CS-4752
Introduction to Computational Intelligence

Lecture 19

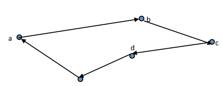
October 25, 2011

Discrete PSO for TSP

- K.P. Wang, L. Huang, C.G. Zhou, W. Pang, Particle swarm optimization for traveling salesman problem, International Conference on Machine Learning and Cybernetics 3 (2003) 1583–1585.

Traveling Salesman Problem

- The salesman must visit every city in his territory exactly once and then return home covering the shortest distance.
- Variables: x_1.. x_n are n city names
- Representation: permutation of n cities.
- 5 cities example: (a, d, g, b, e)



TSP Search Space

- Given *n* unique objects, *n!* permutations of the objects exit. Searching the shortest path is an NP-hard problem.
- In TSP, there are multiple equivalent solutions.
  - If starting point is not important, and the distance from city i to j is the same as that from city j to i, each tour a-b-c-d-e can be represented in 2n ways and give the same distance.
- Search space: n!/(2n)=(n-1)!/2

PSO for TSP

- The solution of a particle is a permutation of all cities.
  - Example: (a, d, g, b, e)
- The velocity of a particle is a sequence of swap operators.
- Velocity examples:
  - swap operator (SO)= SO(1,2) //swap first visited city with the second visited city
  - swap sequence (SS)=(SO(1,2,), SO(5,4), SO(5,1))

Particle Solution Update

x(t) = x(t-1) + v(t)

- Applying SO to a permutation:
  - (a, d, g, b, e) + SO(1,2) = (d, a, g, b, e)
- Apply a sequence of SO (SS) to a permutation:
  - SS=(SO(1,2,), SO(5,4), SO(5,1))
  - (a, d, g, b, e) +SS -> (d,a,g,b,e) -> (d,a,g,e,b) -> (b,a,g,e,d)

Particle Velocity Update

- $v(t)=v(t-1) \oplus \alpha*(pbest-x(t-1)) \oplus \beta(gbest-x(t-1))$
- Merging two Swap Sequences
  - SS1 = (SO(1,2,), SO(5,4), SO(5,1))
  - SS2 = (SO(1,3,), SO(5,1), SO(2,1))
  - SS1$\oplus$ SS2 = (SO(1,2,), SO(5,4), SO(5,1), SO(1,3,), SO(5,1), SO(2,1))

Subtract Two Permutations

- A: (a, c, d, e, b), B: (c, a, b, e, d)
- There is a SS that transforms A to B.
  - a is in position 1 in A and 2 in B: $SO_1(1,2)$
  - $A+SO_1=A'=$ (c,a,d,e,b)
  - b is in position 5 in A' and 3 in B: $SO_2(5,3)$
  - $A'+SO_2=$(c,a,b,e,d)
- SS=($SO_1(1,2)$, $SO_2(5,3)$)
- A-B = SS

Particle Velocity Update

- Each velocity is a swap sequence.
- $v(t) = v(t-1) \oplus \alpha*(pbest-x(t-1)) \oplus \beta*(gbest-x(t-1))$
- α, β are random number between 0 and 1.
- The probability that all swap operators in swap sequence (pbest – x(t-1)) are included in the updated velocity is α.
- The probability that all swap operators in swap sequence (gbest – x(t-1)) are included in the updated velocity is β.
- There is improvement on this in another paper.

TSP-PSO algorithm

- Random initialization of permutation and swap sequences.
- For each time step

  Update gbest if needed,

  Update pbest if needed.

  For each particle in the swarm

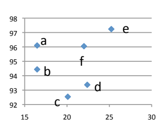
  $v(t) = v(t-1) \oplus \alpha*(pbest-x(t-1)) \oplus \beta*(gbest-x(t-1))$;

  $x(t) = x(t-1) + v(t)$;

  End
- End

Exercise

	x	y
a	16.47	96.1
b	16.47	94.44
c	20.09	92.54
d	22.39	93.37
e	25.23	97.24
f	22	96.05



	a	b	c	d	e	f
a						
b	1.66					
c	5.077204	4.088325				
d	6.519149	6.015921	2.445179			
e	8.833867	9.196608	6.964883	4.80026		
f	5.530226	5.759601	3.996023	2.708228	3.442238	

Euclidean Distance Table

Exercise

t=0, gbest=E

particle	permutation	velocity	fitness
A	c a e f b d	(2,1),(3,2)	31.57401
B	b d e f a c	(2,5),(6,1)	28.95417
C	f d a b e c	(4,3),(1,5)	31.04489
D	d b f e a c	(3,6),(5,6)	31.57401
E	c d e b a f	(2,4),(1,3)	27.6283

$v(t) = v(t-1) \oplus \alpha*(pbest-x(t-1)) \oplus \beta*(gbest-x(t-1))$;

$x(t) = x(t-1) + v(t)$;

Particle A: Beta=1

E-A: SO1=(2,5), E'=E+(2,5)=(c,a,e,b,d,f), SO2=(6,4), E''=E'+SO2=(c,a,e,f,d,b)

SO3=(5,6), E'''=E''+SO3=(c,a,e,f,b,d)

v(t) =((2,1),(3,2))+((2,5),(6,4),(5,6))=((2,1),(3,2),(2,5),(6,4),(5,6))

A(t=1)=(c,a,e,f,b,d)+((2,1),(3,2),(2,5),(6,4),(5,6))=(a,c,e,f,b,d)+((3,2),(2,5),(6,4),(5,6))=

(a,e,c,f,b,d)+((2,5),(6,4),(5,6))=(a,b,c,f,e,d)+ ((6,4),(5,6))=(a,b,c,d,e,f)+(5,6)=(a,b,c,d,f,e)

Fitness=23.17784