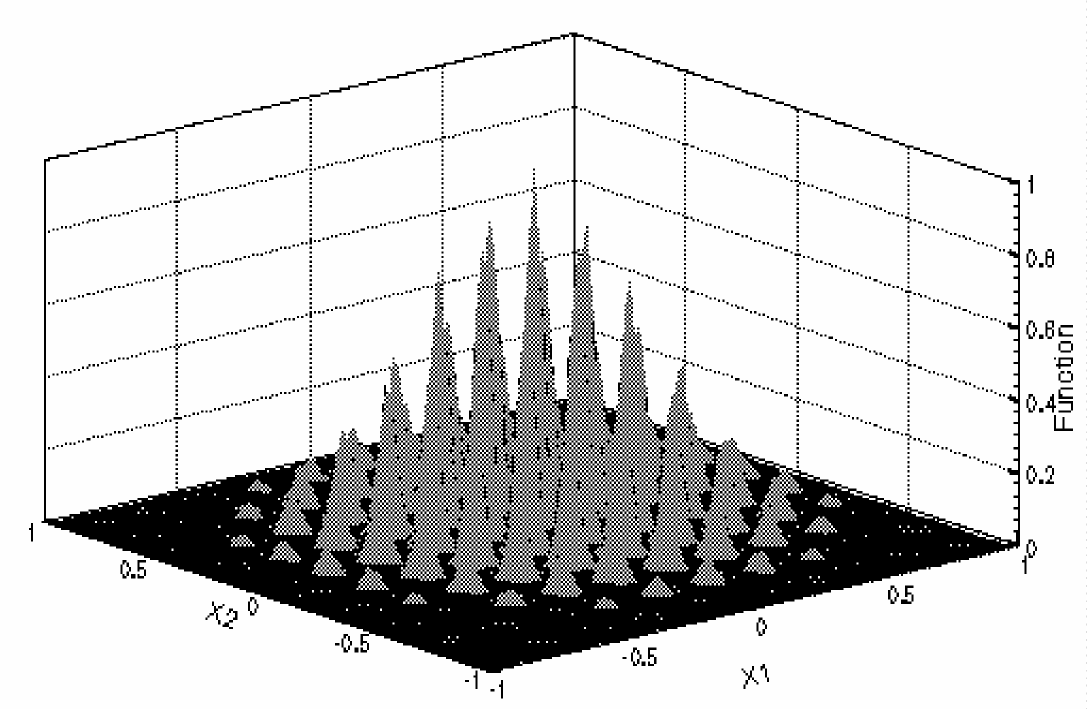




$$f(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$$

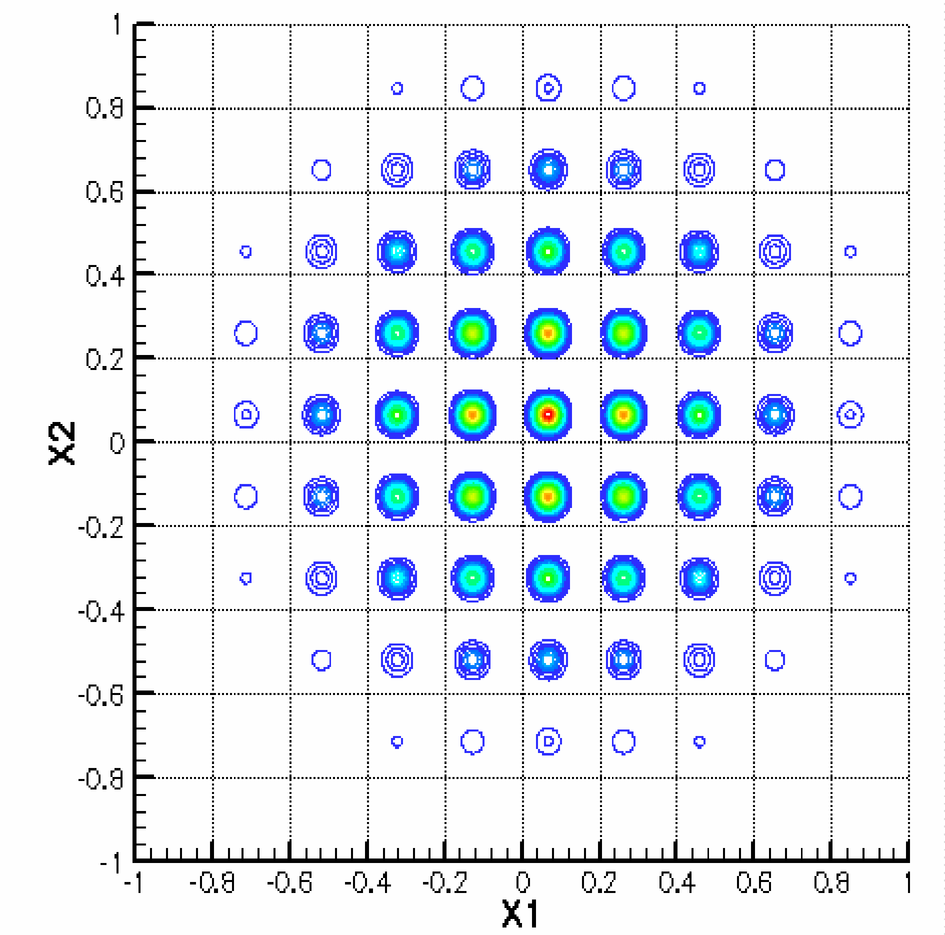
$$-50 \le x_i \le 50,\ i = 1, 2$$

Multimodal Function --- depend on starting point



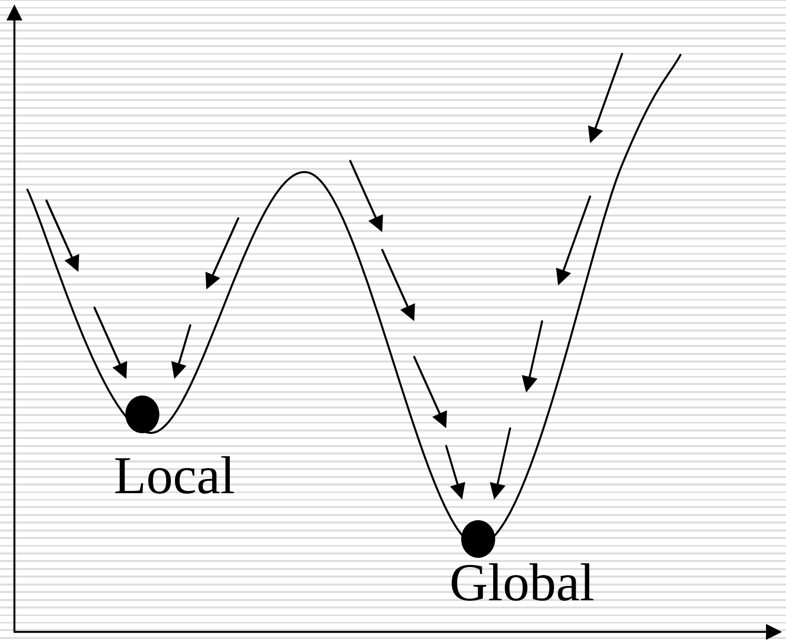
$$f(\vec{x}) = \prod_{i=1}^{2} \sin^6(5.1\pi\, x_i + 0.5)\exp(-4\ln 2(x_i - 0.0667)^2 / 0.64)$$

$$0 \le x_i \le 1, i = 1,2$$

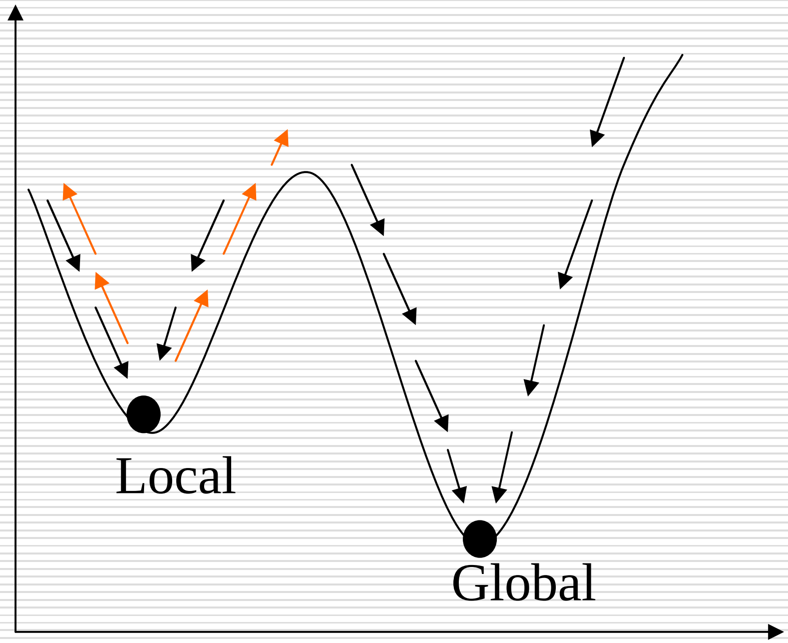


$$f_{\max}(0.06683, 0.06683) = 1.0$$

Gradient Search vs. Heuristic Search



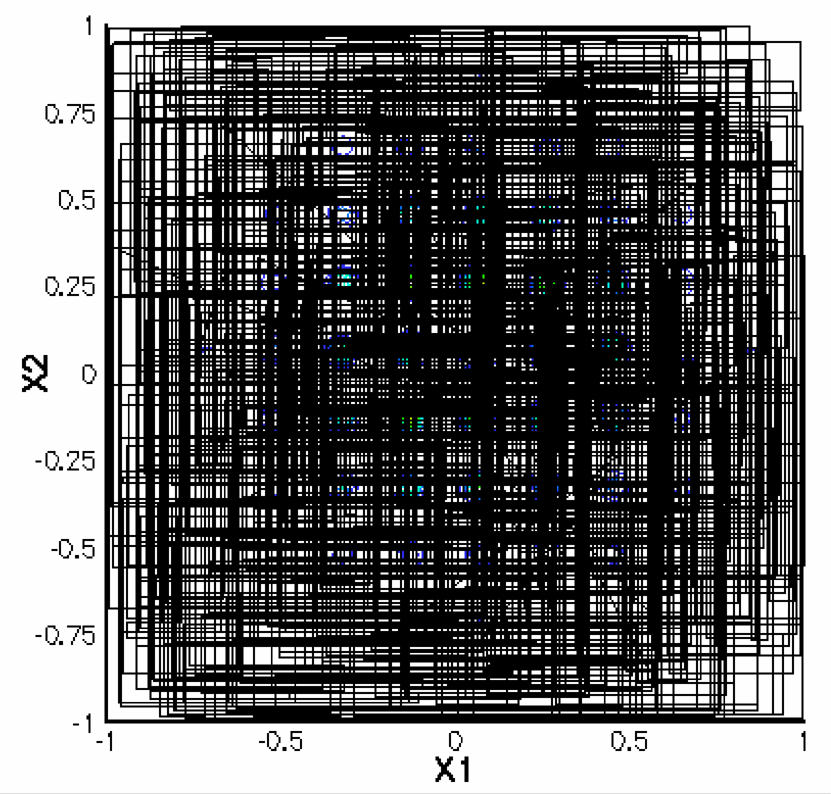
Gradient Search



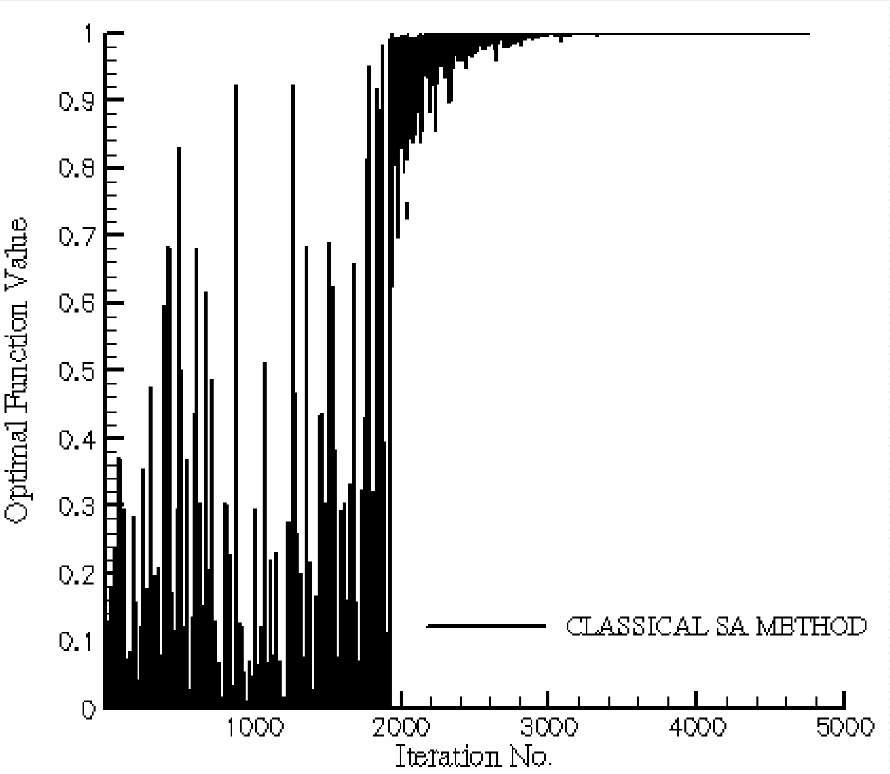
Heuristic Search

Meta-heuristic Search (Probability)

$$P(s') = \begin{cases} 1, & \text{, if } f(s') \le f(s) \\ \exp\left(\dfrac{-(f(s') - f(s))}{T}\right), & \text{if } f(s') > f(s) \end{cases}$$

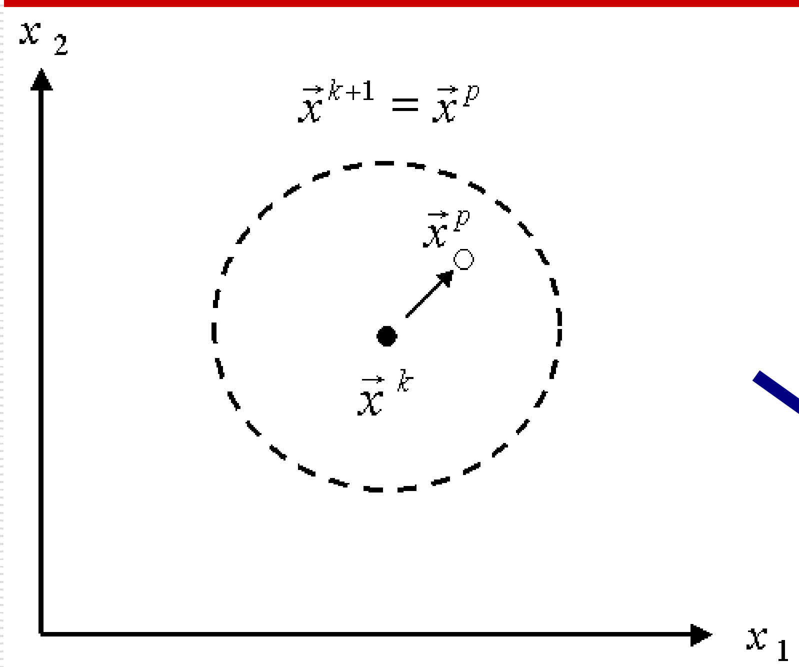


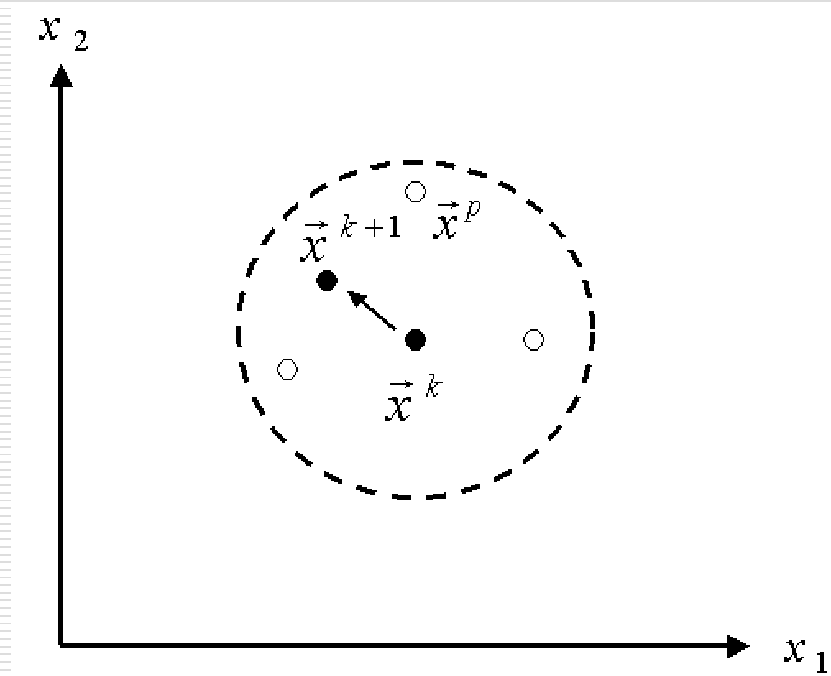
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.5
0
0.5
1
X1
X2



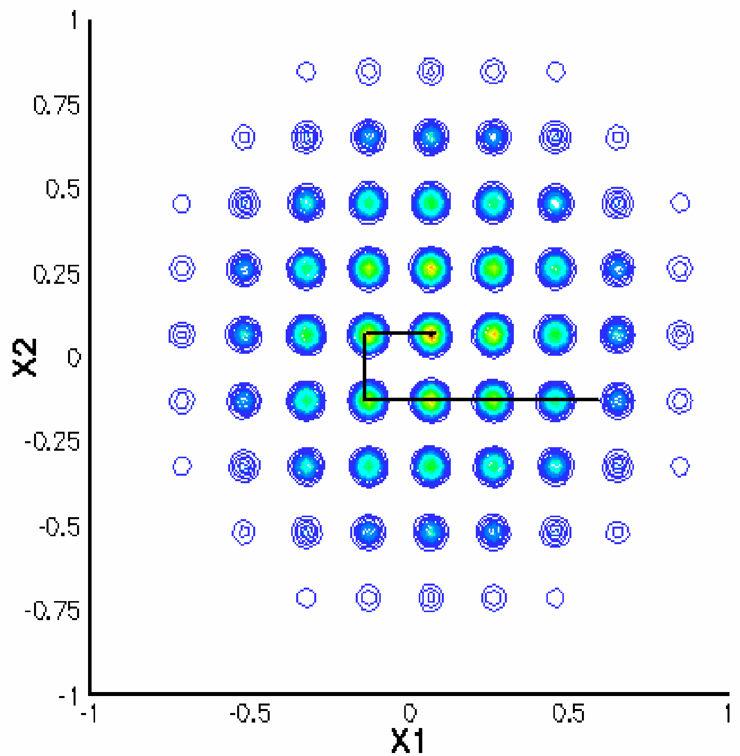
Optimal Function Value
Iteration No.
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1000
2000
3000
4000
5000
CLASSICAL SA METHOD

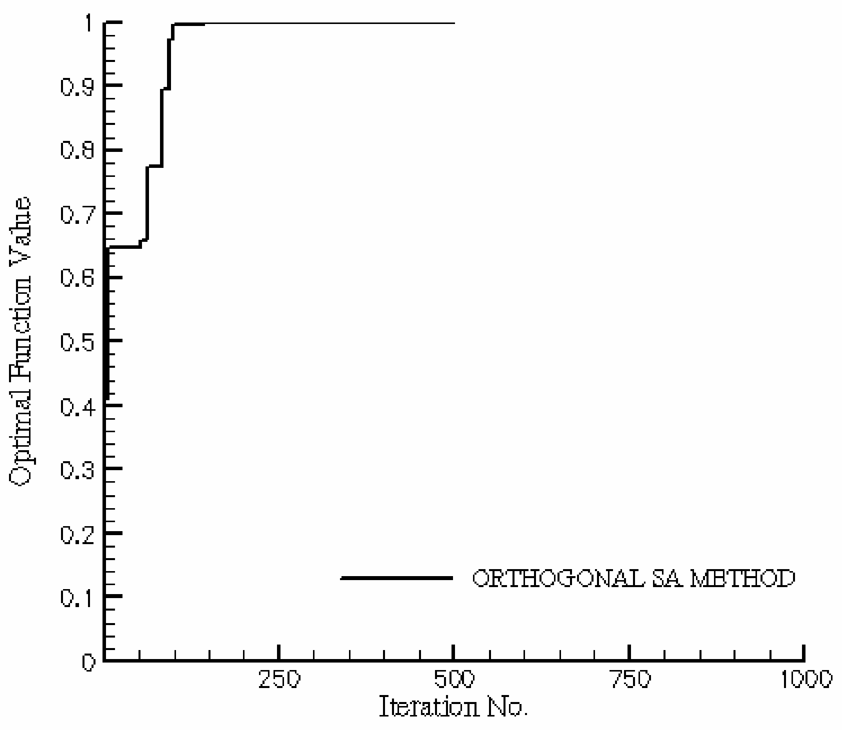
Single Point Search vs. Population-based Search



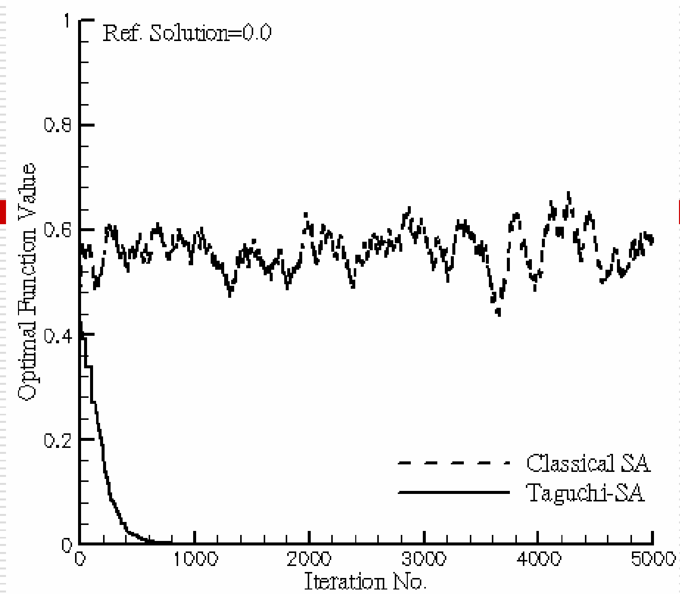


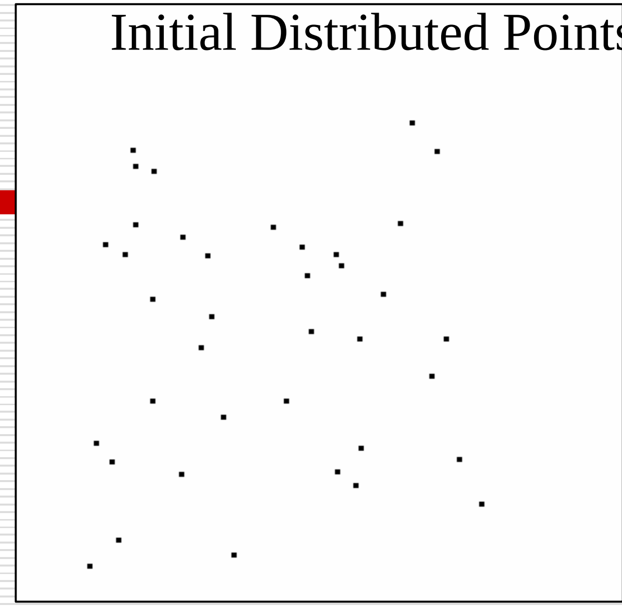
Teamwork

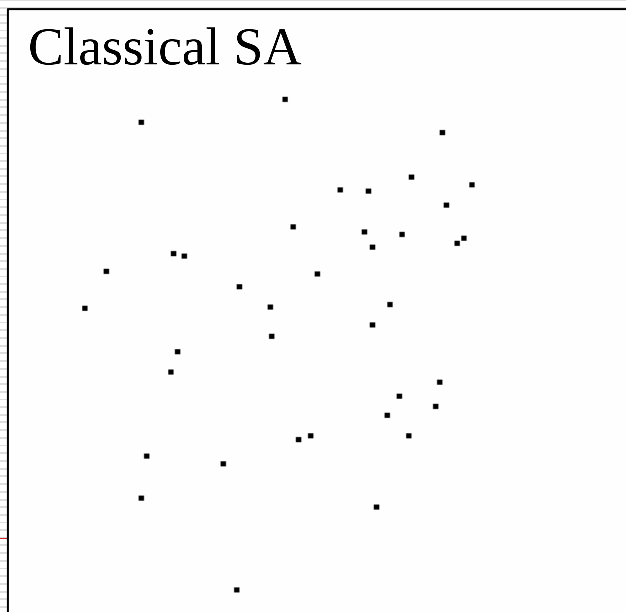

X2
X1
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
-1
-0.5
0
0.5
1

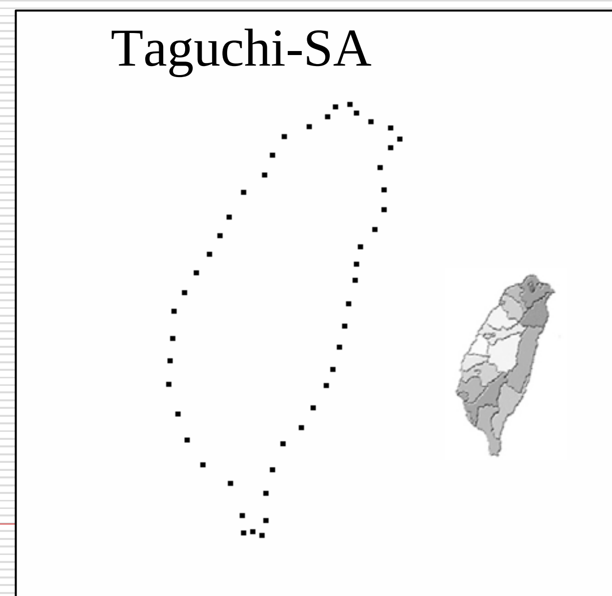

Optimal Function Value
Iteration No.
ORTHOGONAL SA METHOD
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
250
500
750
1000

Pattern Recognition

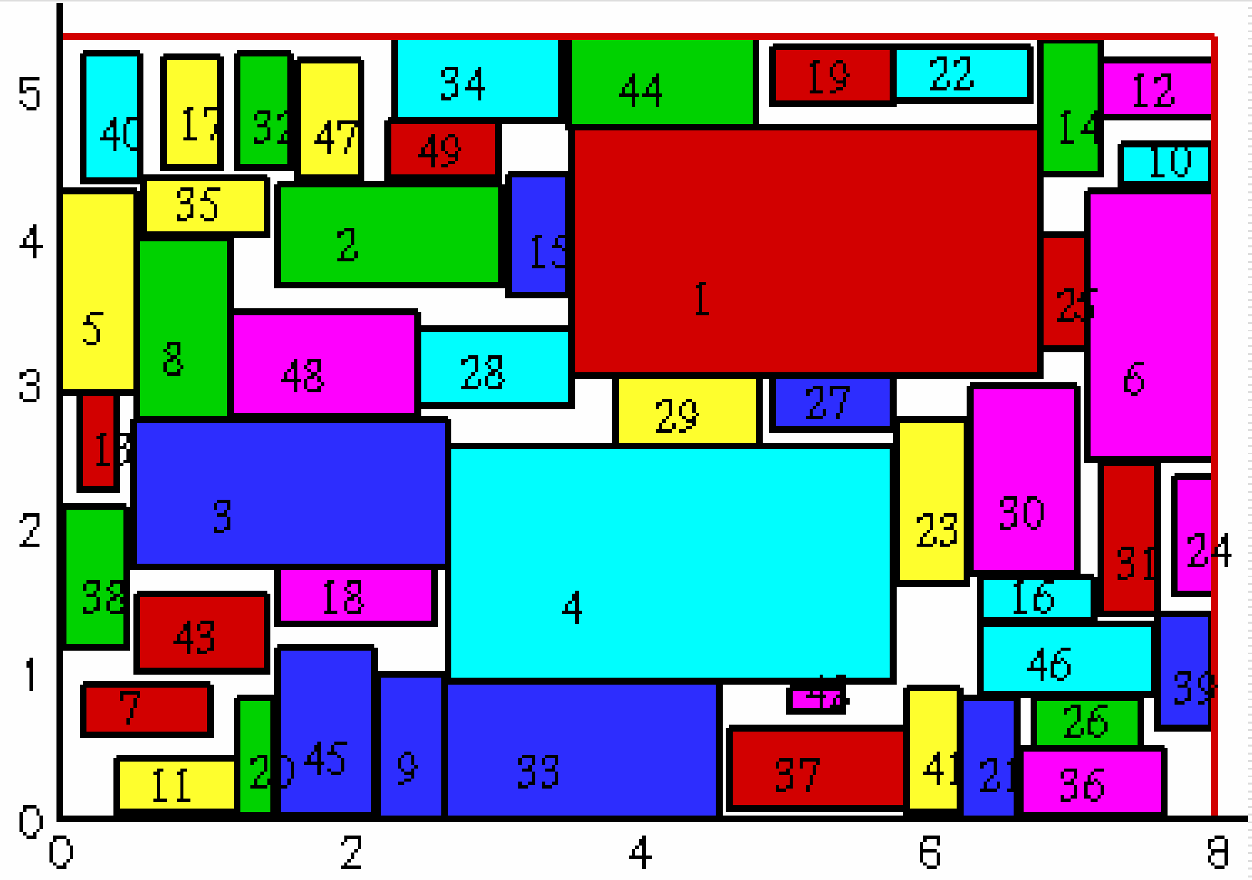








NP-hard Problem?

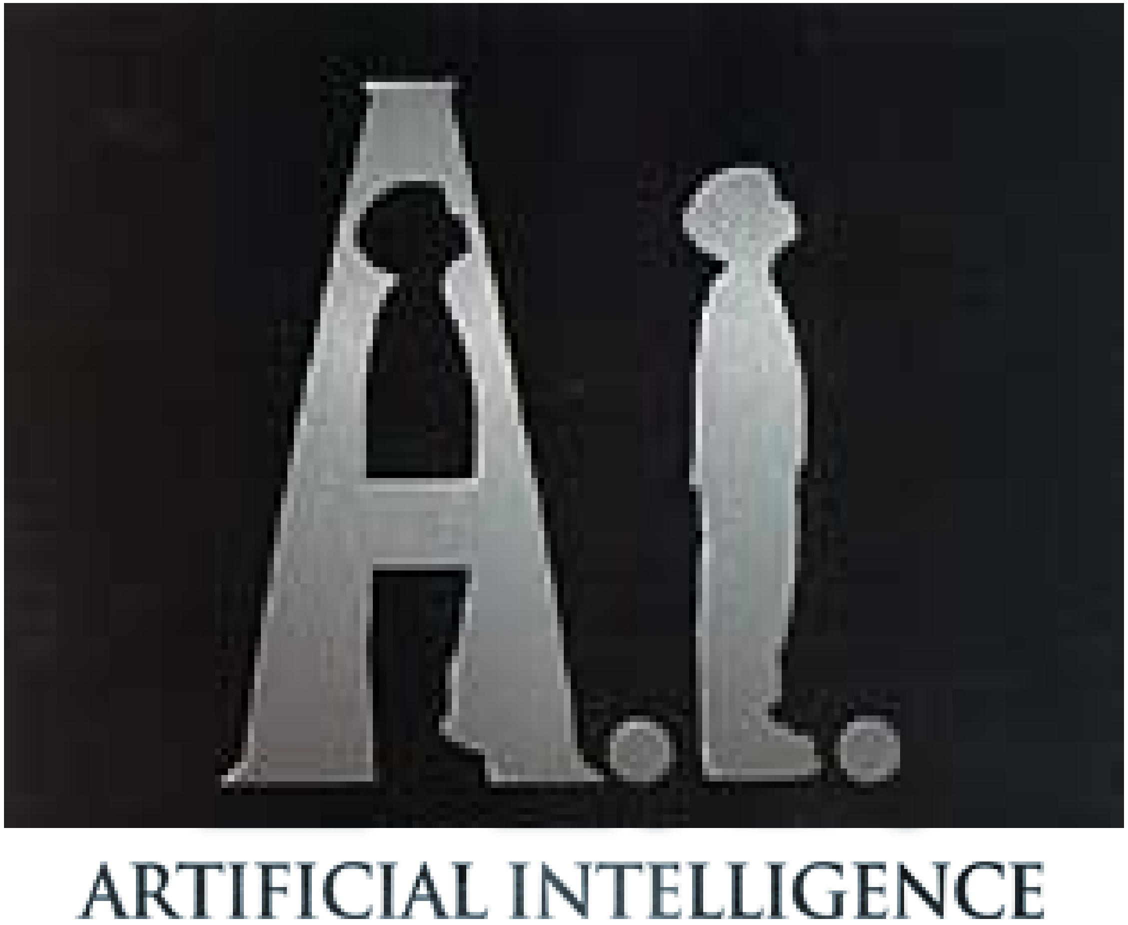


Circuit Ami49

Population-based Algorithm is better than single pt. search

Circuit	Modules	Total area of Modules	Opt	Comput. Results		Dead Space
				SA	OSA	
Apte	9	$46.65m^2$	47.31	49.73742	47.7613	2.52%
Xerox	10	$19.32m^2$	20.03	22.37095	20.31467	4.91%
Hp	11	$8.92m^2$	9.279	10.28119	9.58910	6.98%
Ami33	33	$1.16m^2$	1.197	1.39784	1.24716	6.99%
Ami49	49	$35.43m^2$	37.14	45.13331	38.5477	8.1%

Computational Intelligence



Metadata→Information→

Knowledge→Intelligence

What Is Intelligence?

Intelligence can be defined as the capability of a system to adapt its behaviour to ever-changing environment.

What is AI?

Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally*

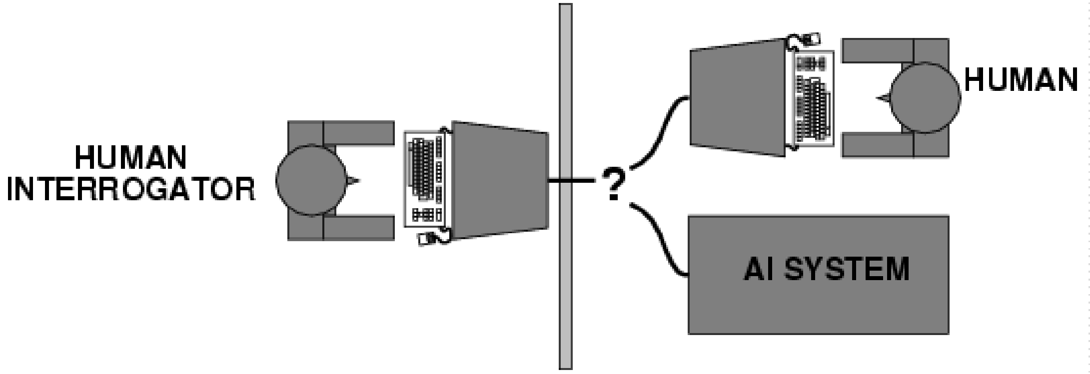
Intelligent Agent

Thinking humanly: cognitive modeling

- 1960s "cognitive revolution": information-processing psychology
- Requires scientific theories of internal activities of the brain
- How to validate? Requires
  - 1) Predicting and testing behavior of human subjects (top-down)
  - or 2) Direct identification from neurological data (bottom-up)
- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience)
- are now distinct from AI

Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence“;"Can machines think?" → "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game



- Suggested major components of AI: knowledge, reasoning, language understanding, learning

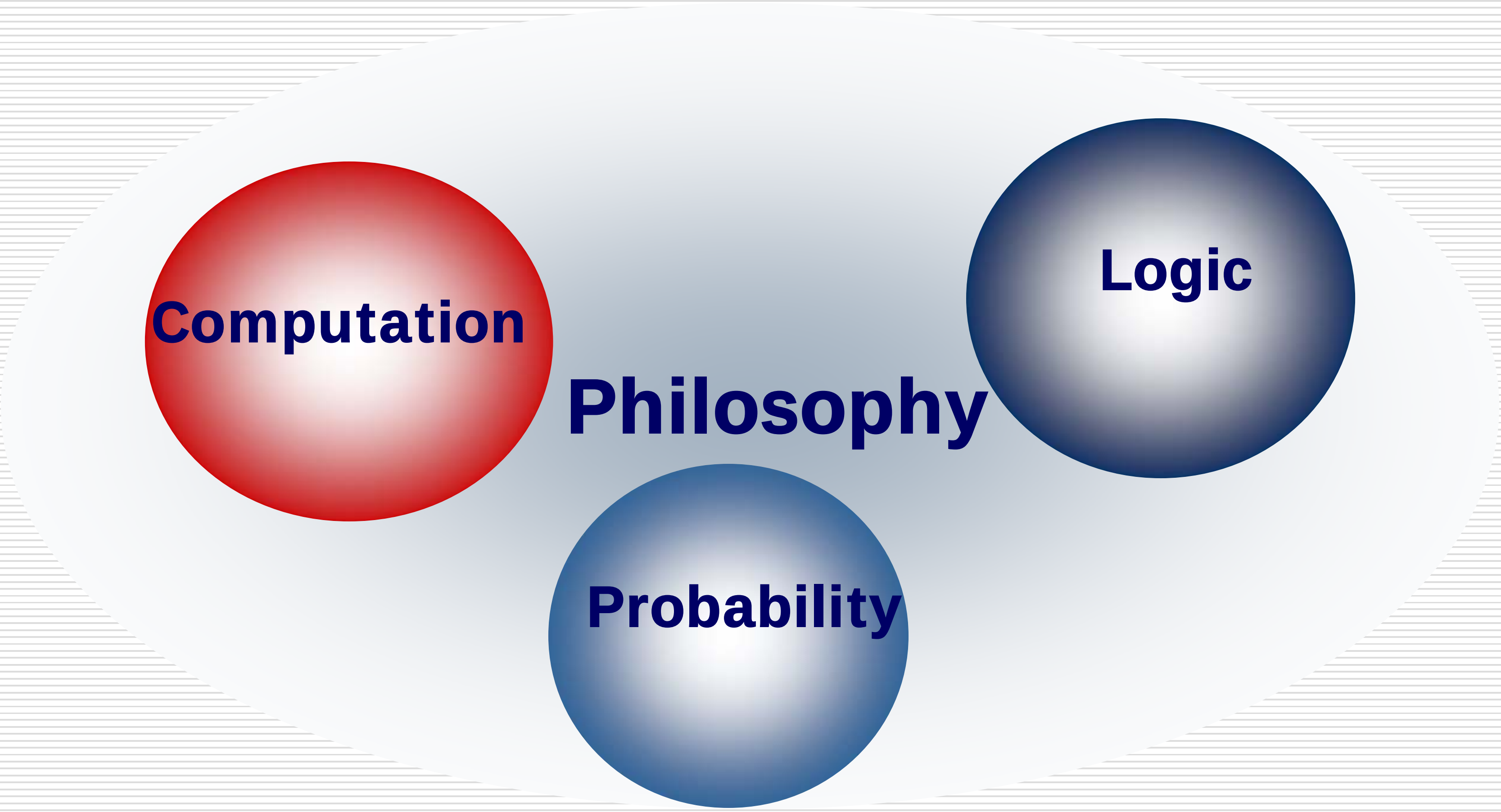
Thinking rationally: "laws of thought"

- Aristotle: what are correct arguments/thought processes?
- Several Greek schools developed various forms of *logic*: *notation* and *rules of derivation* for thoughts; may or may not have proceeded to the idea of mechanization
- Direct line through mathematics and philosophy to modern AI
- Problems:
  1. Not all intelligent behavior is mediated by logical deliberation
  2. What is the purpose of thinking? What thoughts should I have?

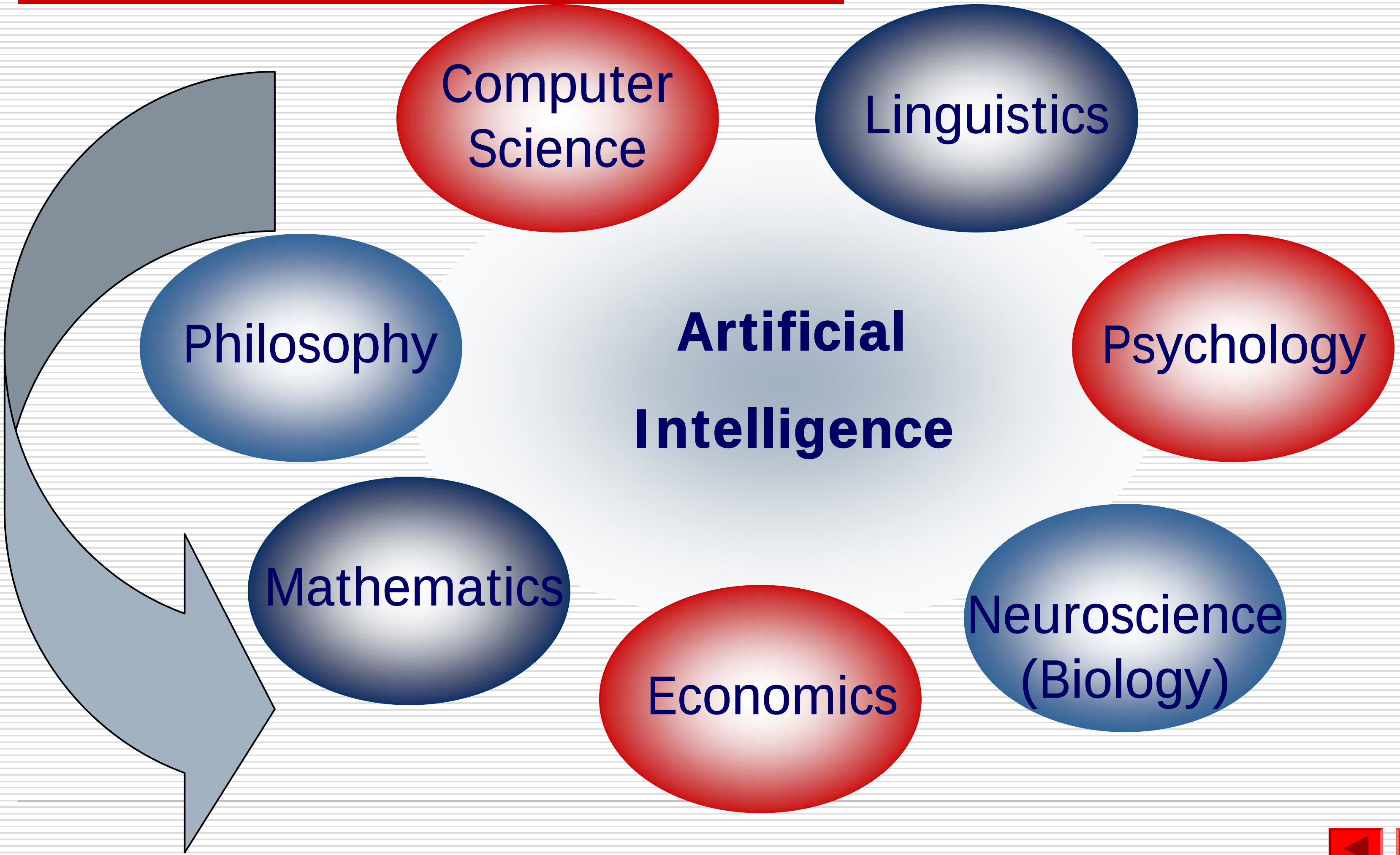
Acting rationally: rational agent

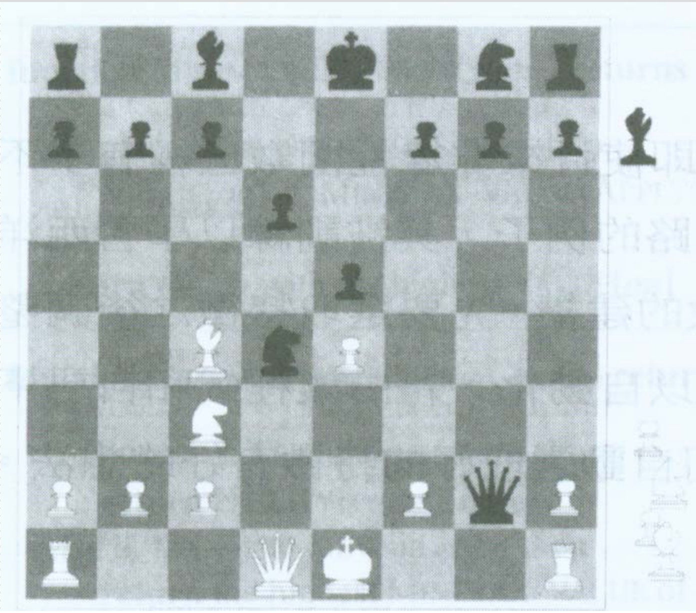
- Rational behavior: doing the right thing
- The right thing: that which is expected to maximize goal achievement, given the available information
- Doesn't necessarily involve thinking – e.g., blinking reflex – but thinking should be in the service of rational action

Main Components



Related Fields

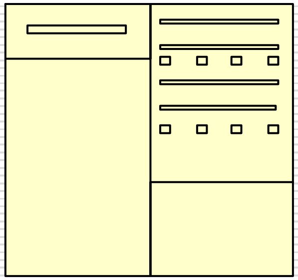




Garry Kasparov

1.Human

2.The world chess champion



1.Supercomputer Deep Blue

2.200 million positions/sec

3.Capable of learning

May 1997 (in New York)

Computer vs. Human

- Human
  - Flounders on complex computations
  - Capable of understanding and reasoning
  - More likely to understand the results and determine what to do next
- Machine
  - Performs precisely defined tasks with speed and accuracy
  - Not gifted with common sense

How Brains Think?

Cognition

Emotions

Being, Insight

?

Neurons

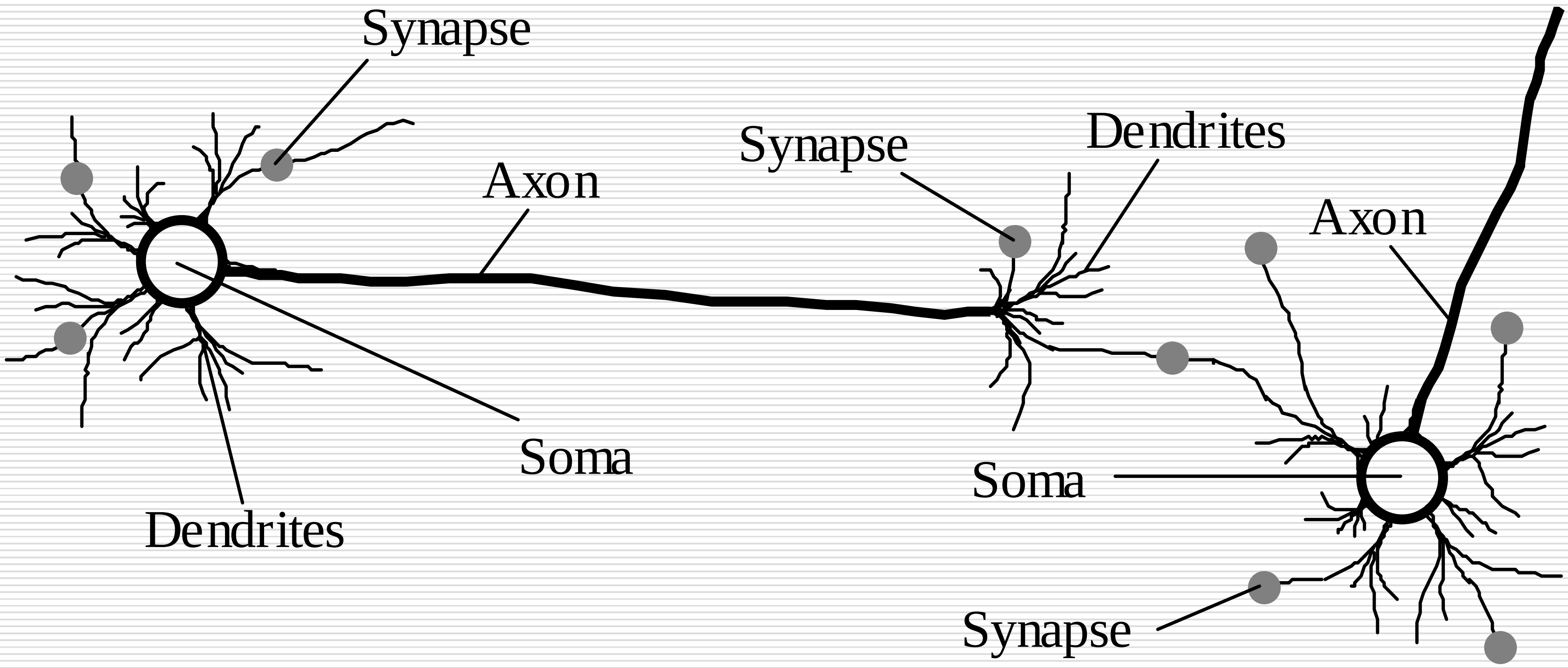
Synapses

Membranes

Bio Chemistry

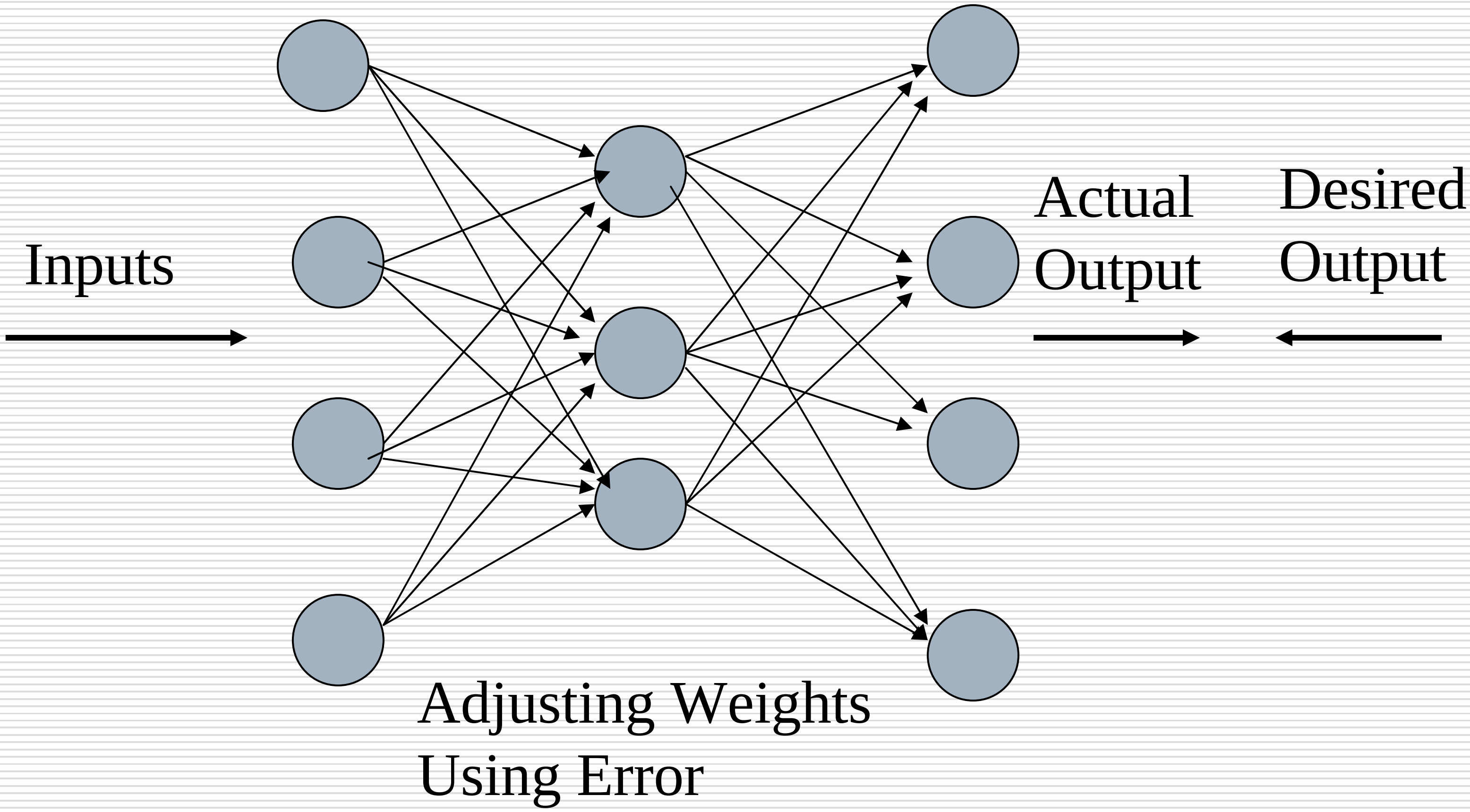
Chemical Bonds

Biological Neural Network



The human brain incorporates nearly 10 billion neurons and 60 trillion connections, synapse, between then.

Back Propagation Network



The goal of ***artificial intelligence*** as a science is to make machines do things that would require intelligence if done by humans.

AI = Machine Learning

AI = Function Approximation

AI = Stochastic Search

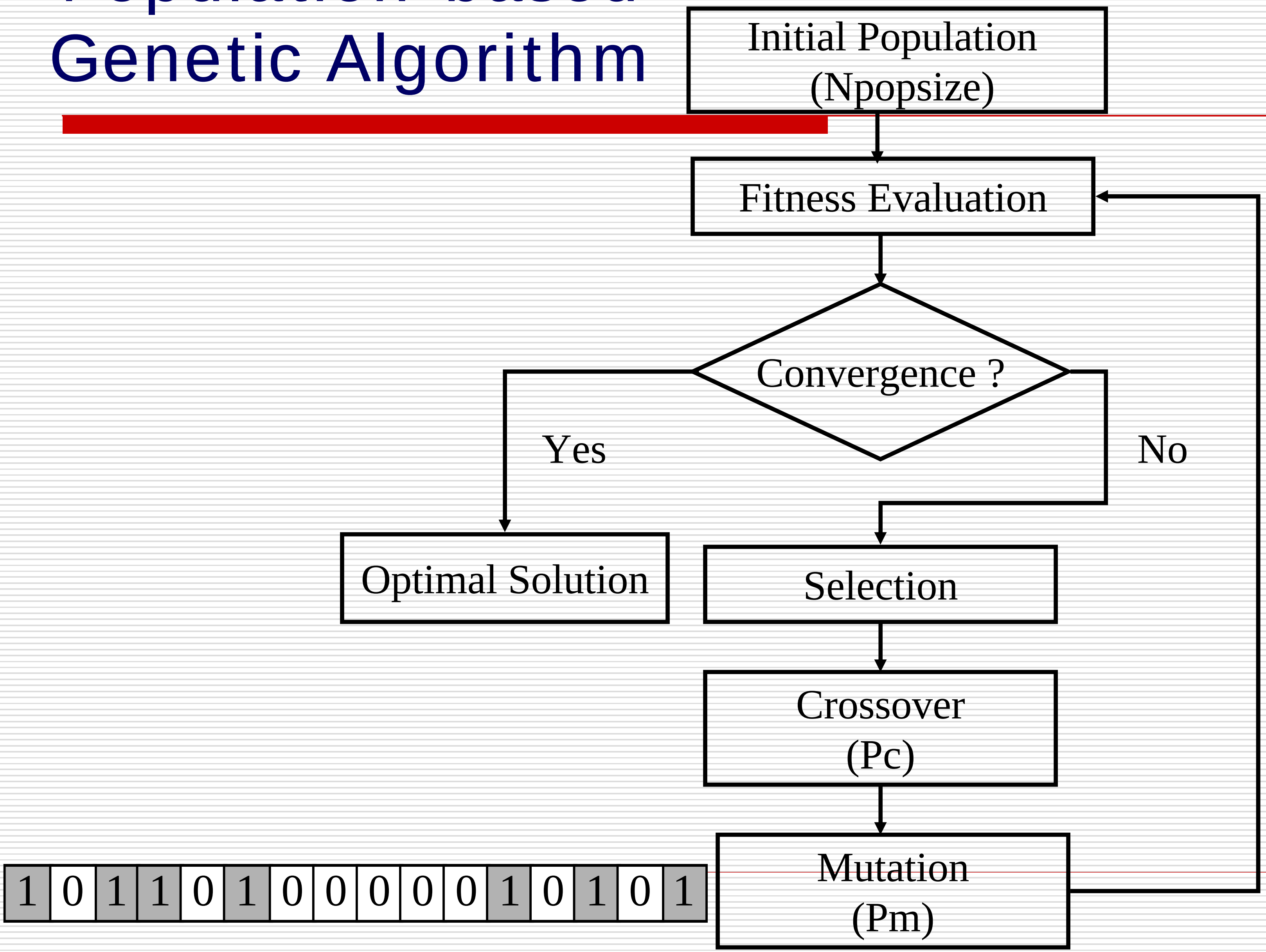
AI = Optimization

Can evolution be intelligent?

- Evolutionary computation simulates evolution on a computer. The result of such a simulation is a series of optimisation algorithms, usually based on a simple set of rules. Optimization iteratively improves the quality of solutions until an optimal, or at least feasible, solution is found.

- The behaviour of an individual organism is an inductive inference about some yet unknown aspects of its environment. If, over successive generations, the organism survives, we can say that this organism is capable of learning to predict changes in its environment.

Population-based Genetic Algorithm



Applications of GA

- image processing;
- laser technology;
- Aeronautics;
- artificial neural networks;
- control;
- robotics;
- prediction of 3-D protein structures;
- VLSI electronic chip layouts;
- laser technology;
- tracking system
- medicine;
- jobshop scheduling;
- facial recognition ;

...

Schema Theory

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{\bar{f}}\left[1-p_c\frac{\delta(H)}{l-1}\right][1-O(H)\times p_m]$$

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{\bar{f}}\left[1-p_c\frac{\delta(H)}{l-1}-O(H)\times p_m\right]$$

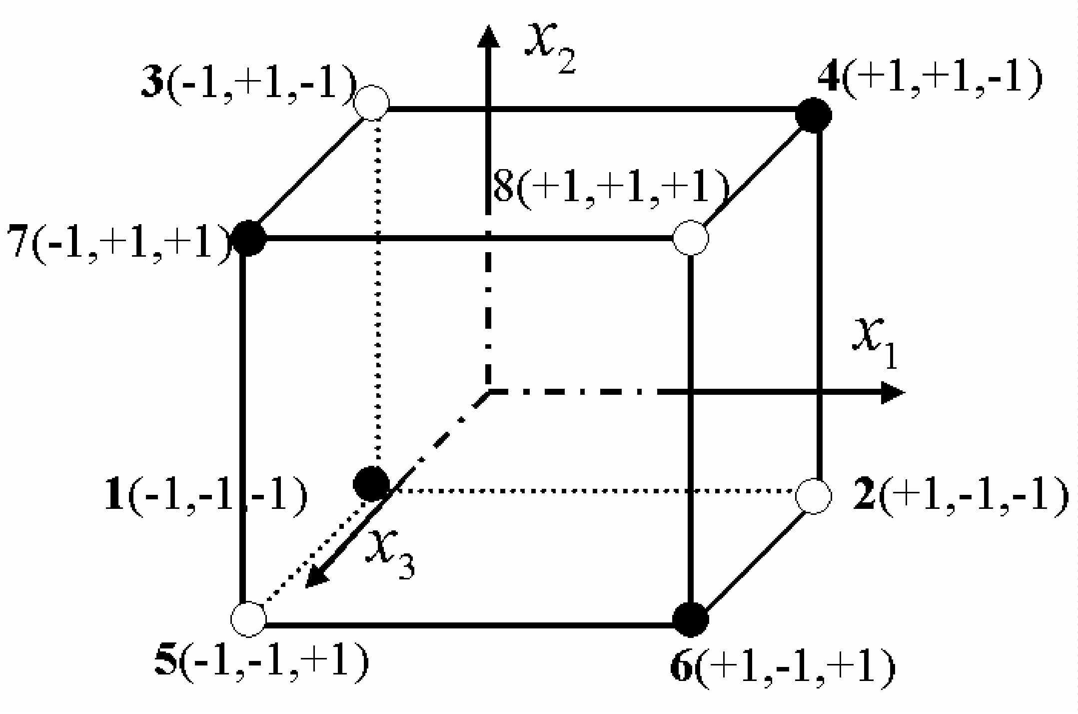
$$m(H,t+1) \geq m(H,0)(1+c)^t\left[1-p_c\frac{\delta(H)}{l-1}-O(H)\times p_m\right]$$

This equation describes the growth of a schema from one generation to the next.

Some Applications on the Optimization

Intelligent Genetic Algorithm

```
Procedure: Intelligent Genetic Algorithm
Begin
create initial population randomly;
do {
choose parent1 and parent2 from population;
 /* REPRODUCTION */
offspring=Intelligent CROSSOVER(parent1, parent2);
 /* Fractional Factorial Design */
MUTATION(offspring);
    } while (stopping criterion not satisfied);
End;
```


x_2
3(-1,+1,-1)
4(+1,+1,-1)
8(+1,+1,+1)
7(-1,+1,+1)
x_1
1(-1,-1,-1)
2(+1,-1,-1)
x_3
5(-1,-1,+1)
6(+1,-1,+1)

Fractional Factorial Design

Fractional Factorial Design

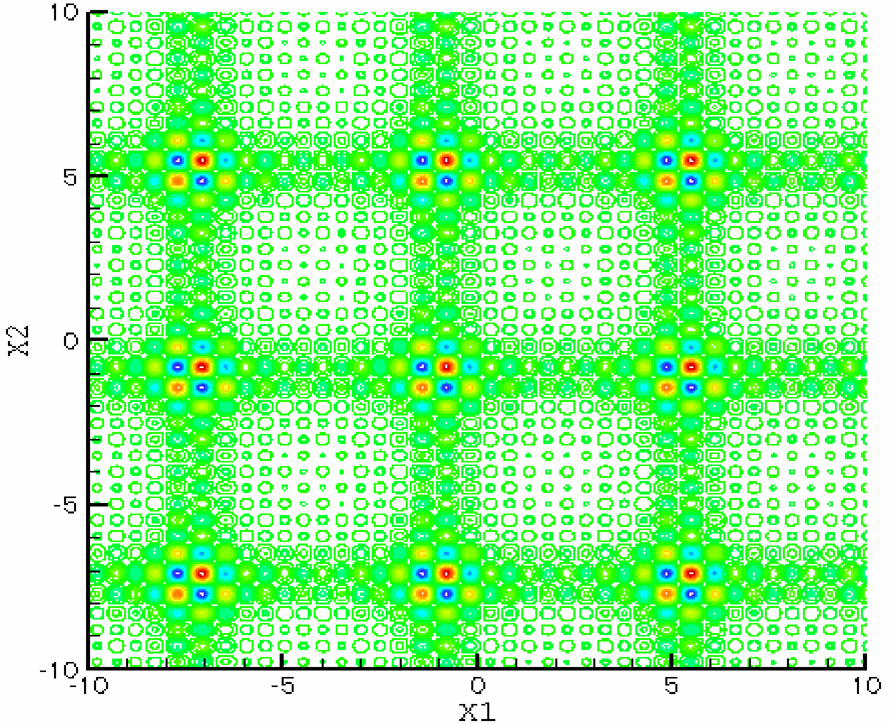
Factor	A	B	AB	C	AC	BC	ABC	Output
Factor Level	$x_1^{\pm}$	$x_2^{\pm}$	$x_3^{\pm}$	$x_4^{\pm}$	$x_5^{\pm}$	$x_6^{\pm}$	$x_7^{\pm}$	
Cell 1	-	-	+	-	+	+	-	F_1
2	+	-	-	-	-	+	+	F_2
3	-	+	-	-	+	-	+	F_3
4	+	+	+	-	-	-	-	F_4
5	-	-	+	+	-	-	+	F_5
6	+	-	-	+	+	-	-	F_6
7	-	+	-	+	-	+	-	F_7
8	+	+	+	+	+	+	+	F_8

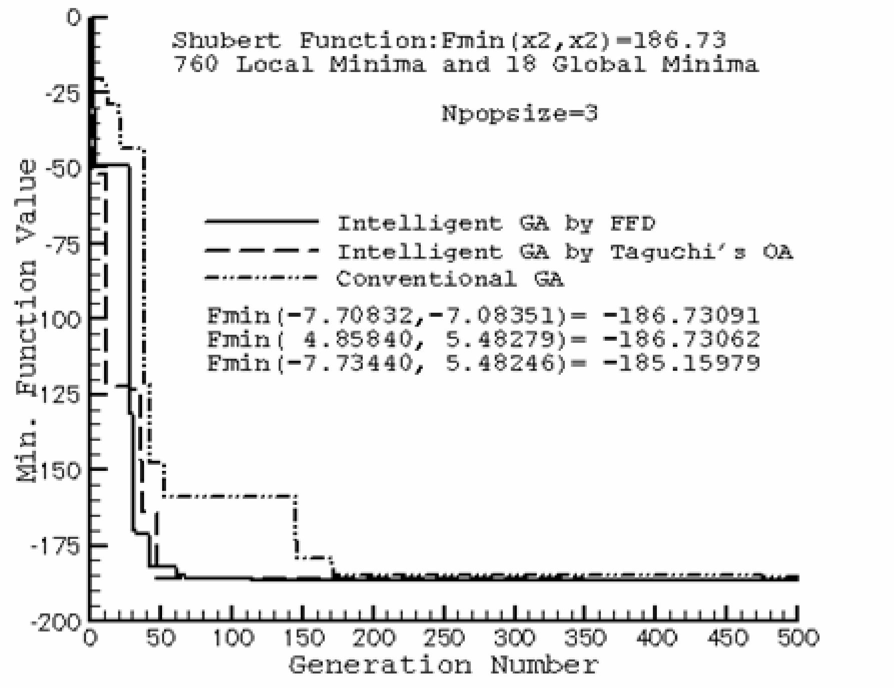
Taguchi's Orthogonal Array

Factor / Exp. No.	1	2	3	4	5	6	7	Output
1	1	1	1	1	1	1	1	F_1
2	1	1	1	2	2	2	2	F_2
3	1	2	2	1	1	2	2	F_3
4	1	2	2	2	2	1	1	F_4
5	2	1	2	1	2	1	2	F_5
6	2	1	2	2	1	2	1	F_6
7	2	2	1	1	2	2	1	F_7
8	2	2	1	2	1	1	2	F_8

$$(x_1^-, x_2^-, x_3^+, x_4^-, x_5^+, x_6^+, x_7^-) \quad (x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, x_4^{(1)}, x_5^{(1)}, x_6^{(1)}, x_7^{(1)})$$

Shubert Function





$$\text{Maximize} \quad F(x_1, x_2) = \prod_{i=1}^{2} \{\sin^6(5.1\pi\ x_i + 0.5) \times \exp(-4\ln 2(x_i - 0.0667)^2 / 0.64)\}$$

$$\text{Subject to} \quad 0 \le x_i \le 1, i = 1,2$$

IGA-LS Algorithm

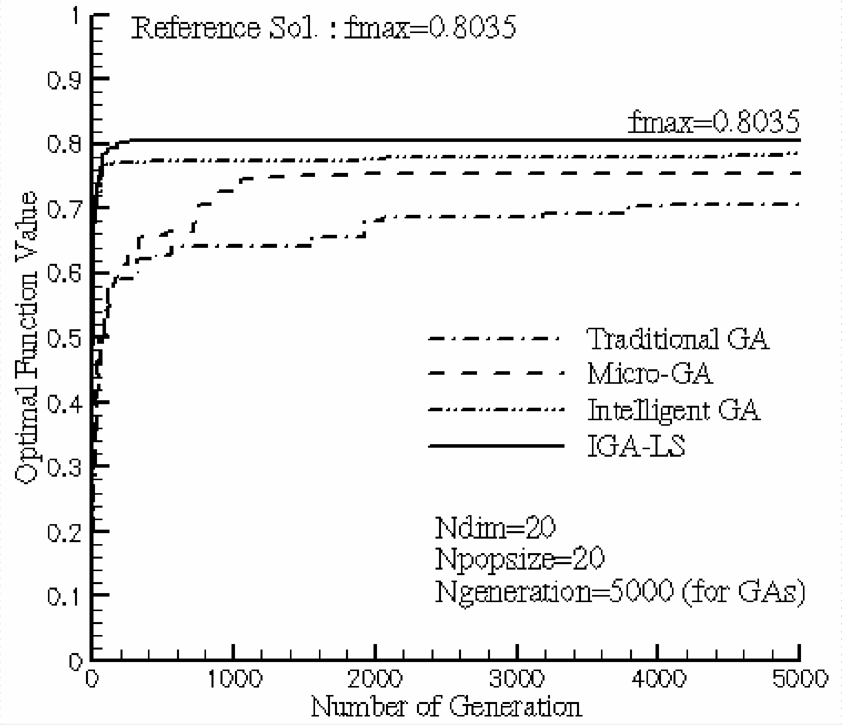
```
Begin
create initial population randomly;
do {
choose a pair of parents from population;
/* SELECTION OPERATOR*/
children=INTELLIGENT CROSSOVER(parent1,parent2);
children=MUTATION(children);
input the variables from the output data of GA;
gradient-based LS approach(children);
} while (stopping criterion of GA not reached);
parents ← children;
End;
```

$$\text{Maximize } f(\vec{x}) = \left| \frac{\sum_{i=1}^{N} \cos^4(x_i) - 2\prod_{i=1}^{N} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{N} i x_i^2}} \right|$$

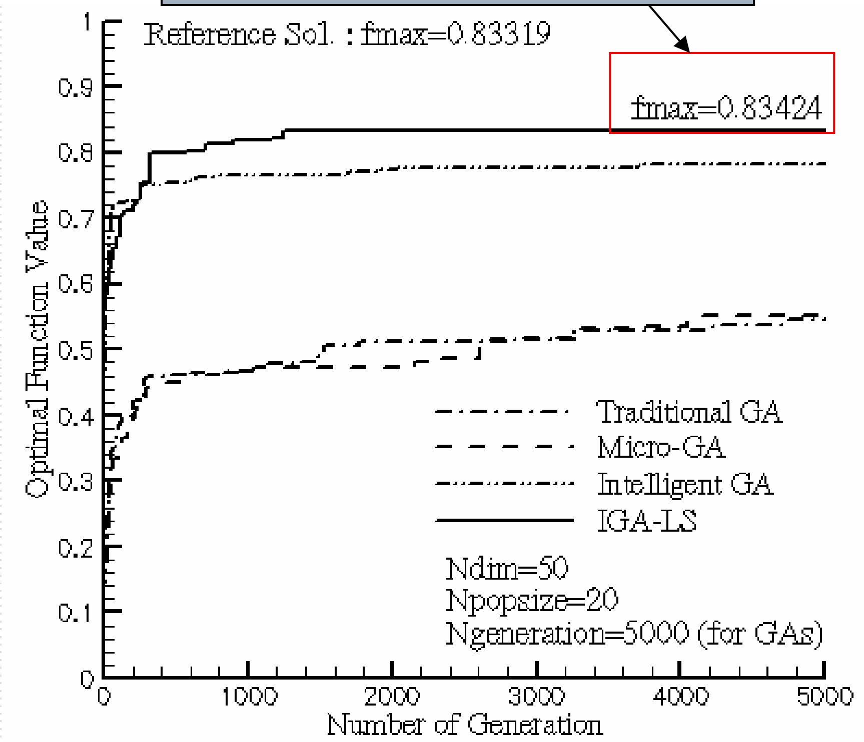
$$\text{Subject to } \prod_{i=1}^{N} x_i > 0.75, \quad \sum_{i=1}^{N} x_i < 7.5N$$

$$0 < x_i < 10$$

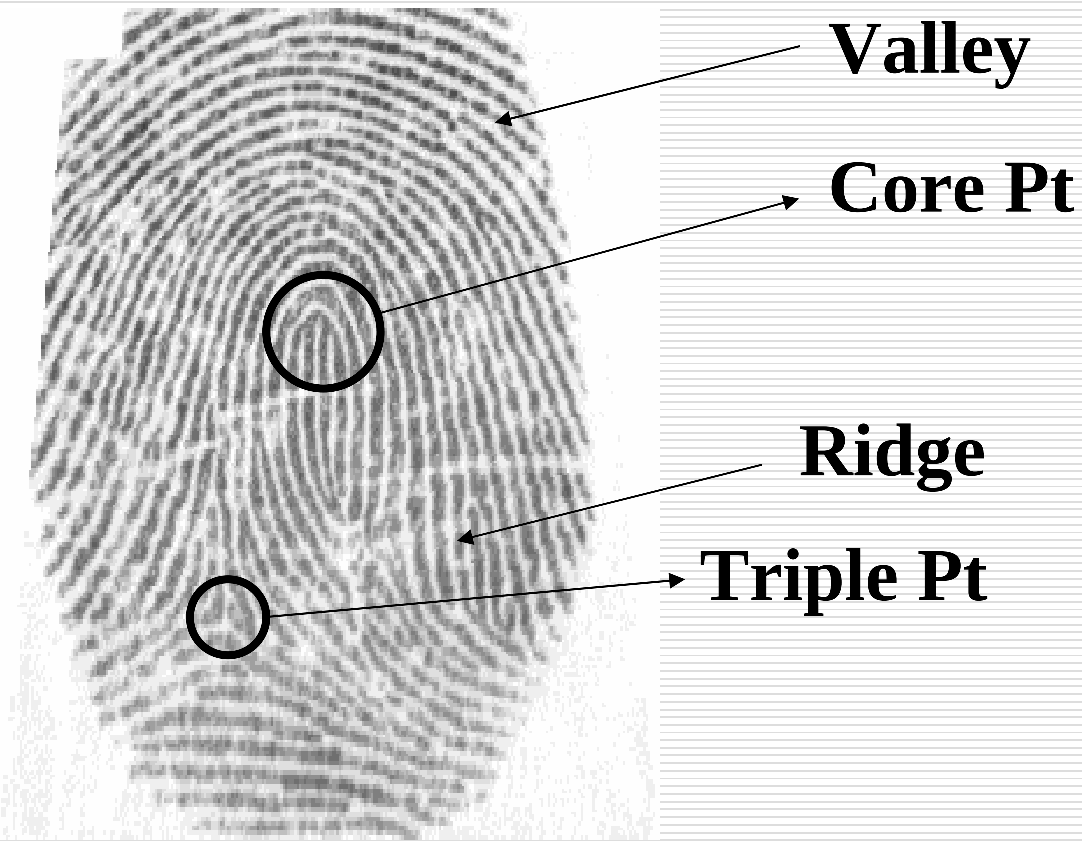
Nonlinear Function with Two Inequality Constraints



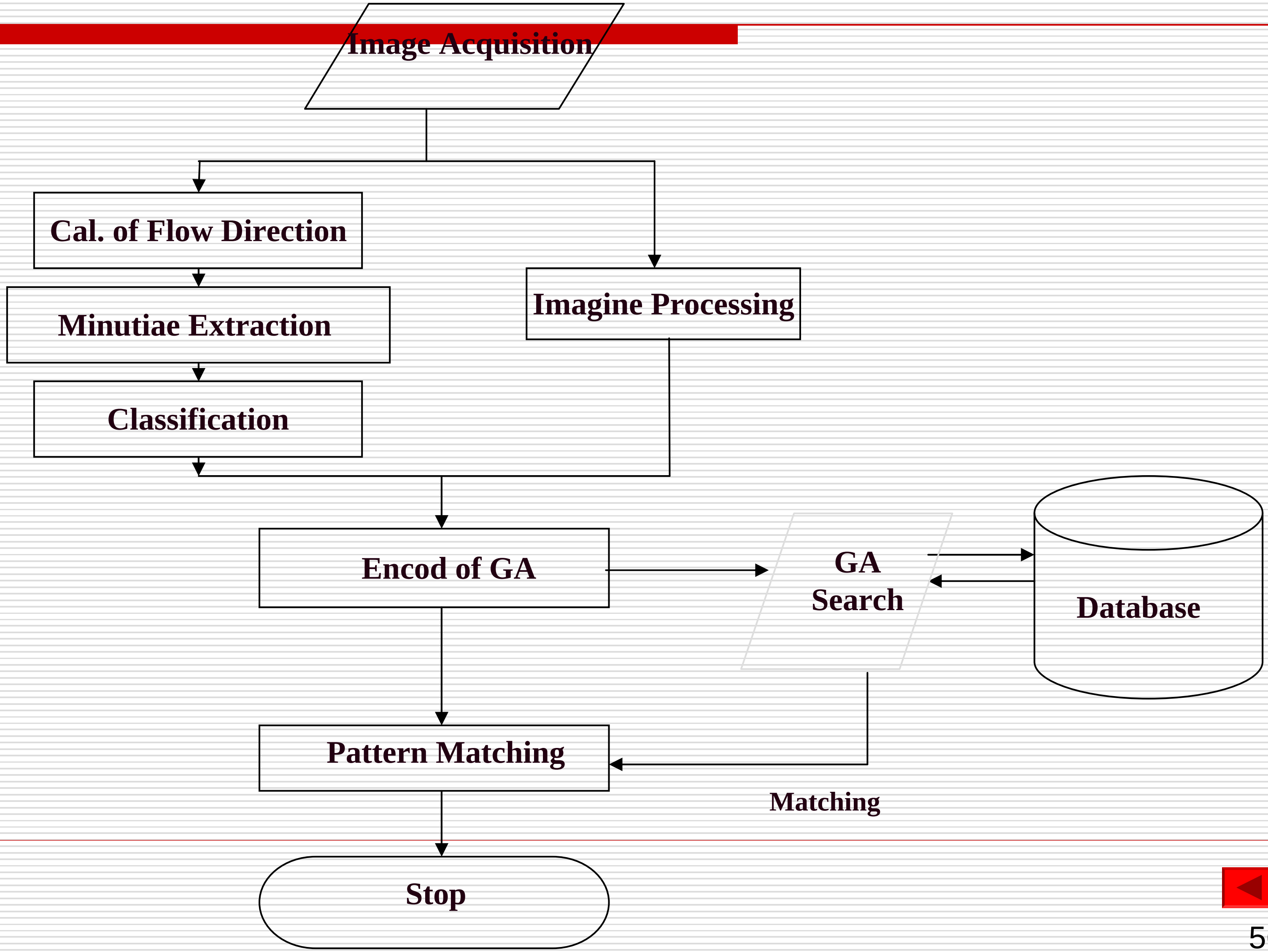
Best Known Solution



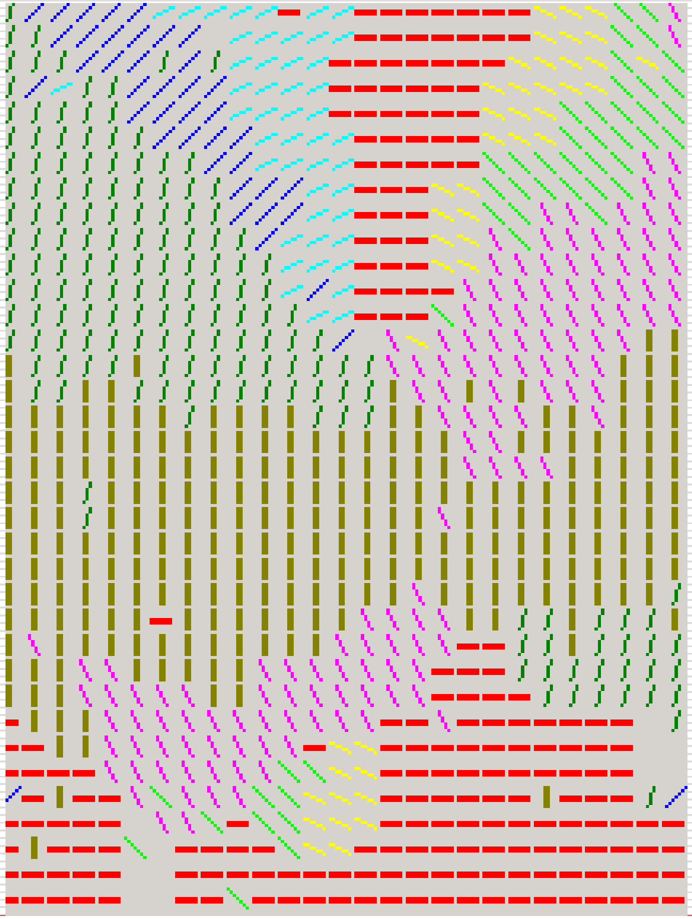
Fingerprint Identification System

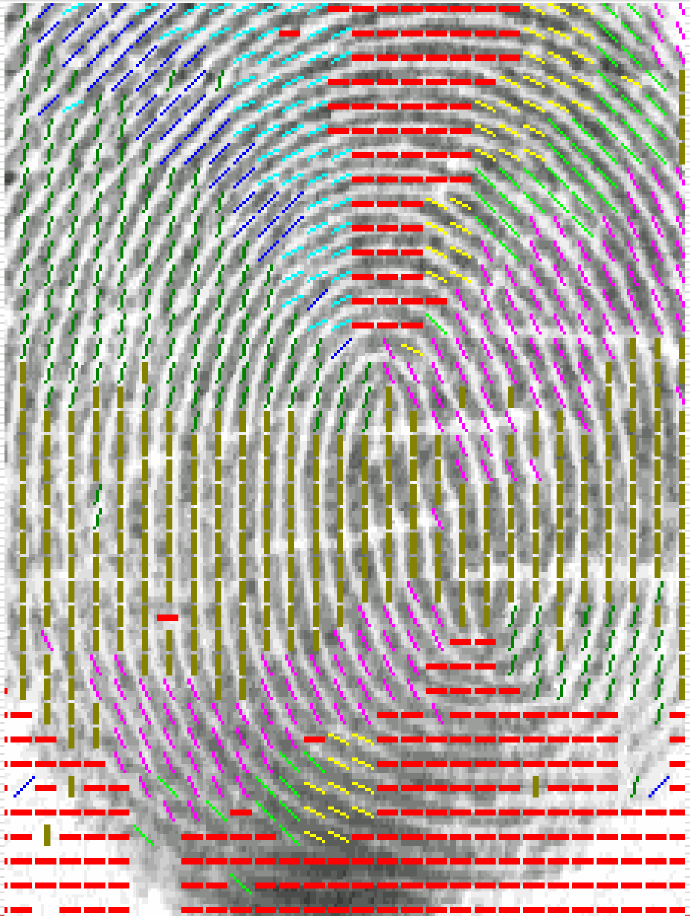


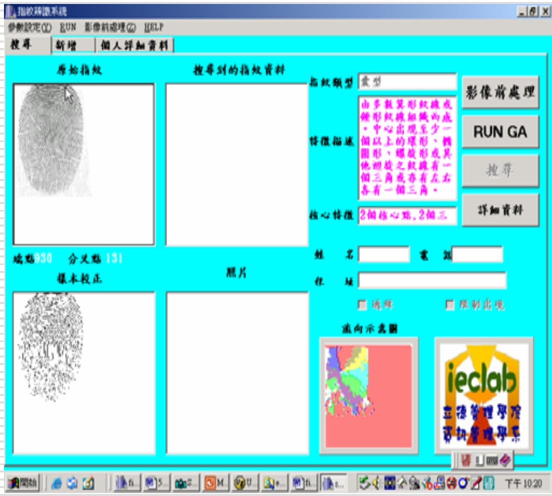
FlowChart of the Fingerprint Identification System

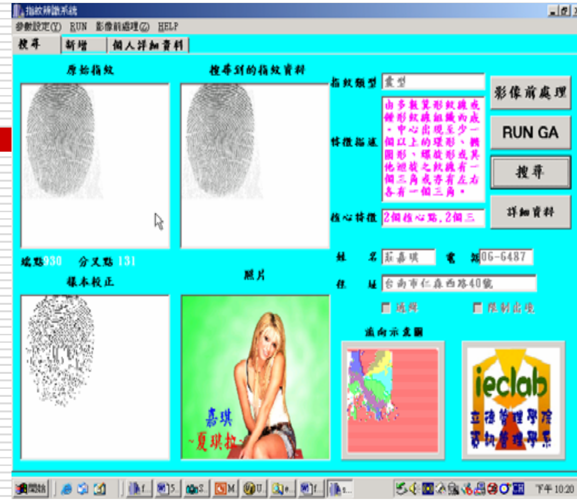


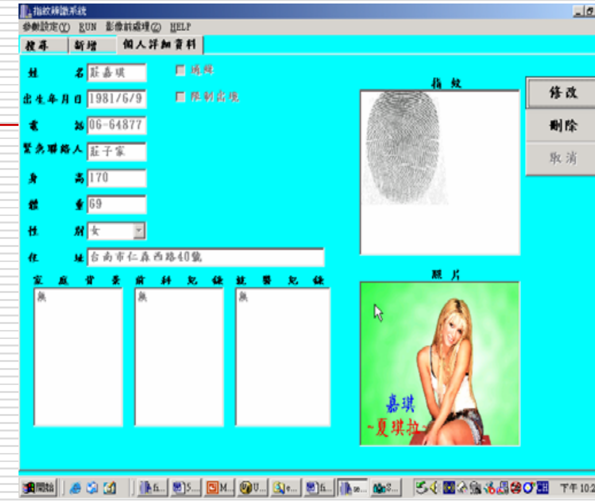
Calculation of Flow Direction

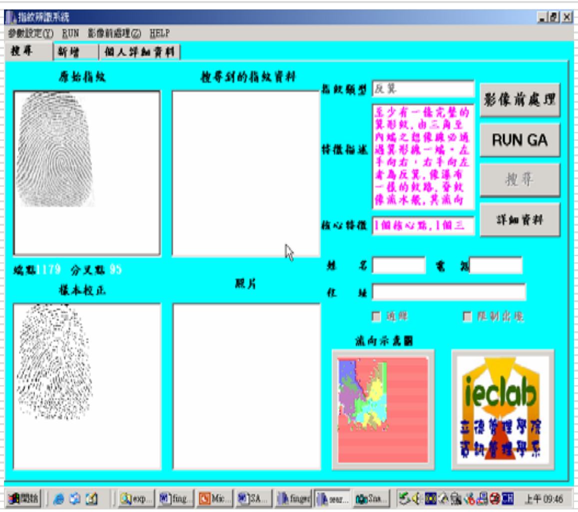


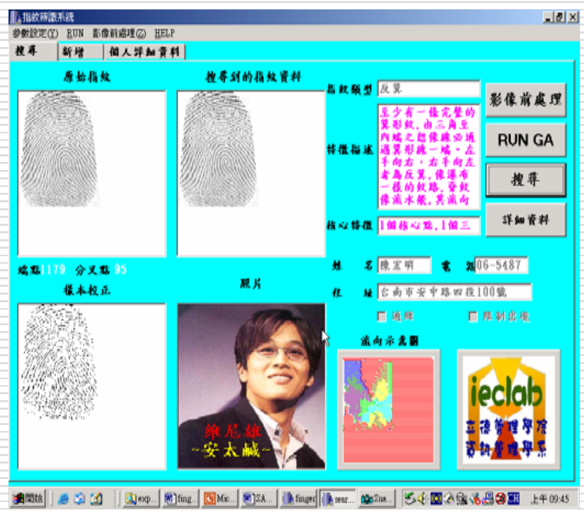


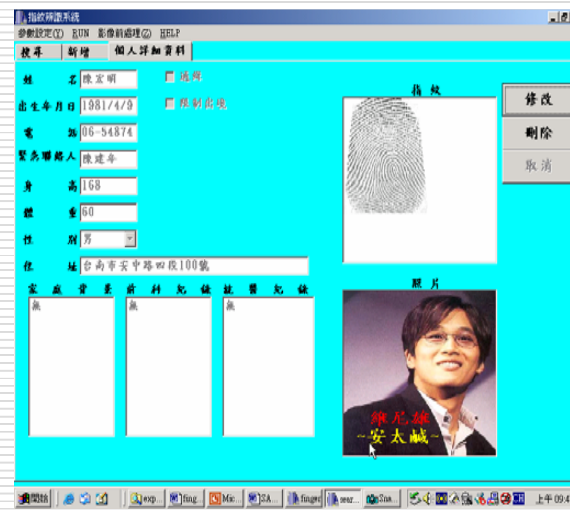

原始指紋
搜尋到的指紋資料
影像前處理
RUN GA
搜尋
詳細資料
照片
流向示意圖
ieclab


原始指紋
搜尋到的指紋資料
影像前處理
RUN GA
搜尋
詳細資料
照片
流向示意圖
ieclab


個人詳細資料
指紋
修改
刪除
取消
照片


原始指紋
搜尋到的指紋資料
影像前處理
RUN GA
搜尋
詳細資料
照片
流向示意圖
ieclab


原始指紋
搜尋到的指紋資料
影像前處理
RUN GA
搜尋
詳細資料
照片
流向示意圖
ieclab


個人詳細資料
指紋
修改
刪除
取消
照片

Multi-Objective OPtimization

- Wide range of problems can be categorised by the presence of a number of *n* possibly conflicting objectives:
  - buying a car: speed vs. price vs. reliability
  - engineering design: lightness vs. strength
  - wireless transmitter placement: cost vs. coverage
- Two part problem:
  - finding set of good solutions
  - choice of best for particular application

Example:

In VLSI design: Appropriately place the functional blocks on the chip such that chip area is minimized and power dissipation is also minimized.

Trade off ? *Lower chip area* ⇨ *higher power dissipiation*

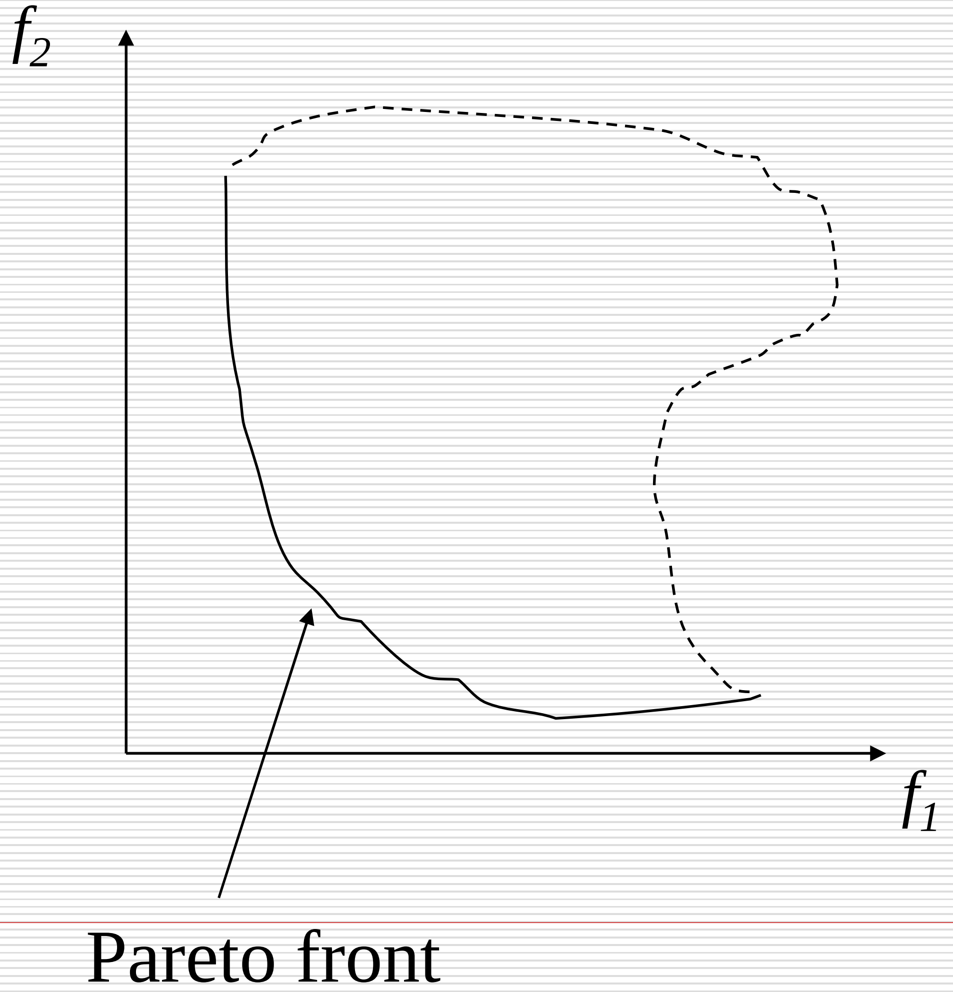
So where is the compromize?

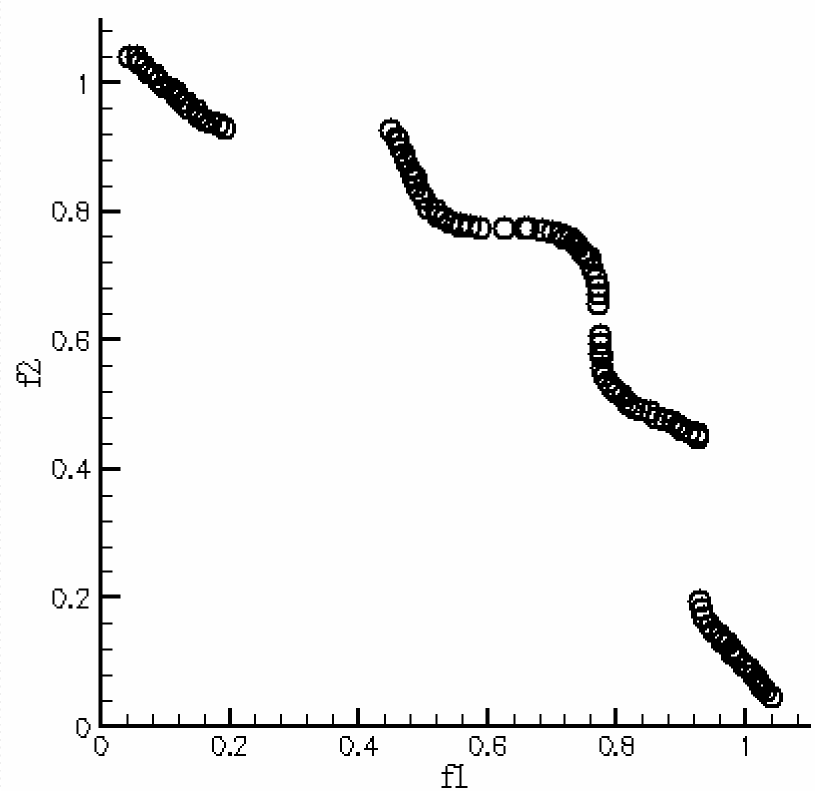
• Difficulty?

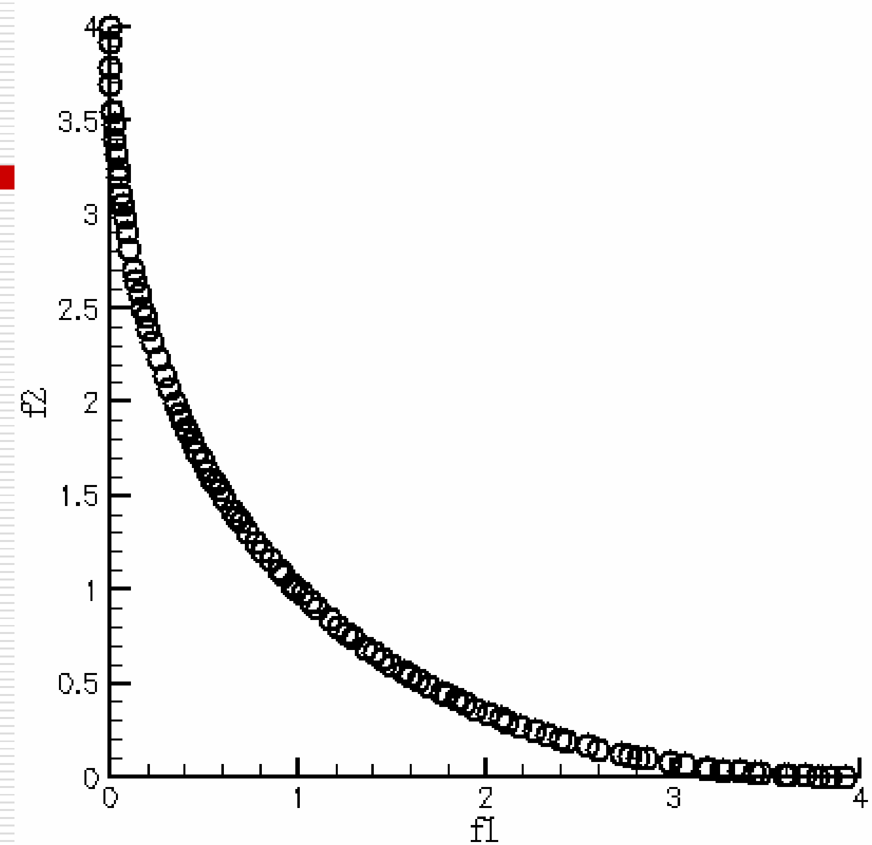
No *accepted* definition of optimum.

In such situations optimizing the different criteria means assigning values to them which are *acceptable* to the designer.

Pareto Front







$$f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}) = x_2$$

$$0 < x_i \le \pi \,, i = 1,2$$

$$g_1(\vec{x}) = x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\tan^{-1}\frac{x_1}{x_2}\right) \ge 0$$

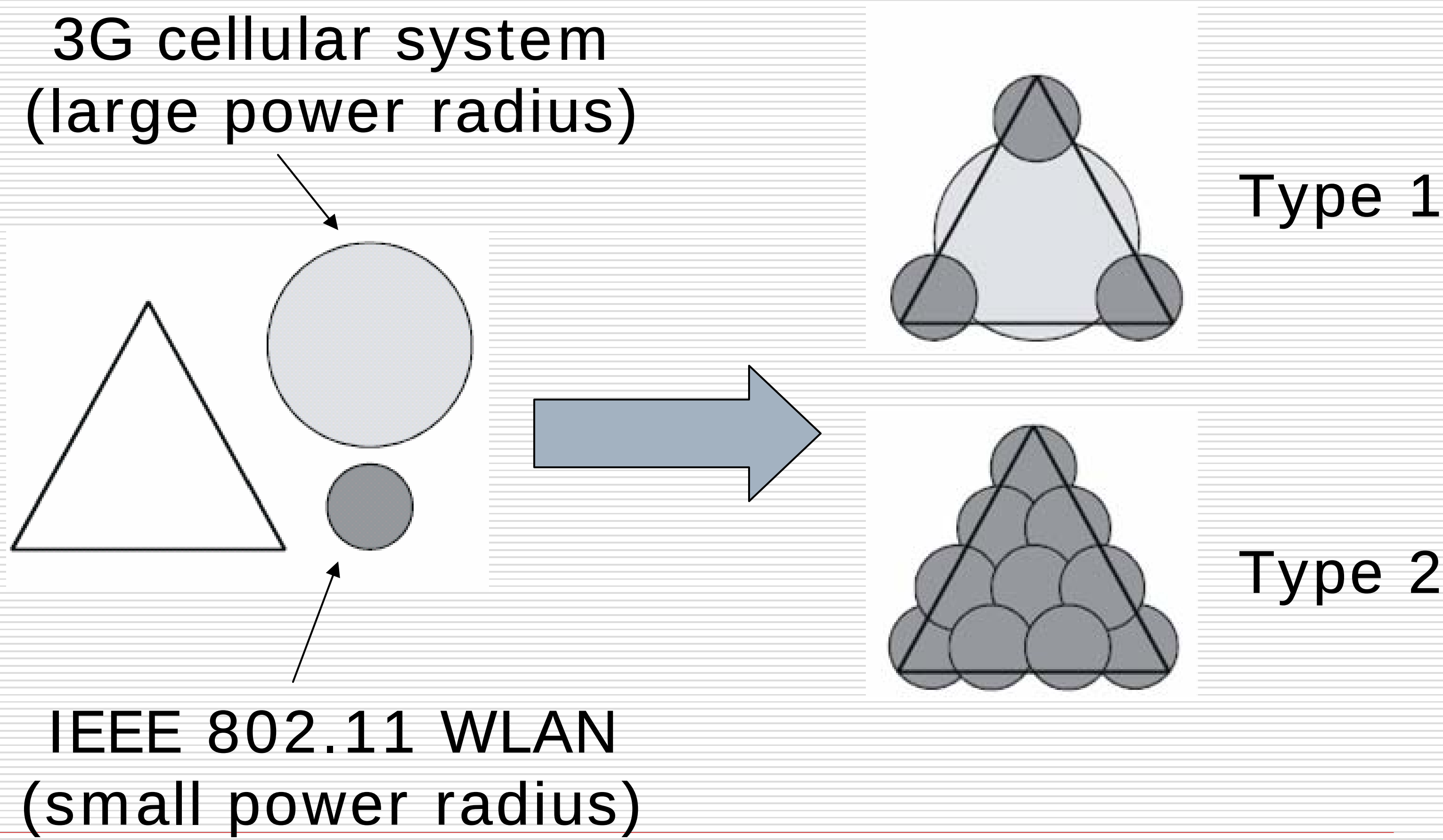
$$g_2(\vec{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \le 0.5$$

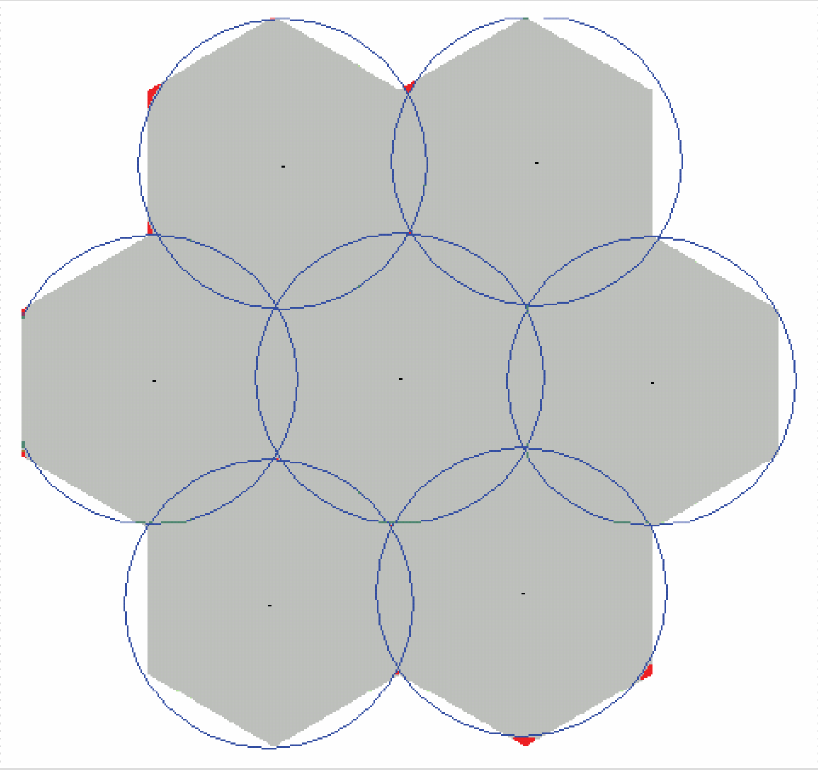
$$f_1(\vec{x}) = x^2$$

$$f_2(\vec{x}) = (x-2)^2$$

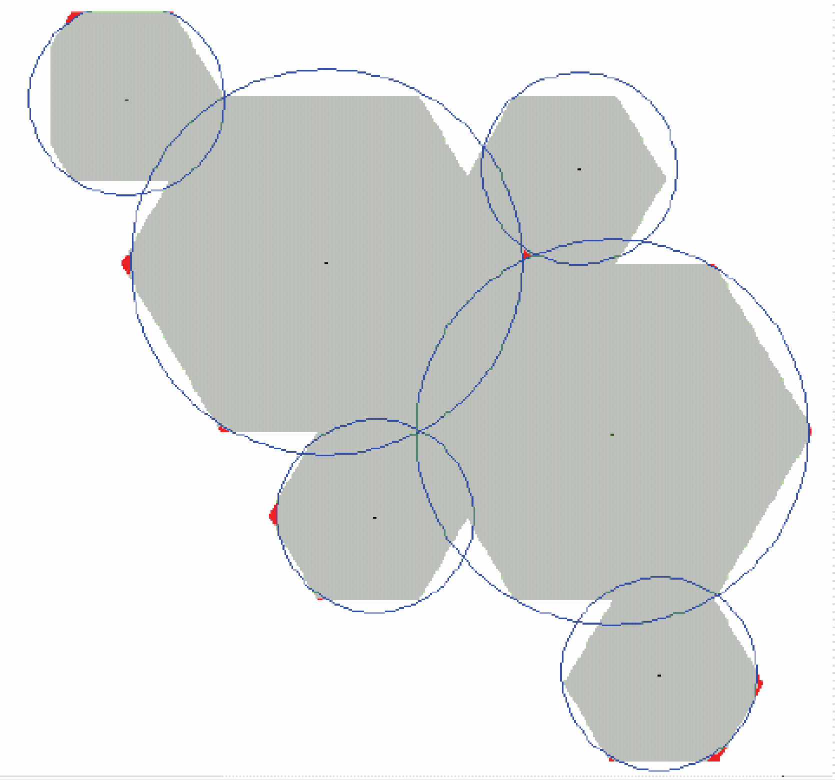
$$-1000.0 < x_i \le 1000.0 \,, i = 1,2$$

Wireless Transmitter Placement





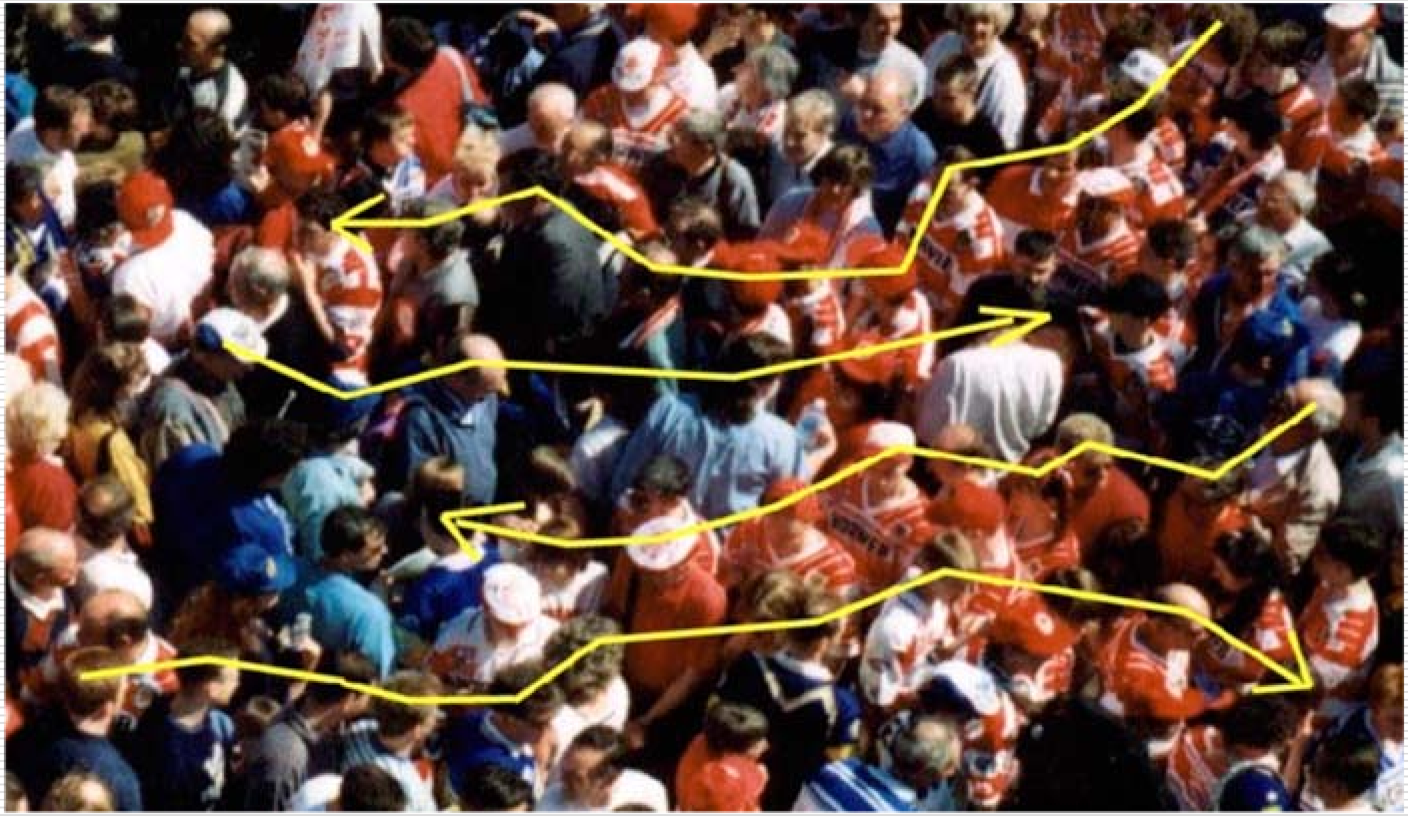
Type 1



Type 2

Swarm Intelligence

Intelligent Human Swarms



Particle Swarm Optimization

- Simulating social behavior
- The original PSO algorithm, proposed by Kennedy and Eberhart in 1995, is based on visualizing the movement of organisms in a flock of birds or schools of fish, and attempts to model the individual's behavior of exploration and exploitation, in the hope of simulating the search behavior of a swarm seen in nature.

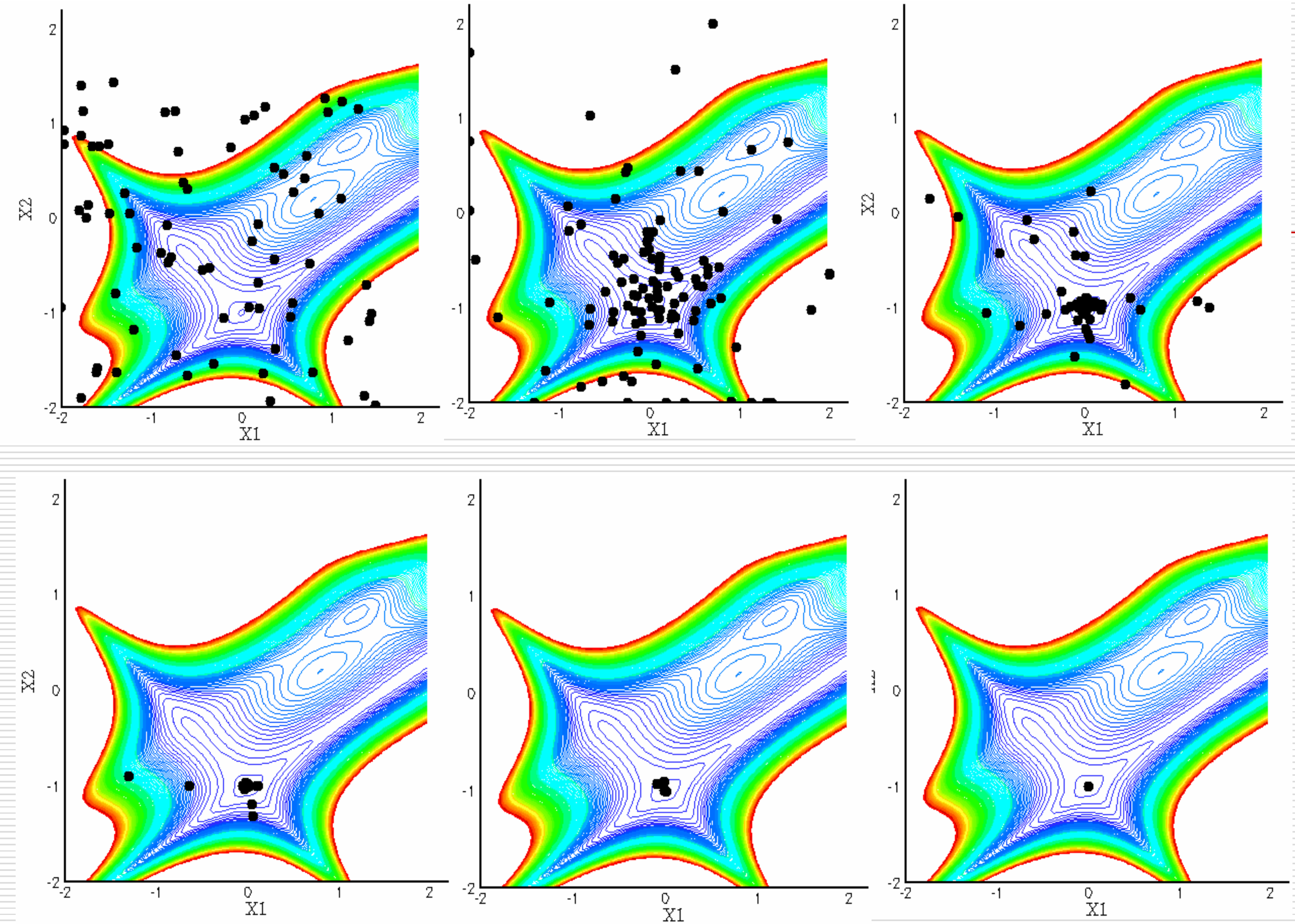
- The methodology of the original PSO is basically uses “particles” flown over the searching space and remembers the best position encountered.
- Particle swarm optimization requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed.

- The Modified PSO (Shi and Eberhart, 1998)

$$\vec{v}_i^{k+1} = w\vec{v}_i^{k} + c_1 \times rand() \times (pbest_i - \vec{x}_i^{k})$$

$$+ c_2 \times rand() \times (gbest - \vec{x}_i^{k})$$

$$\vec{x}_i^{k+1} = \vec{x}_i^{k} + \vec{v}_i^{k+1}, \quad i = 1,2, \ldots, N_{particle}$$



Minimize $\quad f(\vec{x}) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$

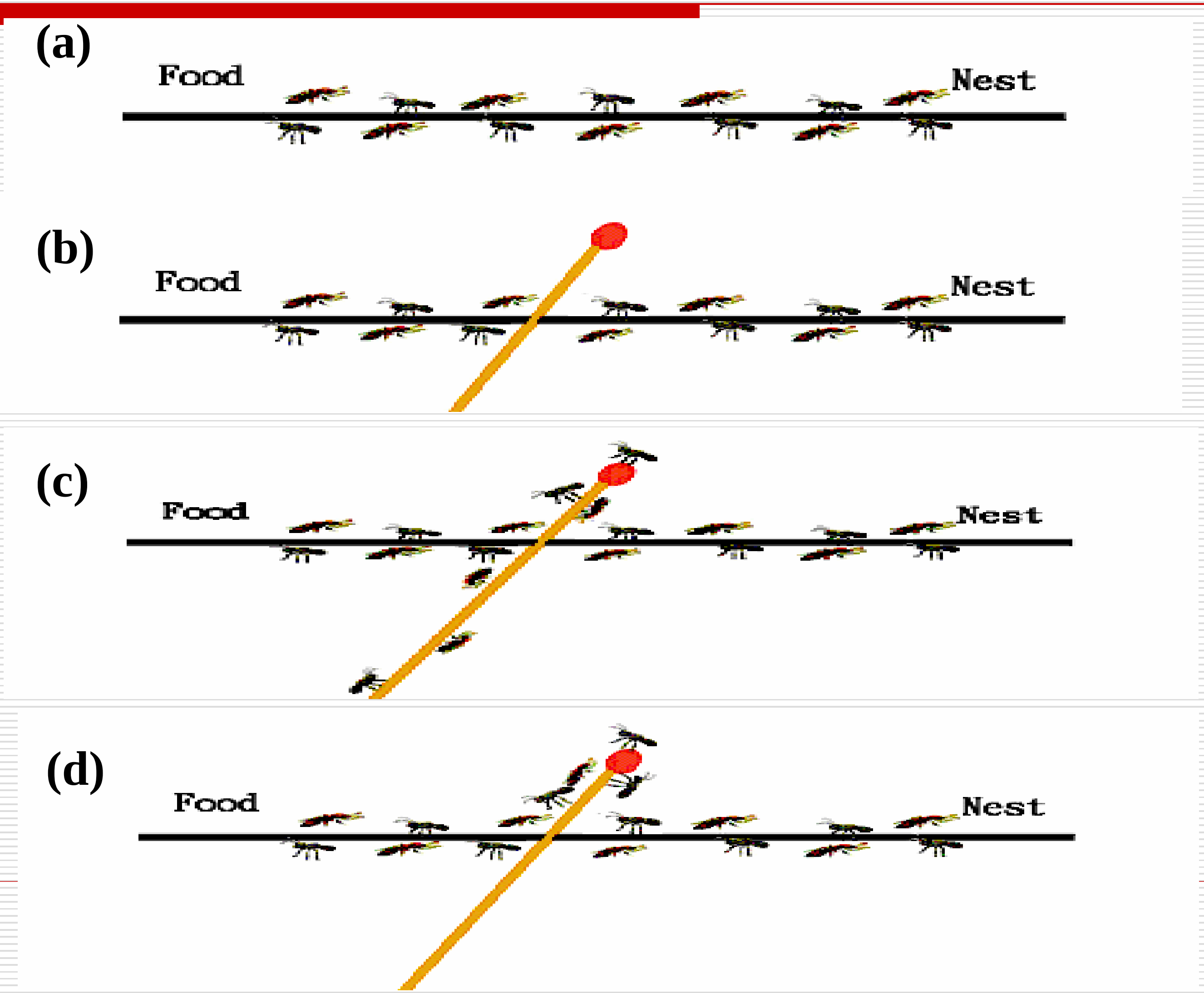
$$\times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

Subject to $\quad -2 \le x_i \le 2 \,, i = 1, 2$

Ant Colony Optimization

- ACO algorithms are inspired by the observation of real ants (Dorigo 1991).
- Real Ants are social insects organized in colonies.
- Ant colonies show a high structural level compared to the simplicity of the single individual.

- Ants coordinate their activities by an indirect form of communication (stigmergy) based on pheromone laying.
- Foraging behavior: searching for food by parallel exploration of the environment.


(a)
Food
Nest
(b)
Food
Nest
(c)
Food
Nest
(d)
Food
Nest

- Search the new shortest path when the old one is no longer feasible due to the appearance of a obstacle.

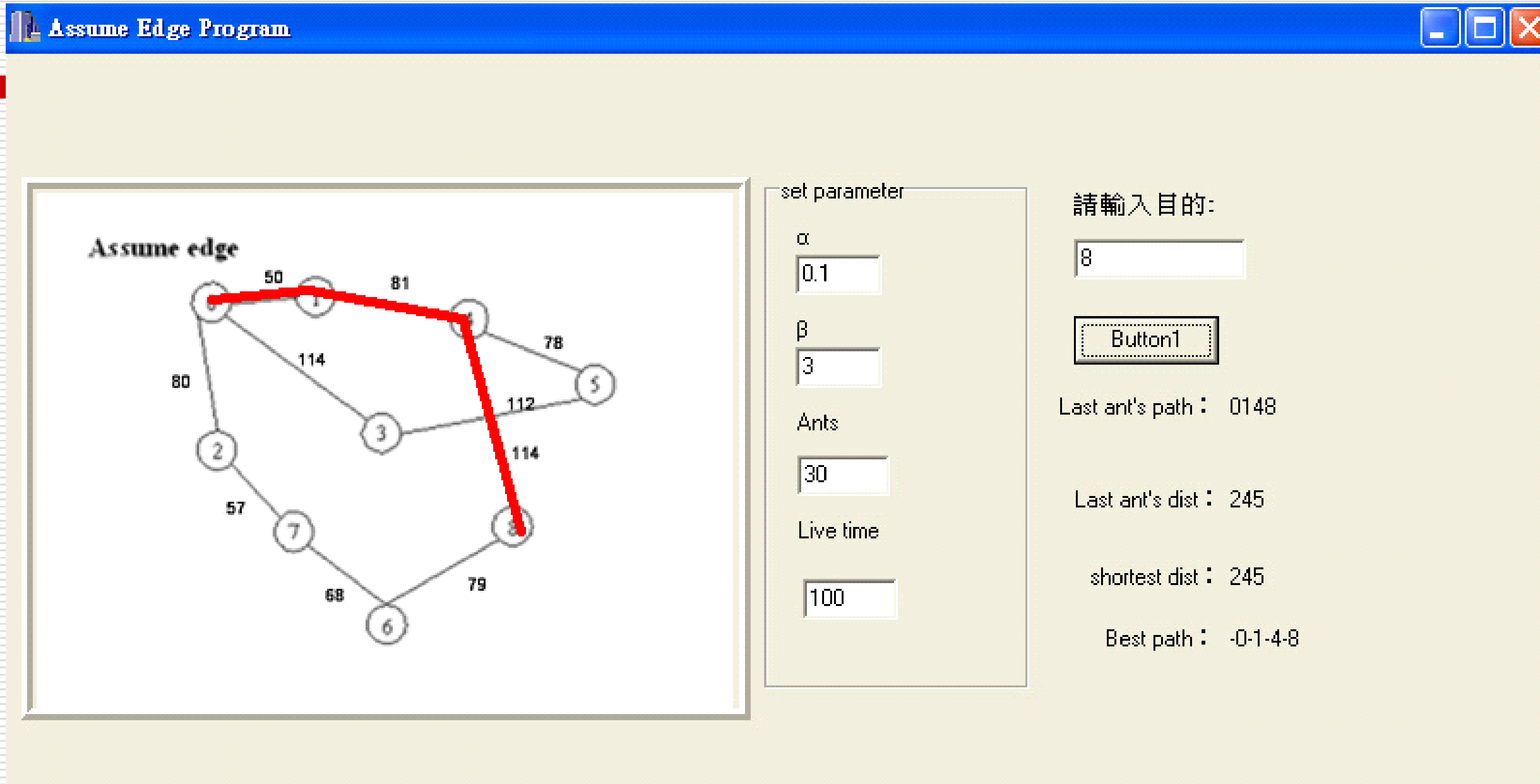
- Exploitation: the best edge is chosen

$$z = \begin{cases} \arg\max_{u \in T_k(r)} \left\{ [\tau(r,u)] \times [\eta(r,u)^{\beta}] \right\} & \text{if } (q \le q_0) \\ Z & \text{otherwise} \end{cases}$$

- Exploration: one of the edge in proportion to its value

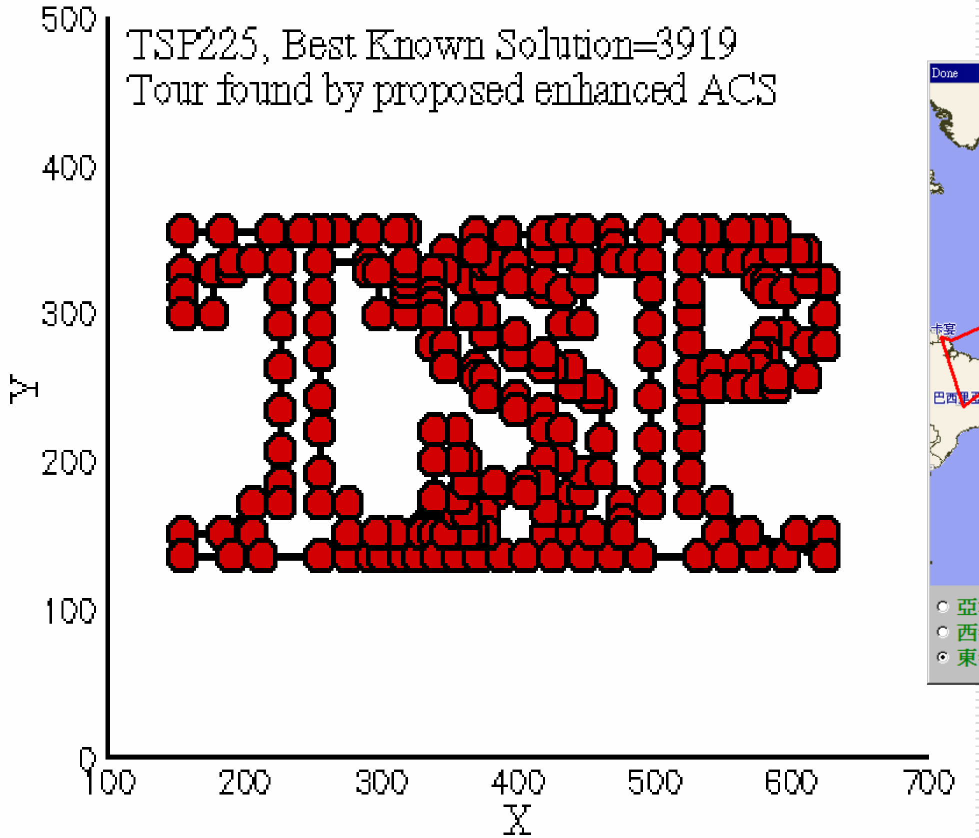
$$p_k(r,z) = \begin{cases} \dfrac{[\tau(r,z)][\eta(r,z)]^{\beta}}{\sum_{u \in T_k(r)} [\tau(r,u)][\eta(r,u)]^{\beta}} & \text{if } z \in T_k(r) \\ 0 & \text{otherwise} \end{cases}$$

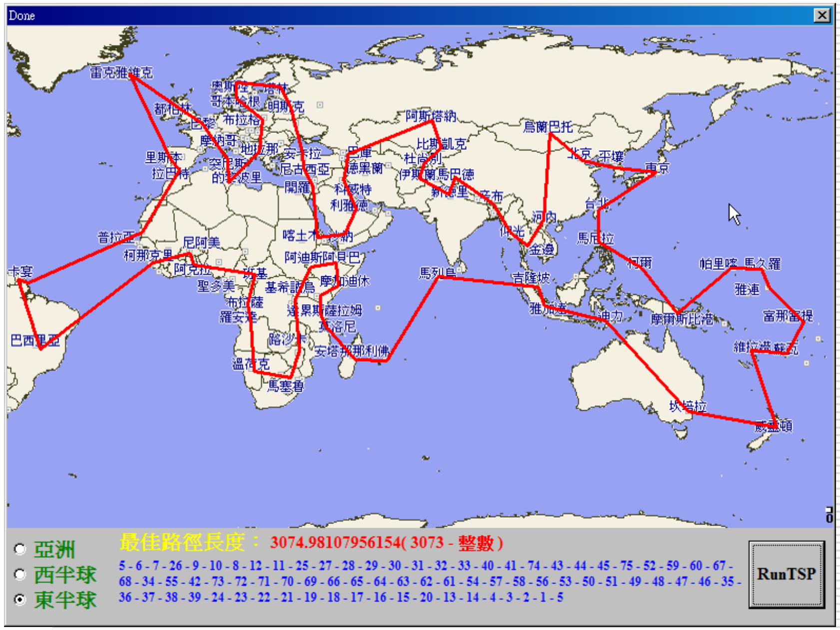
Network Routing



Start Number：0
Goal Number：8
Solution：0->1->4->8
Shortest Distance：245

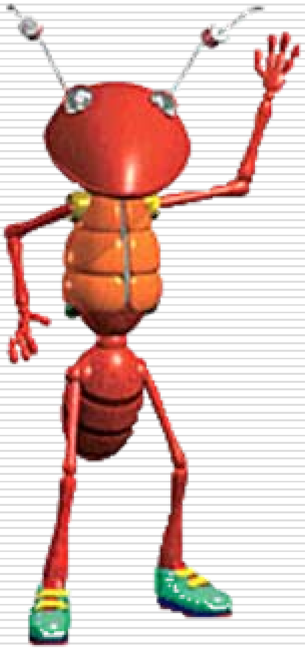
Traveling Salesman Problem





Genetically Modified Ant System

Ideas from Genetic Algorithms →

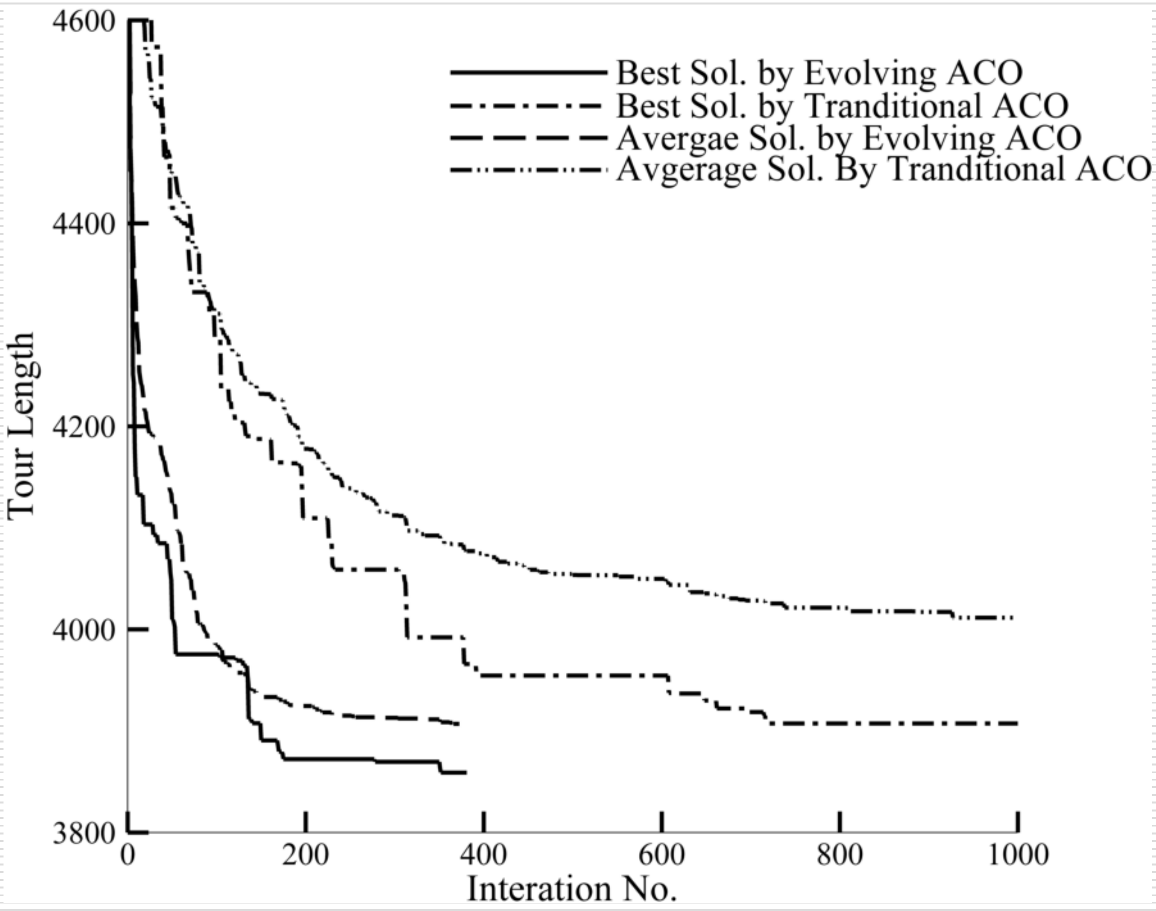


Evolving ACO

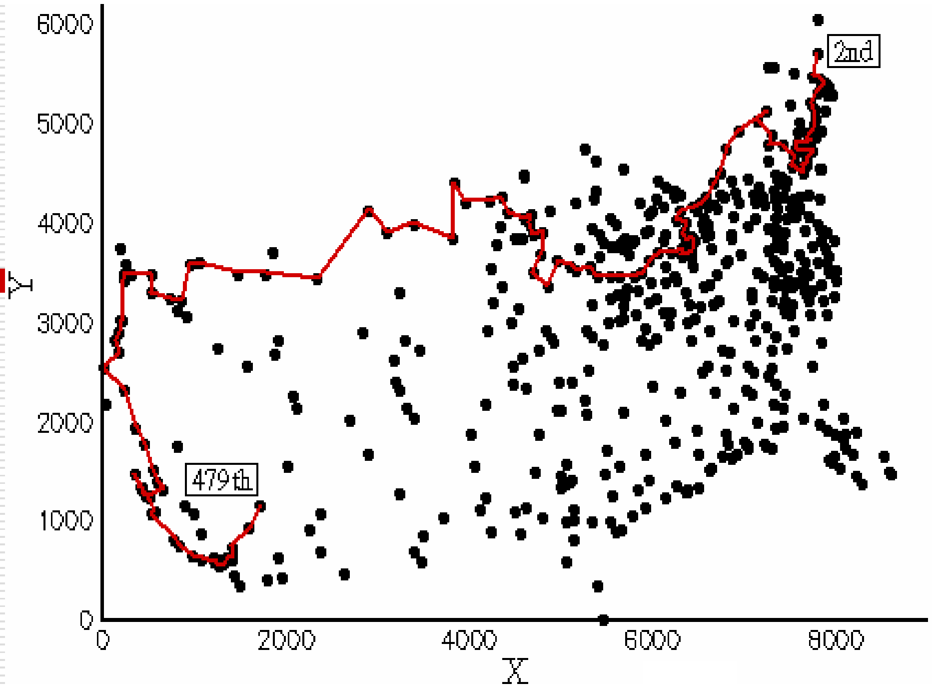
Parameter 1	$1 \leqq m \leqq 2 * n$
Parameter 2	$0.0 \leqq q_0 \leqq 1.0$
Parameter 3	$0.0 \leqq \alpha \leqq 5.0$
Parameter 4	$0.0 \leqq \beta \leqq 10.0$
Parameter 5	$0.0 \leqq \rho_{\text{local}} \leqq 1.0$
Parameter 6	$0.0 \leqq \rho_{\text{global}} \leqq 1.0$
Parameter 7	$0.0 \leqq Q \leqq 100.0$
Parameter 8	$0 \leqq l \leqq 9$
Parameter 9	$1 \leqq h \leqq 500$
Parameter 10	$1 \leqq i \leqq 10$

Enhancement in Performance

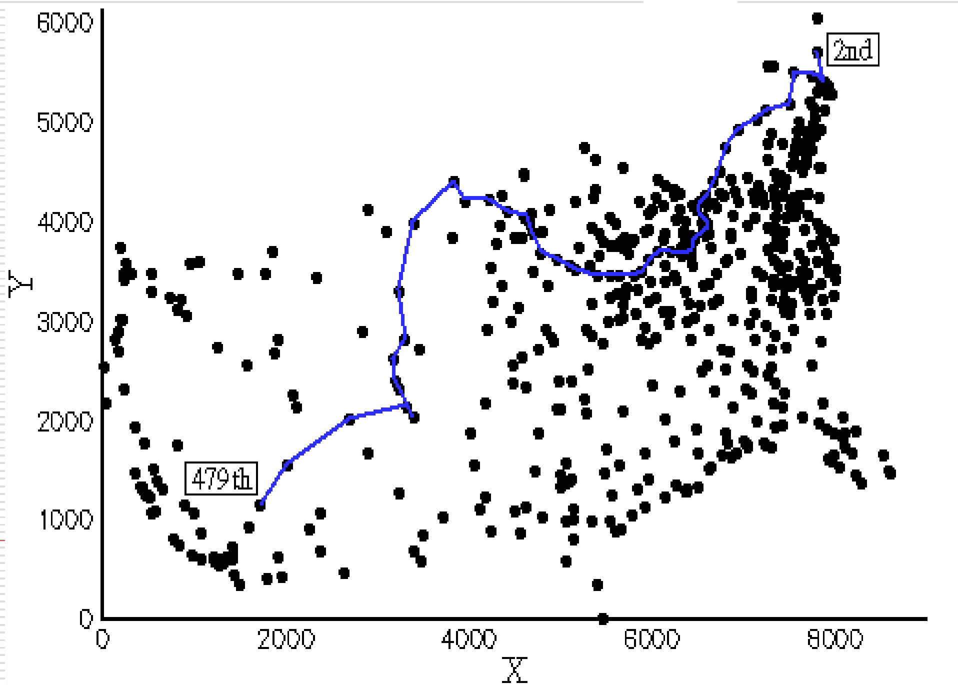
例題	演算法	參考解	最佳解	標準解誤差值 (%)	收斂疊代
Oliver30	ACO	421	421	0 %	350
	EVOACO		421	0%	311
St70	ACO	675	677	3.0%	825
	EVOACO		675	0%	378
KroA100	ACO	21282	21394	5.3%	475
	EVOACO		21282	0%	386
KroA200	ACO	29368	29860	1.7%	830
	EVOACO		29368	0%	412
TSP225	ACO	3916	3959	2.2%	710
	EVOACO		3913	0%	381



Best Sol. by Evolving ACO
Best Sol. by Tranditional ACO
Avergae Sol. by Evolving ACO
Avgerage Sol. By Tranditional ACO
Tour Length
Interation No.
4600
4400
4200
4000
3800
0
200
400
600
800
1000

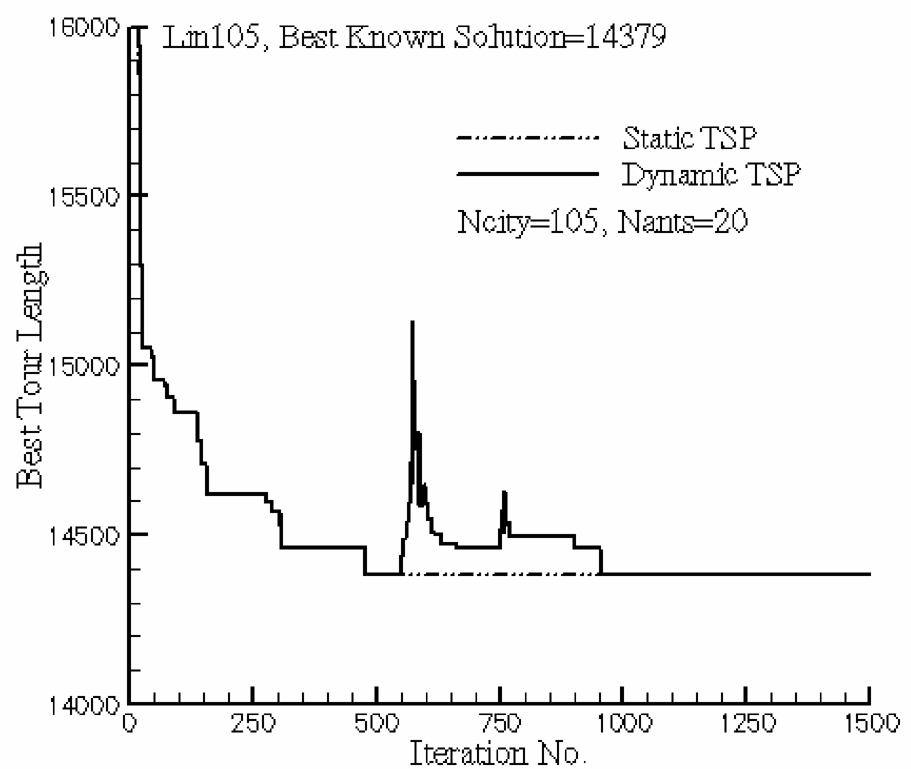


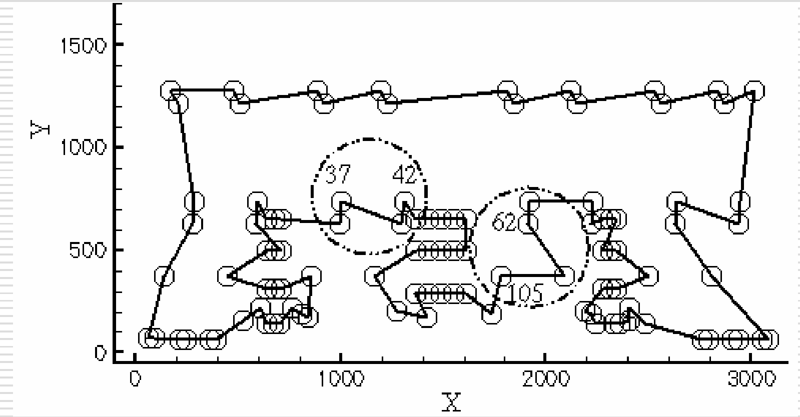
Original ACS

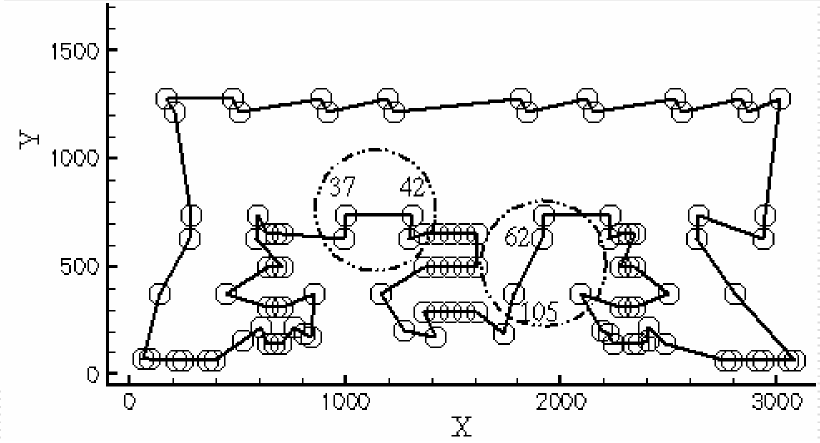


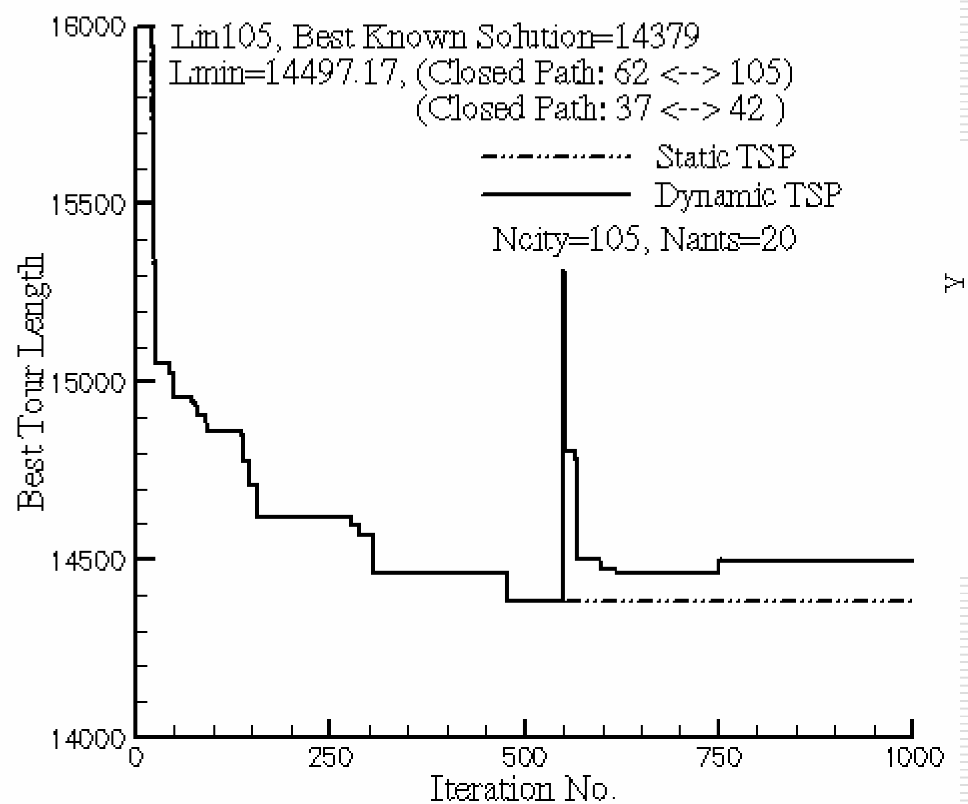
Enhanced ACS

Dynamic TSP Problems

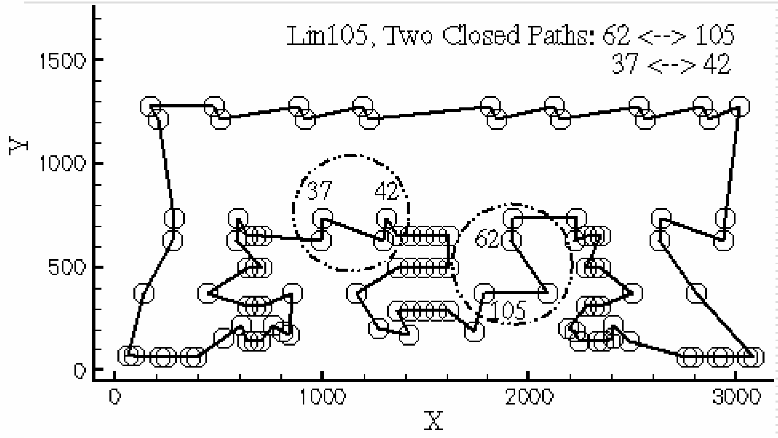






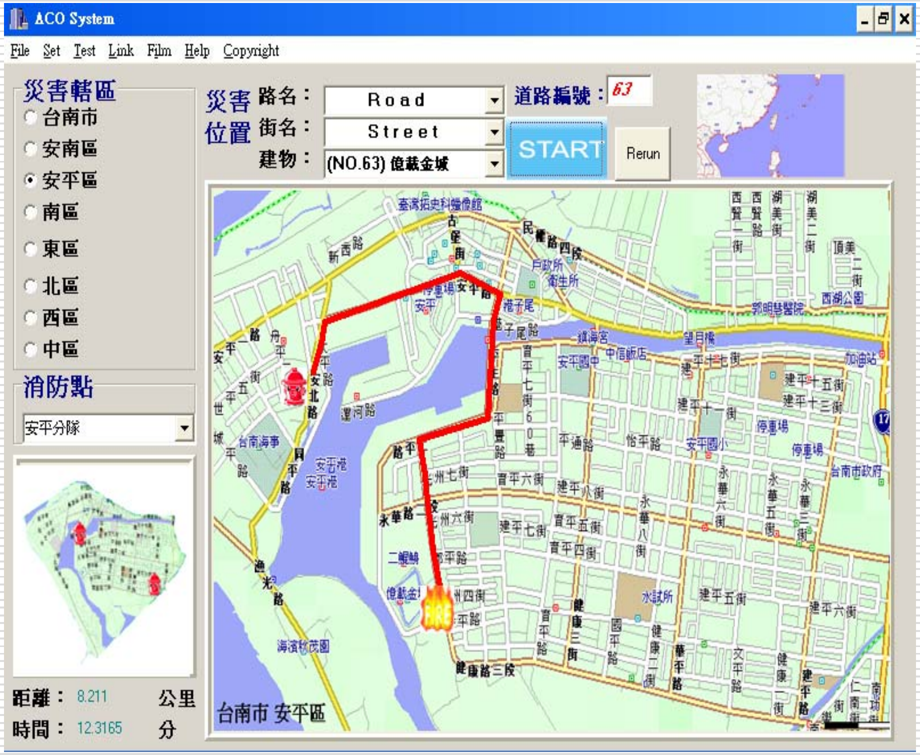


Lin105, Best Known Solution=14379
Lmin=14497.17, (Closed Path: 62 <--> 105)
(Closed Path: 37 <--> 42)
Static TSP
Dynamic TSP
Ncity=105, Nants=20
Best Tour Length
Iteration No.
16000
15500
15000
14500
14000
0
250
500
750
1000

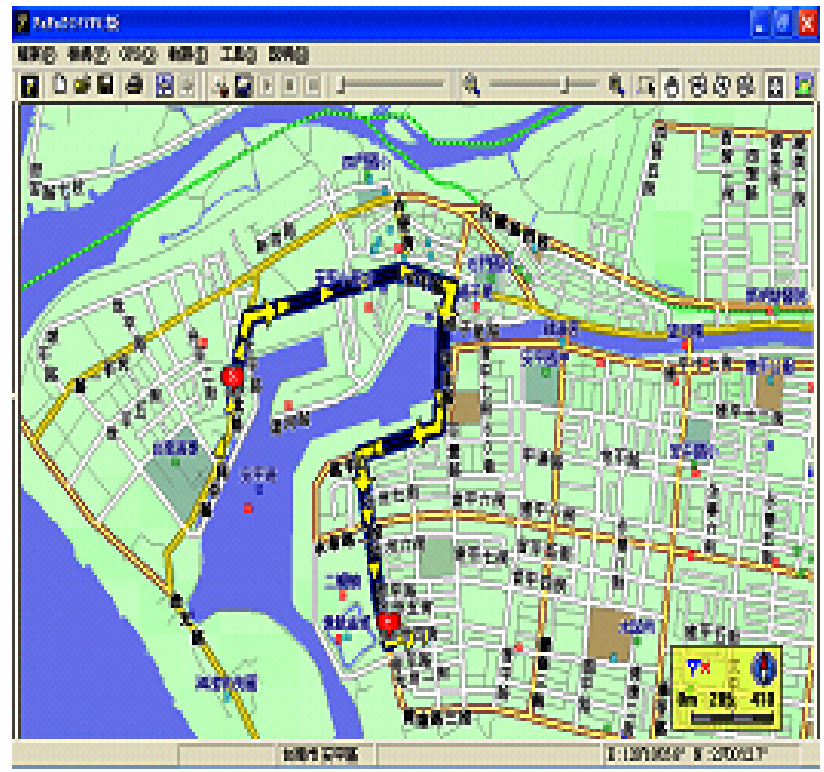


Lin105, Two Closed Paths: 62 <--> 105
37 <--> 42
37
42
62
105
Y
X
1500
1000
500
0
0
1000
2000
3000

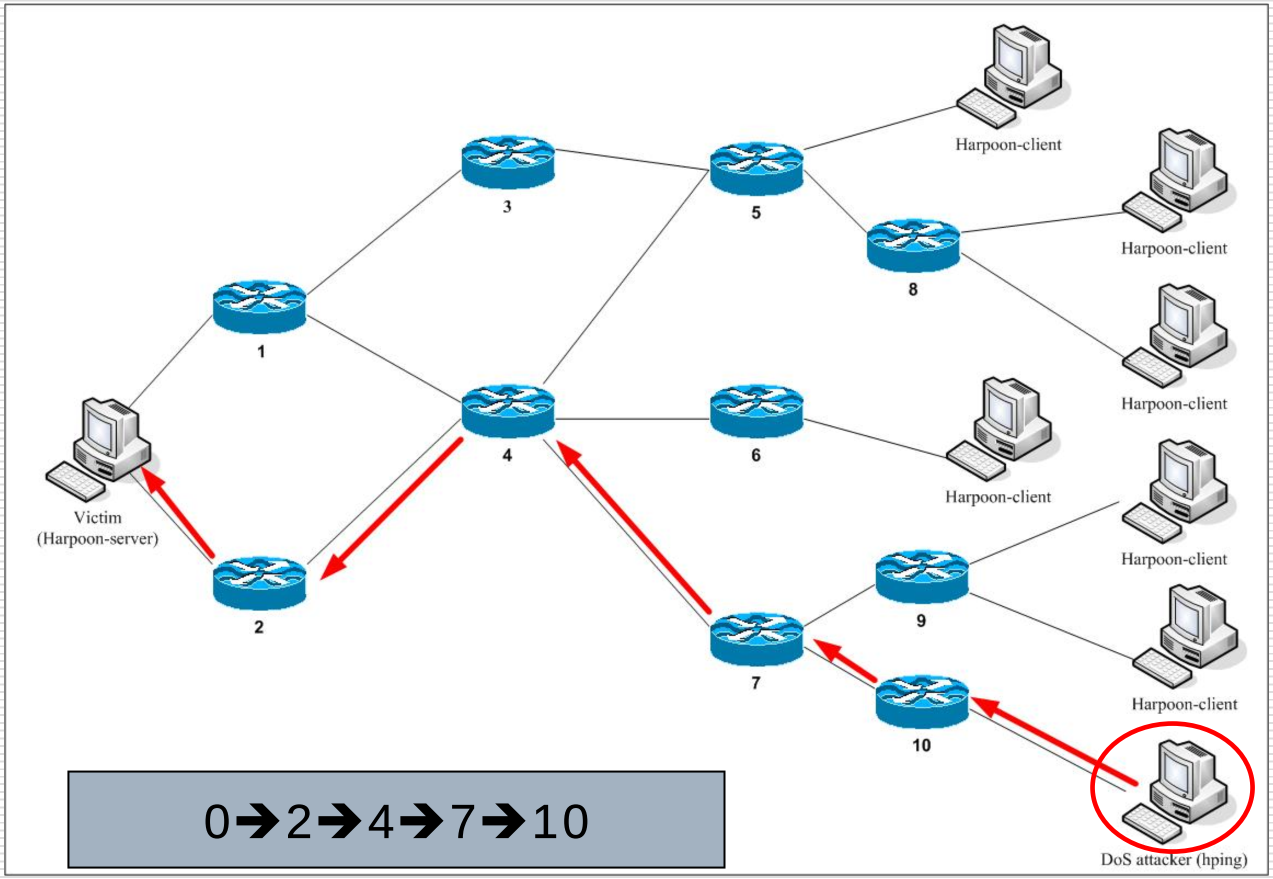
Route Programming Problems

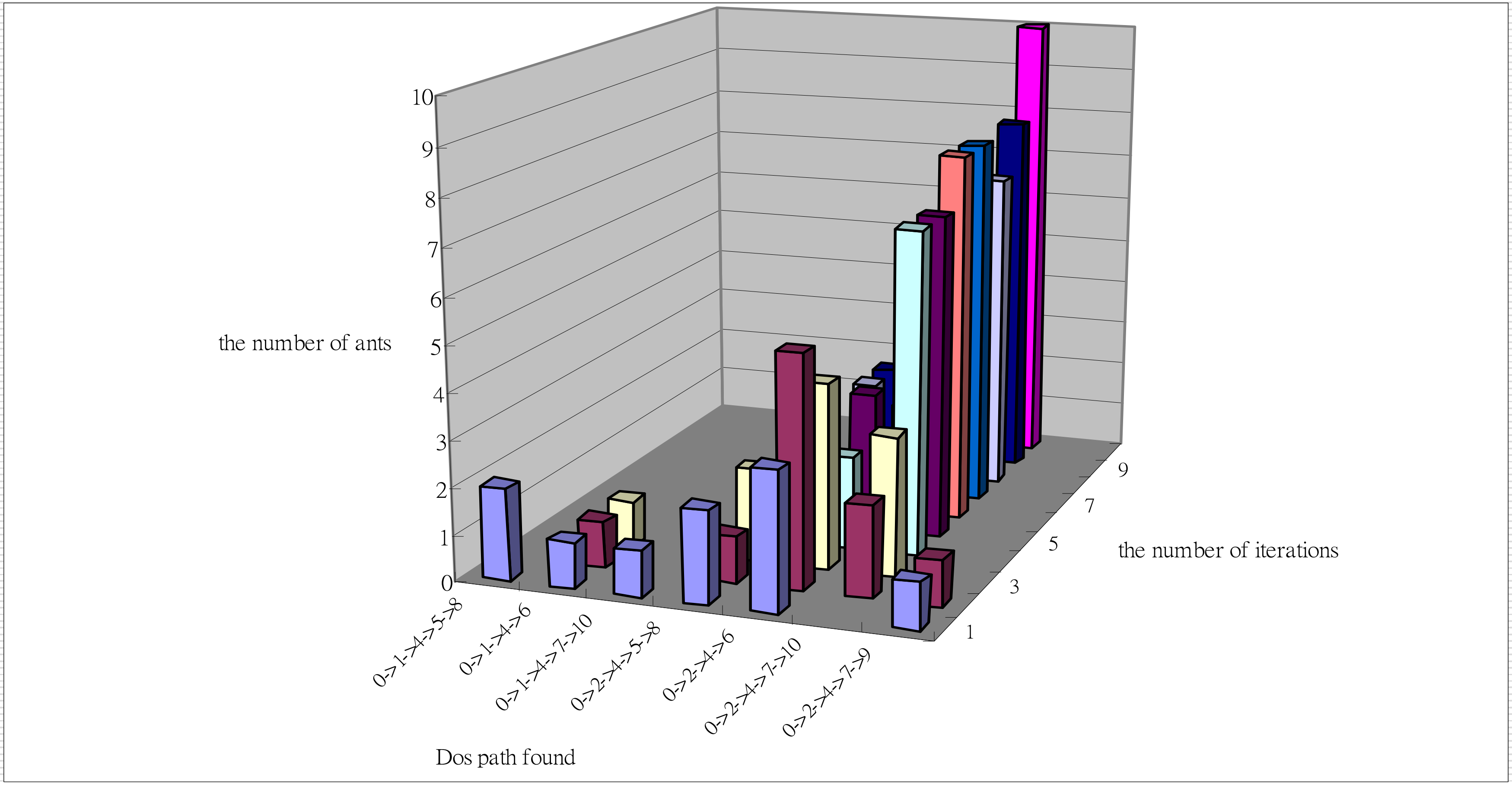



Done
TSP Link Film Copyright
災害轄區
台南市
安南區
安平區
南區
東區
北區
西區
中區
消防點
安平分隊
Small Map
Map Index
災害位置
路名：Road
街名：Street
建物：(NO.63) 億載金城
道路編號 63
迭代數目 200
Start
Return
支援分隊
台南市 安平區
距離: 3.3452 公里
時間: 5.0179 分



Denial of Service (DoS)Traceback




the number of ants
10
9
8
7
6
5
4
3
2
1
0
0->1->4->5->8
0->1->4->6
0->1->4->7->10
0->2->4->5->8
0->2->4->6
0->2->4->7->10
0->2->4->7->9
Dos path found
1
3
5
7
9
the number of iterations

Conclusion

- Performance enhancement of optimization can be achieved by reinforcing the global and local searches.
- Classical optimization methods depend on the derivatives of objective function and constraints, and always result in getting bad solutions with local optima.
- Evolutionary computation methodology makes search process more autonomous, efficient, natural and intelligent; swarm intelligence makes higher-level collaborative paradigm.
- An efficient optimizer is the central module for the engineering optimization design.

「鐘聲二十一響」— 往更高一層次去思考!

Thank You for Your Attention