15-382 Collective Intelligence - S19

Lecture 26: Swarm Intelligence 7 / Ant Colony Optimization 3

Teacher:
Gianni A. Di Caro

جامعة كارنيجي ميلون في قطر
Carnegie Mellon University Qatar

ANT-ROUTING TABLE: COMBINING PHEROMONE AND HEURISTIC

- The values of τ_i and η_i at each node i must be combined in order to assign a unique quality value to each locally available next hop $j \in \mathcal{N}(i)$ (to bias the construction process):

$$\mathcal{A}_i(j) = f_\tau(\vec{\tau}_i, j) \circ f_\eta(\vec{\eta}_i, j)$$

- $\mathcal{A}_i(j)$ is called the Ant-routing table: it summarizes all the information locally available in j to an ant agent to make next hop selection

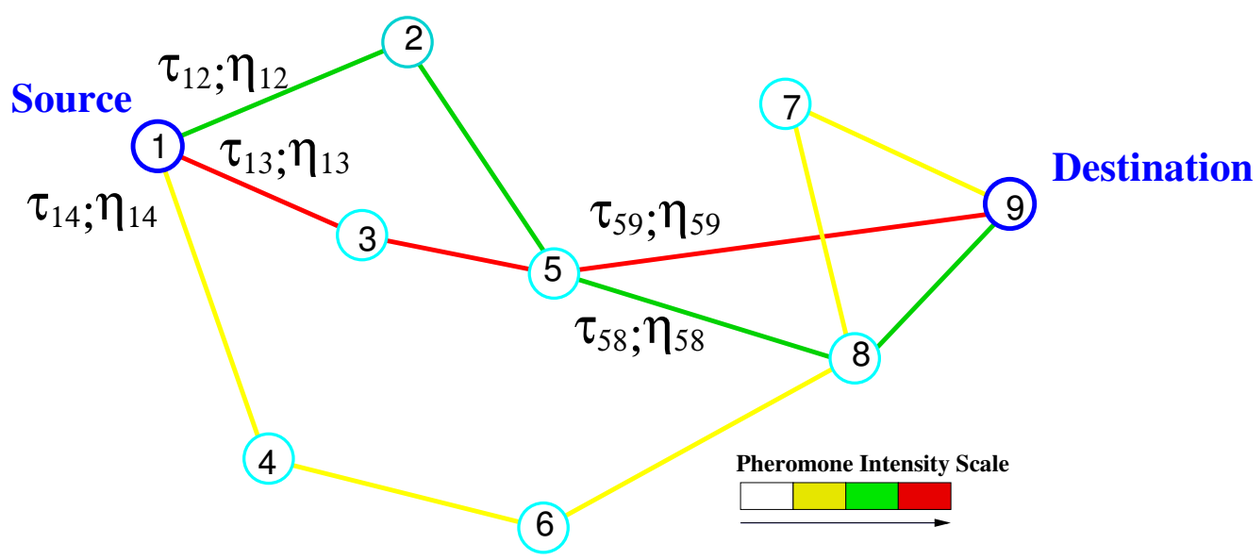
- Examples (most used combinations):
  - $\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}$ (multiplicative combination, α and β weighting parameters)
  - $\alpha\tau_{ij} + (1-\alpha)\eta_{ij}$ (additive combination, α relative weighting parameter)

- The functional form of f_τ and f_η and of their composition defines how the ant-learned pheromone information and the heuristic information are combined and weighted in order to locally take optimized decisions
  - Which is the right balance between τ and η ?

STATE-TRANSITION: MOVING TO THE NEXT HOP/ STATE

- At the t-th step of its solution construction process the ant agent k arrives at decision node i and performs the following actions:

  1. Calculates the values $\mathcal{A}_i(j)$ for all $j \in \mathcal{N}^F(i) \subseteq \mathcal{N}(i)$ that are still feasible given the status of the solution construction process, that is, given the ant partial solution ξ_t^k
  2. Derives from each $\mathcal{A}_i(j)$ a probability value $\pi_\epsilon(i,j)$, that reflects the estimated quality of adding j to the partial solution in the perspective of building a good complete solution to the problem
  3. Draws a uniform random number and selects the next hop according to the selection probabilities $\pi_\epsilon(i,j)$ (the larger $\pi_\epsilon(i,j)$, the larger the probability of j being selected)
  4. Expand the partial solution ξ_t^k according to the selection resulting from 3.



FUNCTIONAL FORM OF THE STOCHASTIC DECISION POLICY

- Common choices to define the selection probabilities for the stochastic decision policy:
  - Random-proportional: $\pi_\epsilon(i,j) = \dfrac{\mathcal{A}_i(j)}{\sum_{k\in\mathcal{N}^F(i)} \mathcal{A}_i(k)}$
  - ϵ-greedy: $\begin{cases} \text{if } p_b > p_u: \quad \pi_\epsilon(i,j) = \begin{cases} 1 & \text{if } j = \arg\max\{\mathcal{A}_i(j),\ j \in \mathcal{N}^F(i)\} \\ 0 & \text{otherwise} \end{cases} \\ \text{else}: \qquad \pi_\epsilon(i,j) = 1/|\mathcal{N}^F(i)|,\ \forall j \in \mathcal{N}^F(i) \end{cases}$

    $p_u \in [0,1)$ is a small threshold value for uniform exploration (e.g., $p_u = 0.05$: if the randomly generated value $p_b \in [0,1]$ is larger than p_u, the best (greedy) selection is issued, otherwise an unbiased uniform selection is issued
  - Soft-max: $\pi_\epsilon(i,j) = \dfrac{e^{\mathcal{A}_i(j)/T}}{\sum_{k\in\mathcal{N}^F(i)} e^{\mathcal{A}_i(k)/T}}, \quad T \approx 0$

POSSIBLE STRATEGIES FOR PHEROMONE UPDATING

- (Real) Ants update pheromone online step-by-step → Implicit path evaluation based on on traveling time and rate of updates
- Ant's way is inefficient and risky (maybe the followed path is really bad or is not even feasible)
- When possible, the "right" way is online delayed + pheromone manager filter:
  - Complete the path
  - Evaluate (assign a score) and Select (is this path worth to be up/down reinforced?)
    - *Selection* can be performed by a pheromone manager that, according to the score, decides whether the pheromone variables associated to the solution should be up/down reinforced or not (e.g., the pheromone manager can use the policy that only the best solution in the current iteration should determine an update in the pheromone variables)
  - "Retrace" and assign credit / reinforce the quality value of the decisions (pheromone variables) issued to built the path
  - Total path cost J can be safely used as reinforcement signal (not always trivial to calculate)

    TSP: $s = (1, 3, 5, 7, 9)$, $J(s) = c_{13} + c_{35} + c_{57} + c_{79} + c_{91}$, $\tau_{13} \leftarrow \tau_{13} + 1/J(s)$, $\tau_{35} \leftarrow \tau_{35} + 1/J(s)$
- Online step-by-step: locally decrease pheromone for exploration (e.g., ACS)
- Offline: *daemon*, evaporation: $\tau_{ij} \leftarrow \rho\tau_{ij}$, $\rho \in [0, 1]$,

ANT AGENT: THE BEHAVIOR

```
Function ant_construct_solution()
initialize_ant_parameters();
t ← 0;  ξ_0 ← ∅;  M ← ∅;
ξ_t ← get_starting_partial_solution();
M ← update_ant_memory(ξ_t, 0);
while (ξ_t ∉ {set of feasible solutions})
   A^t ← read_local_ant-routing_table(ξ_t);
   P^t ← compute_transition_probabilities(A^t, M, PROBLEM_CONSTRAINTS);
   ξ_{t+1} ← apply_ant_decision_policy(π_ε, P^t, PROBLEM_CONSTRAINTS);
   move_to_new_state(ξ_{t+1});
   if (online_step_by_step_pheromone_update)
      update_pheromone_variable_used_to_select_the_state_transition();
      update_ant-routing_table();
   end if
   M ← update_ant_memory(ξ_{t+1}, get_transition_cost(ξ_t, ξ_{t+1}));
   t ← t + 1;
end while
J ← evaluate_constructed_solution(M, ξ_t);
return ξ_t, J;        RETURN THE COMPLETE SOLUTION AND ITS VALUE

Function ant_online_delayed_pheromone_update()
update_pheromone_variables_used_in_solution(J, M);
update_ant-routing_tables(J, M);
```

ANT SYSTEM (1994)

Function AntSystem()

Pheromone model: $\tau_{ij} \equiv$ (estimated) quality of selecting city j when i is the current city

Heuristic variables: $\eta_{ij} \equiv 1$ / distance from city i to city j

$m \leftarrow$ number of ants per iteration (i.e., samples for *policy evaluation*)

$nn_{tour} \leftarrow$ find initial solution with nearest neighbor heuristic

Init pheromone: $\tau_{ij} = \tau_0 = 1/(n \cdot nn_{tour}),\ \forall i, j \in \{1, \ldots, n\}$

for $t := 1, \ldots iterations_num$

  for $k := 1, \ldots, m$

    ant_construct_solution$(k, \neg online_step_by_step_pheromone_update)$

  for $k := 1, \ldots, m$ // ALL ANTS UPDATE PHEROMONE VARIABLES

    ant_online_delayed_pheromone_update(k)

  foreach edge$(i, j),\ i, j \in \{1, \ldots, n\}$ PHEROMONE EVAPORATION ON ALL EDGES

    $\tau_{ij}(t) \leftarrow \rho \tau_{ij}(t-1), \quad \rho \in [0, 1]$

return best_solution_generated

ANT SYSTEM (1994)

- Ant-routing table and probabilistic selection policy:
  - ξ_r^k is the partial solution for ant k at the r-th construction step, i is the last city added to ξ_r^k, and $\mathcal{N}_i^k$ indicates the feasible neighbor of ξ_r^k, that is the set of cities that can be feasibly added in i to the solution being constructed by ant k (e.g,, $n = 5,\ \xi_3^k = (1,3,4), \mathcal{N}(\xi_3^k) = \mathcal{N}_4^k = \{2,5\}$)

The Ant-routing table at city i at time t : $\mathcal{A}_{ij}(t) = \tau_{ij}(t)\eta_{ij}^{\beta}$

City selection probability in i, for ant k: $$\pi_{ij}^k(t) = \begin{cases} \dfrac{\mathcal{A}_{ij}(t)}{\sum_{l \in \mathcal{N}_i^k} \mathcal{A}_{il}(t)} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{otherwise} \end{cases}$$

- Pheromone updating (+ evaporation):

$$\tau_{ij}(t) \leftarrow \rho\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t), \quad m = \text{ants per iteration}$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} 1/L^k(t) & \text{if } (i,j) \in T^k(t) \\ 0 & \text{if } (i,j) \notin T^k(t) \end{cases}$$

T^k = tour built by ant k at iteration t, L^k is its length

AS: EXPLORATION - EXPLOITATION TRADEOFF

- A balance between pheromone intensity, τ_{ij}, and heuristic information, η_{ij}
- If $\alpha = 0$:
  - No pheromone information is used, i.e. previous search experience is neglected
  - The search then degrades to a stochastic greedy search
- If $\beta = 0$:
  - The attractiveness of moves is neglected
  - The search algorithm is similar to SACO
- Heuristic information adds an explicit bias towards the most attractive solutions, e.g.

$$\eta_{ij} = \frac{1}{d_{ij}}$$

AS: PHEROMONE EVAPORATION

- To improve exploration abilities, and to prevent premature convergence:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$$

  with $\rho \in [0, 1]$
- ρ specifies the rate at which pheromones evaporate, causing ants to "forget" previous decisions
- ρ controls the influence of search history
- For large values of ρ, pheromone evaporates rapidly, while small values of ρ result in slower evaporation rates
- Large values therefore implies more exploration, more random search

AS: PHEROMONE UPDATE

Pheromone is iteratively deposited in an *additive cumulative* modality based on solution quality

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t)$$

where

$$\Delta\tau_{ij}^{k}(t) = \frac{1}{L^k(t)}$$

$L^k(t)$ is the length of the path constructed by ant k at time step t
n_k is the number of ants

QUESTIONS

1. Why an additive, cumulative rule for pheromone updating and not an average, for instance?
   (not looking for averages, but for the "sparse" best solutions)

2. Is there any potential problem with pheromone bounds?
   (get to zero, unlimited growth)

3. Is there any potential problem of premature convergence?

4. Is it a good idea to have a large number of samples / ants given the adopted rule for pheromone updating?
   (all solutions do pheromone updating → A lot of "bad" ones!)

5. How do we balance **policy evaluation** and **policy improvement**?

AS: OTHER PHEROMONE UPDATE RULES

Idea: assign credits relative to some Q costant value related to problem's costs

Q = an upper bound estimate on the length of the optimal tour, in *Ant-cycle*

Q = small value related to the range of cost values, *Ant-density* & *Ant-Quantity*

- Three variations in the way pheromone deposits are calculated
- Ant-cycle AS:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{f(x^k(t))} & \text{if link } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \text{otherwise} \end{cases}$$

- Ant-density AS:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q & \text{if link } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \text{otherwise} \end{cases}$$

- Ant-quantity AS:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{d_{ij}} & \text{if link } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \text{otherwise} \end{cases}$$

AS: ELITIST PHEROMONE UPDATE

- The best ants add pheromone proportional to quality of their paths

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t) + n_e \Delta\tau^e_{ij}(t)$$

where

$$\Delta\tau^e_{ij}(t) = \begin{cases} \frac{Q}{f(\tilde{x}(t))} & \text{if } (i,j) \in \tilde{x}(t) \\ 0 & \text{otherwise} \end{cases}$$

e is the number of elite ants
$\tilde{x}(t)$ is the current best route

- Objective is to direct the search of all ants to construct a solution to contain links of the current best route(s)

ANT COLONY SYSTEM (1998)

- ACS addresses main AS' shortcomings and introduces new components
- A different transition rule is used
- A different pheromone update rule is defined
- Step-by-step local pheromone updates are introduced
- Candidate lists are used to favor specific nodes and save a lot of computation (at each step, check among $n \ll |\boldsymbol{E}|$ possible decisions, $|\boldsymbol{E}|$ can easily be 10^k, $k > 3$)
- Later (and more performing) versions make use of a daemon component based on local search → <u>SoA heuristic for TSP and similar problems</u>

ANT COLONY SYSTEM (1996)

Function AntColonySystem()

Pheromone model: $\tau_{ij} \equiv$ (estimated) quality of selecting city j when i is the current city

Heuristic variables: $\eta_{ij} \equiv 1$ / distance from city i to city j

$m \leftarrow$ number of ants per iteration (i.e., samples for *policy evaluation*)

$nn_{tour} \leftarrow$ find initial solution with nearest neighbor heuristic

Init pheromone: $\tau_{ij} = \tau_0 = 1/(n \cdot nn_{tour}),\ \forall i, j \in \{1, \ldots, n\}$

for $t := 1, \ldots iterations_num$

in-parallel for $k := 1, \ldots, m$ /* Ants construct solutions in parallel */

$T^k(t), L^k(t) \leftarrow$ ant_construct_solution($online_step_by_step_pheromone_update$)

/* Only best ant tour generated so far is selected for pheromone update */

best_so_far_ant_update_pheromone($\{T^k(l), L^k(l)\}, l = 1, \ldots, t,\ k = 1, \ldots, m$)

return best_solution_generated

ACS: TRANSITION RULE

- The pseudo-random-proportional action rule:

$$j = \begin{cases} \arg\max_{u \in \mathcal{N}_i^k(t)} \{\tau_{iu}(t)\eta_{iu}^{\beta}(t)\} & \text{if } r \leq r_0 \\ J & \text{if } r > r_0 \end{cases}$$

ε-greedy policy

where $r \sim U(0,1)$, and $r_0 \in [0,1]$ is a user-specified parameter

- $J \in \mathcal{N}_i^k(t)$ is a node randomly selected according to probability

$$p_{iJ}^k(t) = \frac{\tau_{iJ}(t)\eta_{iJ}^{\beta}(t)}{\sum_{u \in \mathcal{N}_i^k} \tau_{iu}(t)\eta_{iu}^{\beta}(t)}$$

$\mathcal{N}_i^k(t)$ is a set of valid nodes to visit

ACS: EXPLOITATION - EXPLORATION TRADEOFF

- Transition rule creates a bias towards nodes connected by short links and with a large amount of pheromone
- Parameter r_0 is used to balance exploration and exploitation:
  - if $r \leq r_0$, the algorithm exploits by favoring the best edge
  - if $r > r_0$, the algorithm explores
  - the smaller the value of r_0, the less best links are exploited, while exploration is emphasized more
- The transition rule is the same as that of AS when $r > r_0$

ACS: PHEROMONE UPDATE AND EVAPORATION

We are looking for the **best**, not the "average"

- Global update rule:
  - Only the globally best ant, $x^+(t)$, is allowed to reinforce pheromone concentrations on the links of the corresponding best path

$$\tau_{ij}(t+1) = (1-\rho_1)\tau_{ij}(t) + \rho_1 \Delta\tau_{ij}(t)$$

where

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{1}{f(x^+(t))} & \text{if } (i,j) \in x^+(t) \\ 0 & \text{otherwise} \end{cases}$$

with $f(x^+(t)) = |x^+(t)|$, in the case of finding shortest paths

  - Favors exploitation
  - $x^+(t)$ as the iteration-best vs global-best

ACS: PHEROMONE UPDATE AND EVAPORATION

- **Persistence, conservative approach:** For small values of ρ_1, the existing pheromone concentrations on the edges evaporate slowly, while the influence of the best route is dampened

- **Volatile, aggressive approach:** For large values of ρ_1, previous pheromone deposits evaporate rapidly, but the influence of the best path is emphasized

- The effect of large ρ_1 is that previous experience is neglected in favor of more recent experiences → more **exploration**

- **Simulated Annealing approach:** If ρ_1 is adjusted dynamically from large to small values, exploration is favored in the initial iterations of the search, while focusing on exploiting the best found paths in the later iterations

ACS: ONLINE PHEROMONE UPDATE

A "good" choice is potentially made locally "less good" after being selected. This is to favor exploring other local choices during the same iteration loop

- Local update rule:
  - Applied by each ant as soon as a new link is added to the path:

$$\tau_{ij}(t) = (1 - \rho_2)\tau_{ij}(t) + \rho_2\tau_0$$

with ρ_2 also in $(0, 1)$, and τ_0 is a small positive constant

Pheromones don't go to zero!

ACS: CANDIDATE LISTS

- $\mathcal{N}_i^k(t)$ is organized to contain a list of candidate nodes
- Candidate nodes are preferred nodes, to be visited first
- Let $n_l < |\mathcal{N}_i^k(t)|$ denote the number of nodes in the candidate list
- The n_l nodes closest to node i, i.t.o. cost, are included in the candidate list and ordered by increasing distance
- When a next node is selected, the best node in the candidate list is selected
- If the candidate list is empty, then node j is selected from the remainder of $\mathcal{N}_i^k(t)$

ACS: (OLD) PERFORMANCE (1997)

TSP problems from **TSPLIB** http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html

Problem name	ACS	GA	EP	SA	Optimum
Eil50	425 (427.96) [1,830]	428 (N/A) [25,000]	426 (427.86) [100,000]	443 (N/A) [68,512]	425 (N/A)
Eil75	535 (542.37) [3,480]	545 (N/A) [80,000]	542 (549.18) [325,000]	580 (N/A) [173,250]	535 (N/A)
KroA100	21,282 (21,285.44) [4,820]	21,761 (N/A) [103,000]	N/A (N/A) [N/A]	N/A (N/A) [N/A]	21,282 (N/A)

Euclidean TSP instances

GA = Genetic algorithm

EP = Evolutionary programming

SA = Simulated annealing

Table shows the best integer tour length, the best real tour length (in parentheses), and the number of tours required to find the best integer tour length (in square brackets)

Results out of 25 trials

ACS: (OLD) PERFORMANCE (1997)

TSP problems from **TSPLIB** http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html

Problem name	ACS best integer length (1)	ACS number of tours generated to best	ACS average integer length	Standard deviation	Optimum (2)	Relative error $\frac{(1)-(2)}{(2)} * 100$	CPU sec to generate a tour
d198 (198-city problem)	15,888	585,000	16,054	71	15,780	0.68 %	0.02
pcb442 (442-city problem)	51,268	595,000	51,690	188	50,779	0.96 %	0.05
att532 (532-city problem)	28,147	830,658	28,523	275	27,686	1.67 %	0.07
rat783 (783-city problem)	9,015	991,276	9,066	28	8,806	2.37 %	0.13
fl1577 (1577-city problem)	22,977	942,000	23,163	116	[22,204 – 22,249]	3.27÷3.48 %	0.48

ACS: DAEMON ACTION, LOCAL SEARCH

Function AntColonySystem-3-Opt()

Pheromone model: $\tau_{ij} \equiv$ (estimated) quality of selecting city j when i is the current city

Heuristic variables: $\eta_{ij} \equiv 1$ / distance from city i to city j

$m \leftarrow$ number of ants per iteration (i.e., samples for *policy evaluation*)

$nn_{tour} \leftarrow$ find initial solution with nearest neighbor heuristic

Init pheromone: $\tau_{ij} = \tau_0 = 1/(n \cdot nn_{tour}),\ \forall i, j \in \{1, \ldots, n\}$

for $t := 1, \ldots iterations_num$

in-parallel for $k := 1, \ldots, m$ /* Ants construct solutions in parallel */

$T^k(t), L^k(t) \leftarrow$ ant_construct_solution($online_step_by_step_pheromone_update$)

foreach $T^k(t),\ k := 1, \ldots, m$

/* Each ant solution becomes a starting point for 3-opt local search */

$T^k(t), L^k(t) \leftarrow$ run_3-opt_local_search_starting_from_ant_tour($T^k(t)$)

/* Only best ant tour generated so far is selected for pheromone update */

best_so_far_ant_update_pheromone($\{T^k(l), L^k(l)\}, l = 1, \ldots, t,\ k = 1, \ldots, m$)

return best_solution_generated

ACS: DAEMON ACTION, LOCAL SEARCH

- **At the end of each iteration, a local search is applied to <u>all tours</u> built by ants**
- The resulting iteration (or global so far) best tour gets pheromone updating
- Selected LS: 3-Opt
- Computationally expensive, but rewarding!

Symmetric TSP instances from the First International Contest on Evolutionary Optimization, IEEE-EC, May 20–22, 1996, Nagoya, Japan

STSP GA is a GA with a Lin-Kernighan local search (called right after the crossover operator in order to produced a population of locally optimized individuals)

Problem name	ACS-3-opt best result (length)	ACS-3-opt best result (sec)	ACS-3-opt average (length) (1)	ACS-3-opt average (sec)	Optimum (2)	% Error (1)-(2) ------ (2)
d198 (198-city problem)	15,780	16	15,781.7	238	15,780	0.01 %
lin318* (318-city problem)	42,029	101	42,029	537	42,029	0.00 %
att532 (532-city problem)	27,693	133	27,718.2	810	27,686	0.11 %
rat783 (783-city problem)	8,818	1,317	8,837.9	1,280	8,806	0.36 %

Problem name	ACS-3-opt average (length) (1)	ACS-3-opt average (sec)	ACS-3-opt % error (1)-(3) ------ (3)	STSP-GA average (length) (2)	STSP-GA average (sec)	STSP-GA % error (2)-(3) ------ (3)	Optimum (3)
d198 (198-city problem)	15,781.7	238	0.01 %	15,780	253	0.00 %	15,780
lin318 (318-city problem)	42,029	537	0.00 %	42,029	2,054	0.00 %	42,029
att532 (532-city problem)	27,718.2	810	0.11 %	27,693.7	11,780	0.03 %	27,686
rat783 (783-city problem)	8,837.9	1,280	0.36 %	8,807.3	21,210	0.01 %	8,806

ACS: DAEMON ACTION, LOCAL SEARCH

Asymmetric TSP instances (more difficult!) from the First International Contest on Evolutionary Optimization, IEEE-EC, May 20–22, 1996, Nagoya, Japan

ATSP-GA is a genetic algorithm + local search

Problem name	ACS-3-opt average (length) (1)	ACS-3-opt average (sec)	ACS-3-opt % error (1)-(3) ------ (3)	ATSP-GA average (length) (2)	ATSP-GA average (sec)	ATSP-GA % error (2)-(3) ------ (3)
p43 (43-city problem)	2,810	2	0.00 %	2,810	10	0.00 %
ry48p (48-city problem)	14,422	19	0.00 %	14,440	30	0.12 %
ft70 (70-city problem)	38,679.8	6	0.02 %	38,683.8	639	0.03 %
kro124p (100-city problem)	36,230	25	0.00 %	36,235.3	115	0.01 %
ftv170 (170-city problem)	2,755	68	0.00 %	2,766.1	211	0.40 %