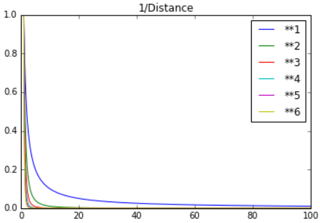
Ant-Colony Optimization

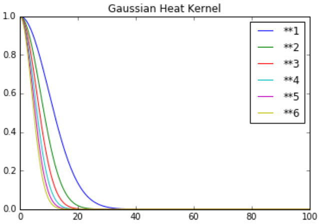
TSP-ACO:

Initialize pheromone values
repeat
  for ant $k \in \{1, \ldots, m\}$\{solution construction
    $S := \{1, \ldots, n\}$ \{set of selectable cities\}
    choose city i with probability p_{0i}
    repeat
      choose city $j \in S$ with probability p_{ij}
      $S := S - \{j\}$
      $i := j$
    until $S = \emptyset$
  endfor
  forall i, j **do**
    $\tau_{ij} := (1 - \rho) \cdot \tau_{ij}$ \{evaporation\}
  endfor
  forall i, j in iteration best solution **do**
    $\tau_{i,j} := \tau_{ij} + \Delta$ \{intensification\}
  endfor
until stopping criterion is met

Function Choices

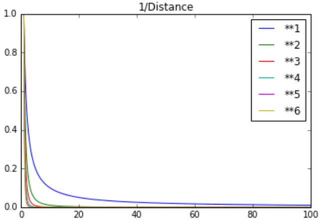
- Heuristics

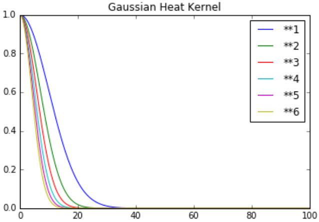




Function Choices

- Heuristics





- Intensification

Best m' ants

```
soluMat[:p_ants][soluMat[:p_ants] > 0] = intensification
# substitute costs with intensification rate for p_ants best solutions
for a in soluMat[:p_ants]:
    pheroMat += a  # add intensification to pheromone matrix
```

Softmax over all ants

```
soluMat = ((soluMat>0)+0) * intensification
# invert distances, so lowest has biggest value
N_inv = 1./(np.array(N))
# normalize
N_inv /= np.max(N_inv)
# apply softmax to hundredth power (power bc we want to spread out vals)
sm_n = softmax(N_inv**100)
for ant_x,s in enumerate(sm_n):
    k = soluMat[ant_x,:,:] * s * intensification
    pheroMat += k
```

Hyperparameters

- **M** Number of ants per iteration
- **M'** Number of ants that increase pheromones
- τ Initial Pheromone Value
- α Influence Of Pheromone
- β Influence of Heuristic
- **evap_rate** Amount of pheromone removed in second phase
- **inten_rate** Amount of pheromone added in third phase
- **q** Probability to follow the strongest trail

Hyperparameter search

- Optimal Results
  - Values for α and β in a previous search
  - Different combinations for M, M', τ, evap_rate, inten_rate, q in a systematic search as well as random search
- Fixed Parameters α and β
  - Different combinations for M, M', τ, evap_rate, inten_rate, q in a random search

```python
trials = 5
iterations = 400
alphas_must = [1, 1]
betas_must = [1, 0]
alphas_best = [1, 2, 3]
betas_best = [4, 5, 8]
# num_ants, p_ants, tau, evap_rate, intensification, q
param_sets_large = [
    [100, 10 , 10, 0.2, 1, 0.1],
    [100, 10 , 10, 0.2, 2, 0.1],
    [100, 1 , 10, 0.2, 1, 0.1],
    [100, 1 , 10, 0.2, 2, 0.1],
    [50, 5 , 10, 0.2, 1, 0.1],
    [50, 1 , 10, 0.2, 2, 0.1]
]
```

```python
param_sets_small = []
for i in range(6):
    n_set = [10, 1]
    n_set.append(np.random.randint(low=2, high=10))
    n_set.append(np.random.uniform(low=0.1, high=0.4))
    n_set.append(np.random.uniform(low=0.8, high=2.5))
    n_set.append(np.random.uniform(low=0.05, high=0.3))
    param_sets_small.append(n_set)
```

Results Hyperparameter search

alpha	beta	ants	p_ants	tau	evap_rate	inten	q	best_res	mean_res
PROBLEM 1 (Nicos optimum: 3632)									
1	4	100	10	10	0.2	2	0.1	3642.0	3696.0
1	4	100	10	10	0.2	1	0.1	3670.0	3730.4
1	4	50	5	10	0.2	1	0.1	3724.0	3766.8
2	5	100	10	10	0.2	2	0.1	3740.0	3777.2
PROBLEM 2 (Nicos Optimum: 2878)									
1	4	100	10	10	0.2	2	0.1	2918.0	2943.4
1	4	100	10	10	0.2	1	0.1	2904.0	2947.2
1	4	50	5	10	0.2	1	0.1	2902.0	2960.2
3	8	100	1	10	0.2	1	0.1	2948.0	2966.6
PROBLEM 3 (Nicos Optimum: 2617)									
1	4	100	10	10	0.2	1	0.1	2662.0	2685.6
1	4	100	10	10	0.2	2	0.1	2668.0	2701.4
1	4	50	5	10	0.2	1	0.1	2663.0	2712.0
2	5	100	10	10	0.2	1	0.1	2668.0	2731.6

Best results

Params	**Problem 1**	**Problem 2**	**Problem 3**
α=1, $\beta = 0$	12372	9721	8662
α=1, $\beta = 1$	5980	4462	4490
$\boldsymbol{\alpha}$**=1,** $\boldsymbol{\beta}$**=4**	3642	2918	2662
Nico's Results	**3632**	**2878**	**2617**

Observations

- Extreme punishing of high distances ($\rightarrow$ high β-values) lead to faster finding of better results (compare hillclimbing)
- Finding optimal values for pheromone-values didn't lead to much improvement
- More ants (higher M and M') increased performance
- Results almost always converge in the first few iterations

Importance of heuristics