AN INTRODUCTION TO ANT SYSTEMS

Éric Taillard

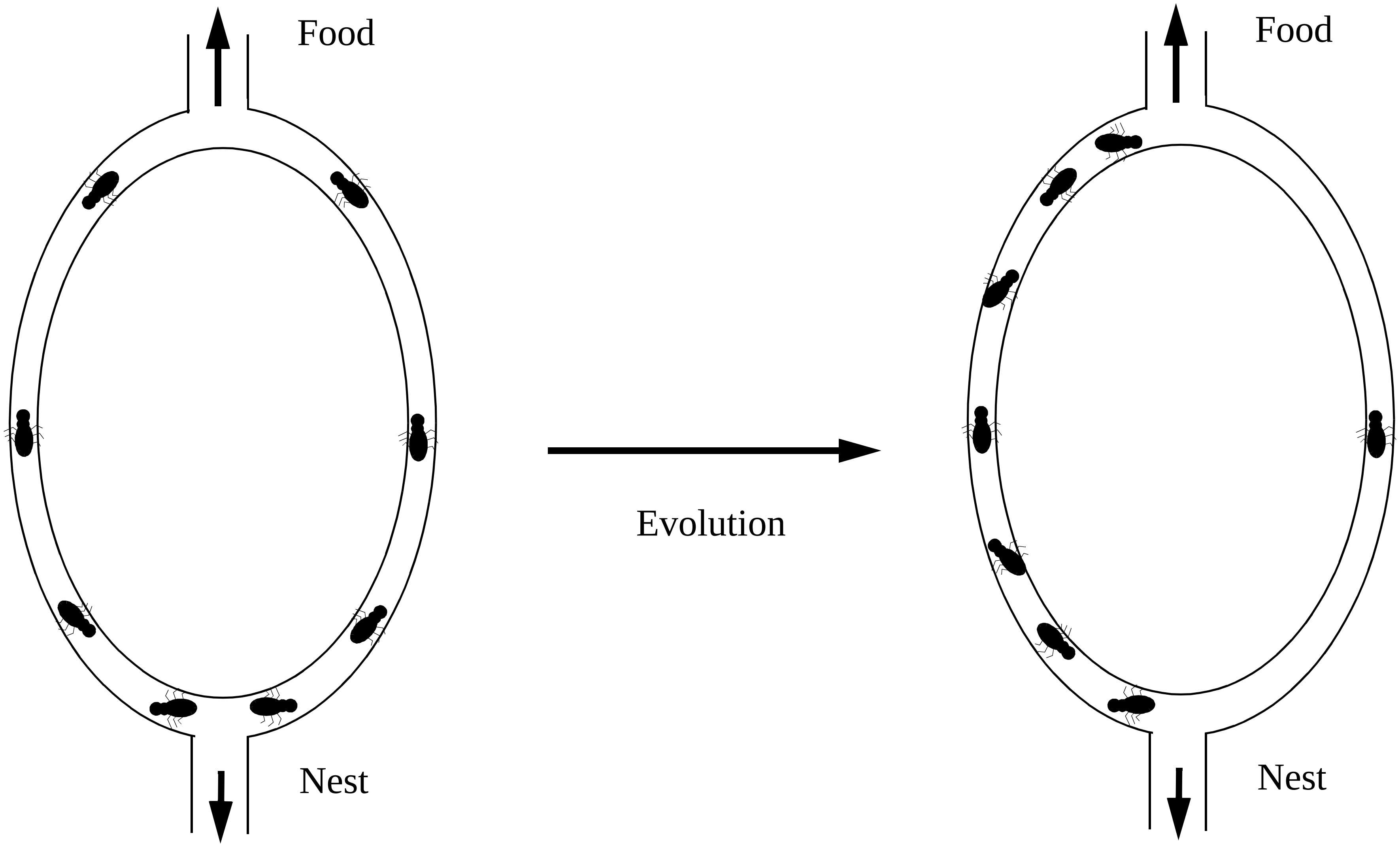
eivd

University of Applied Sciences of Western Switzerland,

Yverdon-les-bains, Switzerland

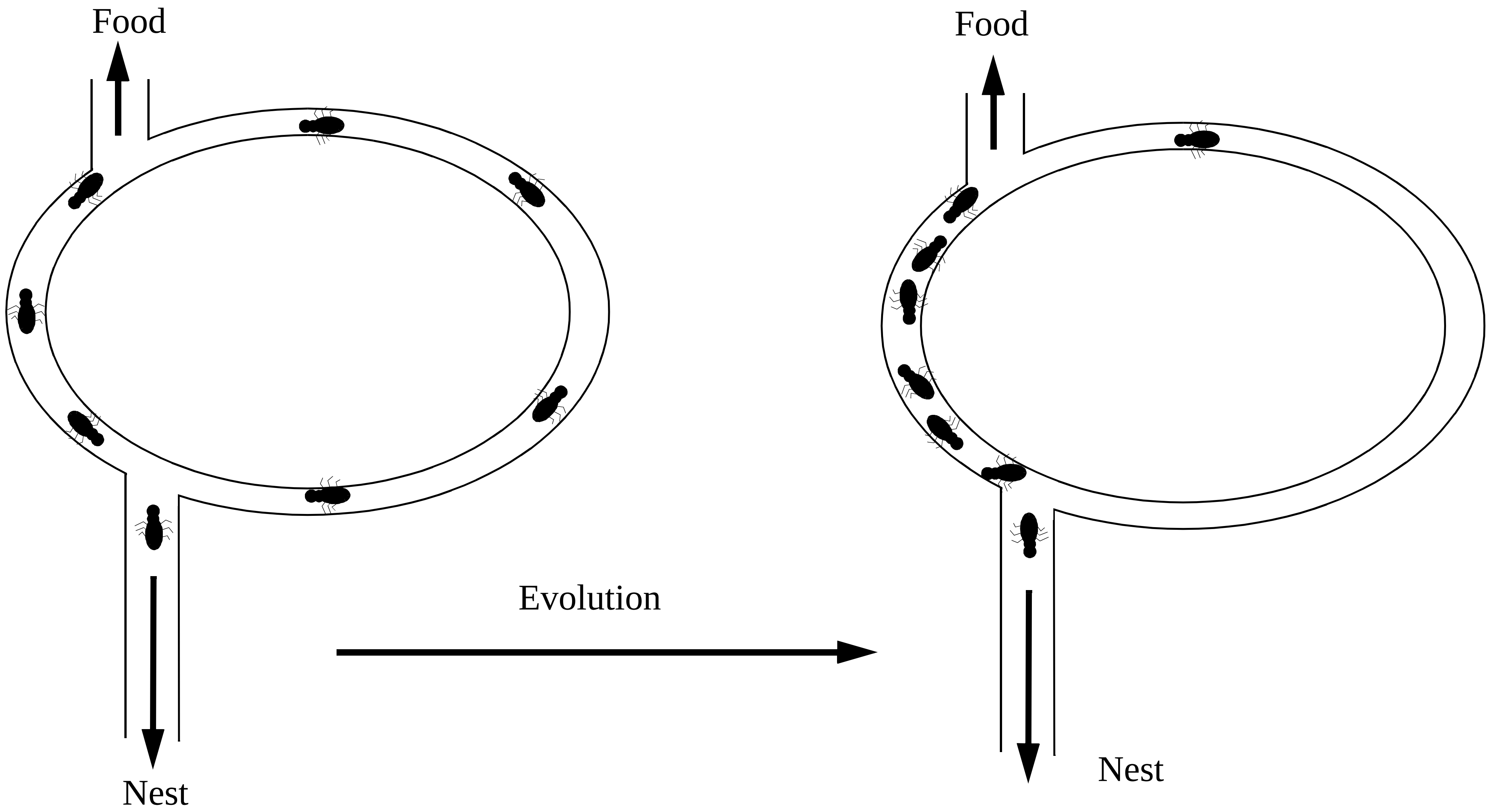
EXPERIMENTS ON REAL ANTS

Goss et al. 1989.

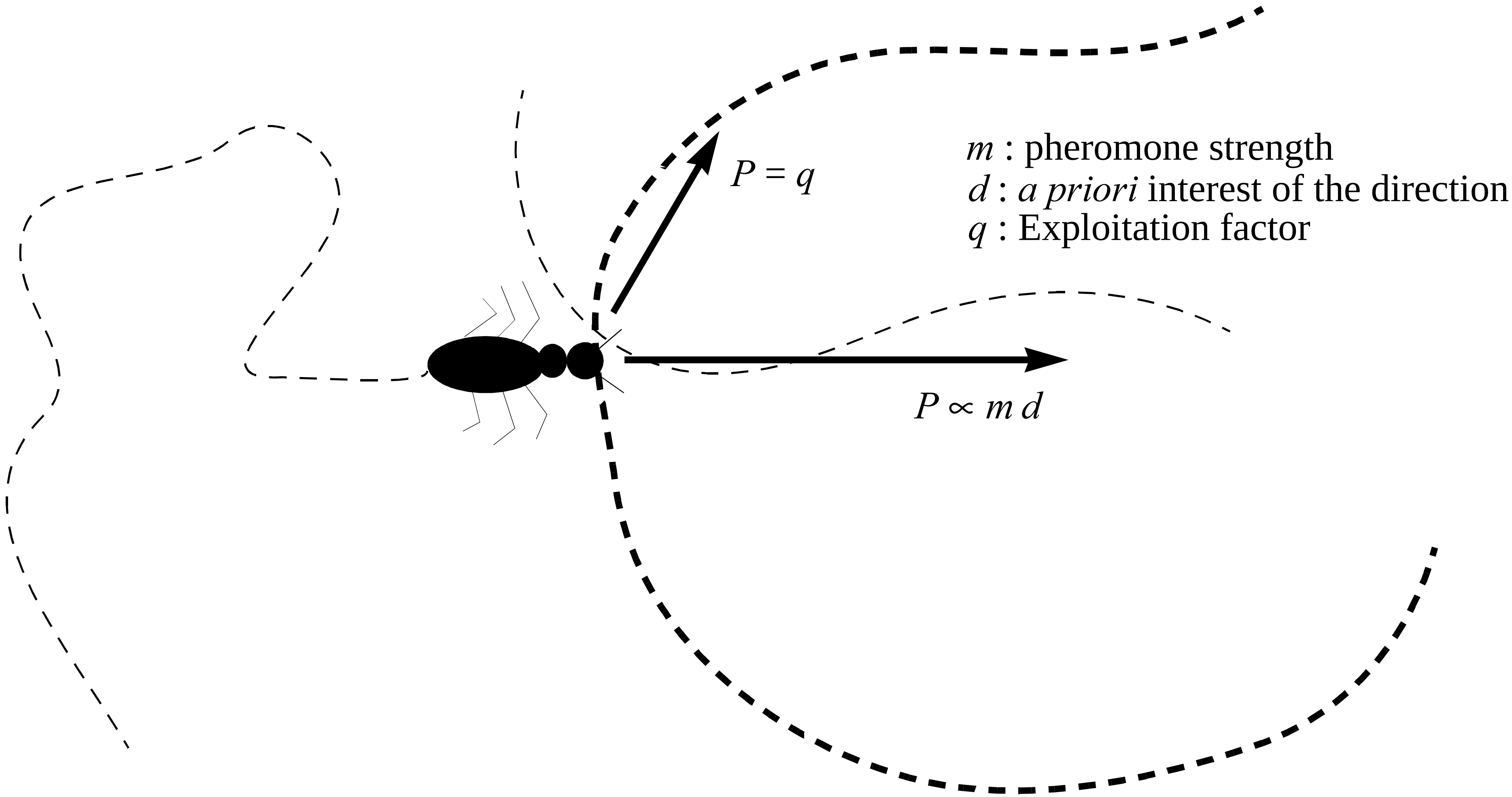


EXPERIMENTS ON REAL ANTS

Deneubourg et al. 1990.



MATHEMATICAL MODEL



SEMINAL TSP APPLICATION

Colorni, Dorigo, Maniezzo (1992)

Build a solution edge after edge.

Magic formulæ :

$P \propto m_e^{\alpha} d_e^{\beta}$, where m_e : pheromone strength, d_e : length of edge e, $0 \leq \alpha, \beta \leq 0$ parameters

Pheromone strength update :

$m_\mathrm{e} \leftarrow \rho m_\mathrm{e}$, for all edge e where $0 \leq \rho \leq 1$: parameter simulating pheromone evaporation

$m_e \leftarrow m_e + Q/L$, for all edge e belonging to a solution of length L built, Q another parameter

Evolution of seminal work :

More elaborated construction procedure (candidate list, additional parameters)

Improvement of the solutions with local search

Intensification and diversification strategies for managing memory $\boldsymbol{m}$.

ARTIFICIAL ANT SYSTEMS

Ant process

Receive problem data, memory state **M** + parameters from Queen process

Build a new solution to the problem probabilistically with the help of the memory.

Send new solution to Queen process

Queen process

Initialize memory **M**

Repeat (in parallel) until a stopping criterion is met

- Choose parameters for a new Ant process and activate it.
- Receive a solution from an Ant process and update memory **M**

Return best solution produced by the system

Queen process ≈ ACO process of Dorigo & Di Caro, 1999

QUADRATIC ASSIGNMENT PROBLEM (QAP)

Data : n×n matrices A = (a_{ij}), B = (b_{rs})

A : flows matrix (units), B : distance matrix (locations)

Objective function : find a permutation π minimizing $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}$

Ant system memory for the QAP : Matrix M = (m_{ir}) = statistic on $\pi_i = r$

MMAS : MAX-MIN ANT SYSTEM

Stützle and Hoos, 1999

Basic idea:

Each entry m_{ir} of the memory is constrained to belong to the interval $m_{min} \leq m_{ir} \leq m_{max}$.

Ant process :

Repeat n times

Select a facility i not yet assigned

Assigned i to location r with probability :

q if $m_{ir} = \max_k m_{ik}$ (exploitation)

$\propto m_{ir}$ otherwise (exploration)

Improve solution with local search (greedy or taboo search)

MMAS : Max-Min Ant System

“Queen process”

Memory initialization $m_{ir} = m_{max}\ \forall\ i, r$

Repeat :

Activate k Ant processes

Wait for k solutions $\pi^{1}, \ldots, \pi^{k}$ (π^{b} is the best among them)

Memory update

$\mathbf{M} \leftarrow \rho\, \mathbf{M}$;

$m_{i\pi_i^b} \leftarrow m_{i\pi_i^b} + 1/c^b$ every odd iteration

$m_{i\pi_i^*} \leftarrow m_{i\pi_i^*} + 1/c^*$ every even iteration (π^* best solution found so far)

$m_{ir} \leftarrow max(min(m_{ir}, m_{max}), m_{min}) \forall i, r$

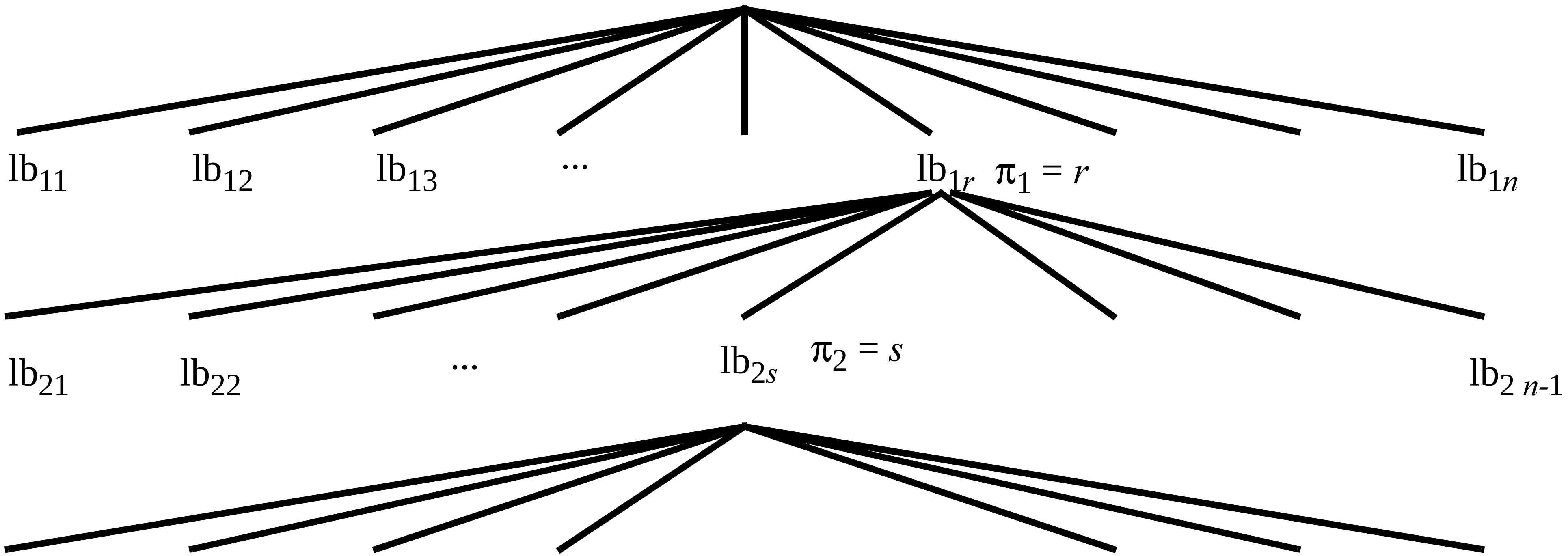
APPROXIMATE NONDETERMINISTIC TREE SEARCH

(ANTS)

Maniezzo, 1998

Basic idea :

At each construction step : compute a lower bound influencing ant's choice



ANTS

Queen process

Compute a lower bound *LB* to the optimum solution cost by linear relaxation

Re-order indices (i) by decreasing value of dual variables associated to locations

Set $m_{ir}^{0} = f$(Dual variables)

Repeat, until a stopping criterion is met :

Activate k Ant processes

Wait for k solutions $\pi^1, \ldots, \pi^k$ from the Ant processes (cost of π^b : c^b)

Compute average solution value $\bar{c}$ of the last p solutions received

Set $$m_{i\pi_i^b} \leftarrow m_{i\pi_i^b} + m^0_{i\pi_i^b} \cdot \frac{\bar{c} - c^b}{\bar{c} - LB} \forall i, b$$

ANTS

Ant process

For $i = 1$ to n do :

Compute the lower bound LB_{ir} of assigning unit i to location r

(given that units $1, \ldots, i - 1$ are already assigned)

Set $\pi_i = r$ with probability proportional to : $\alpha m_{ir} + (1 - \alpha) LB_{ir}$

Improve π with local search

Return π to Queen process

HYBRID ANT SYSTEM (HAS)

Gambardella, Taillard, Dorigo (1999)

Ant process (M, π)

Repeat p times :

Select i randomly, uniformly in 1, …, n

Select j probabilistically :

With probability q : j such that $m_{i\pi_j} + m_{j\pi_i}$ is maximum

j in 1, …, n with probability proportional to $m_{i\pi_j} + m_{j\pi_i}$

Swap π_i and π_j

Improve π with fast local search $\rightarrow \mu$

Return μ to Queen process

HAS

Queen process

$m_{ir} = \varepsilon$

For $b = 1, \ldots k$

Generate π^b randomly, uniformly ; improve π^b with a local search

π^* : best solution in π^b , $b = 1, \ldots k$

Repeat, until stopping criterion is met :

Activate k Ant processes (M, π^b) $(b = 1, \ldots, k)$

Receive k solutions μ^b $(b = 1, \ldots, k)$

Update π^*

$\mathbf{M} \leftarrow \rho \mathbf{M}$

$$m_{i\pi_i^*} \leftarrow m_{i\pi_i^*} + 1/c^*$$

Update pool of solution π^b , $b = 1, \ldots, k$ with intensification and diversification strategies

FAST ANT SYSTEM (FANT)

Taillard, 1998

Basic idea : Simple Ant Systems, incorporating self-adaptive intensification and diversification.

Ant process

Build a new solution randomly ; probability of setting $\pi_i = r \propto m_{ir}$

Improve π with local search ; return π to Queen process

Queen process

Set $v \leftarrow 1$ and $m_{ir} \leftarrow v \; \forall i, r$

Repeat, until stopping criterion is met :

Activate an Ant process

Receive π from Ant process

If $\pi = \pi^*$, set $v \leftarrow v + 1$ and $m_{ir} \leftarrow v \; \forall i, r$ (diversification)

If π better than π^*, set $\pi^* \leftarrow \pi$; $v \leftarrow 1$ and $m_{ir} \leftarrow v \; \forall i, r$ (intensification)

Set $m_{i\pi_i} \leftarrow m_{i\pi_i} + v$ and $m_{i\pi_i^*} \leftarrow m_{i\pi_i^*} + R, \forall i$

BI-QUADRATIC ASSIGNMENT PROBLEM

Problem formulation:

$$\min \sum_h \sum_i \sum_j \sum_k \sum_r \sum_s \sum_t \sum_u a_{hijk} b_{rstu} x_{hr} x_{is} x_{jt} x_{ku}$$

$$\text{s. t. :} \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j$$

NUMERICAL RESULTS (BI-QAP)

Best method previously available : GRASP Mavridou, Pardalos, Pitsoulis, Resende 1998.

Problem name	n	Optimal value	FANT			GRASP			Time [seconds]	
			Min	Avg	Max	Min	Avg	Max	FANT	GRASP
biQAP_10	10	133216	1	1.5	4	1	1.7	2	0.08	2.3
biQAP_12	12	120600	2	7.1	15	1	7.2	18	1.16	23.1
biQAP_14	14	251356	1	6.8	24	1	2.2	4	3.47	23.9
biQAP_16	16	441696	1	9.5	38	2	13.4	34	13.4	289.6
biQAP_18	18	811338	3	8.8	22	1	2.8	5	35.0	138.7
biQAP_20	20	1487296	1	2.4	5	1	2.0	5	17.0	184.2
biQAP_22	22	3118716	1	2.3	4	1	1.5	2	27.0	267.1
biQAP_24	24	1780378	3	41.7	149	1	28.2	66	713	8092.2
biQAP_26	26	5305872	1	1.9	4	1	3.1	9	45.0	1664.3
biQAP_28	28	4606532	1	5.8	13	1	4.7	15	199	4130.4
biQAP_30	30	4468860	2	43.2	276	4	27.3	138	2007	30344.0
biQAP_32	32	8088327	2	7	20	1	6.6	16	451	12585.1
biQAP_34	34	9445329	2	15.2	29	4	14.3	63	1371	36936.5
biQAP_36	36	11297520	2	21.3	51	2	21.8	58	2374	79281.6

Table 1 :Comparison of FANT and GRASP for the bi-quadratic assignment problem. Number of iterations for getting optimal value and CPU time on SGI R10000 180MHz (FANT) and SGI R4400 150MHz (GRASP). Problem instances of Burkard and Çela (1995)

NUMERICAL RESULTS (*P*-MEDIAN)

Best methods available

Heuristic Concentration (HC, Rosing, 1998, private comm.)

Taboo Search (TS, Rolland, Schilling, Current, 1997)

Variable Neighbourhood Search (VNS, Hansen, Mladenovic 1998)

n	p	Best known (FANT)	Quality (% over best known)					Time (seconds)			
			FANT 15	FANT 40	VNS	HC	TS	FANT 15	FANT 40	VNS	TS
200	10	48912	0.70	0.04	4.02	0.00	0.68	87	233	279	382
200	15	31153	1.61	0.31	5.46	0.00	2.80	134	358	289	401
200	20	23323	2.22	1.60	5.49	0.65	0.74	196	520	318	416
300	10	82664	2.32	0.95	6.40	0.00	0.47	231	614	504	1241
300	15	52772	3.55	2.71	4.47	0.00	1.81	332	884	492	1322
300	20	38244	3.50	2.69	6.08	1.35	2.49	467	1243	518	1378
400	10	123464	3.70	1.67	3.93	0.00	3.79	446	1187	816	2911
400	15	79872	4.49	2.73	4.86	0.75	5.15	672	1789	740	3097
400	20	58408	2.98	2.05	5.00	0.96	1.26	956	2559	743	3218
500	10	150112	1.92	0.57	3.76	0.00	1.52	699	1863	1136	9732
500	15	97624	2.72	1.30	4.62	0.00	0.79	1010	2702	1022	9731
500	20	72714	3.80	2.80	5.32	0.60	0.60	1420	3780	1015	9748
		Average	2.79	1.61	4.95	0.29	1.84	554	1478	655	3631

Table 2 :Comparison of FANT (15 and 40 iterations) with VNS (100000 iterations), HC and TS. FANT and re-implementation of VNS was run on Sun Sparc 5, TS on SGI R4400 144MHz. Problem instance of Rolland et al. 1997. All best solution known obtained by long FANT runs.

COMPARING NON-DETERMINISTIC HEURISTICS

Experiments : Run every of the *k* heuristics *b* times for the same problem instance (long runs)

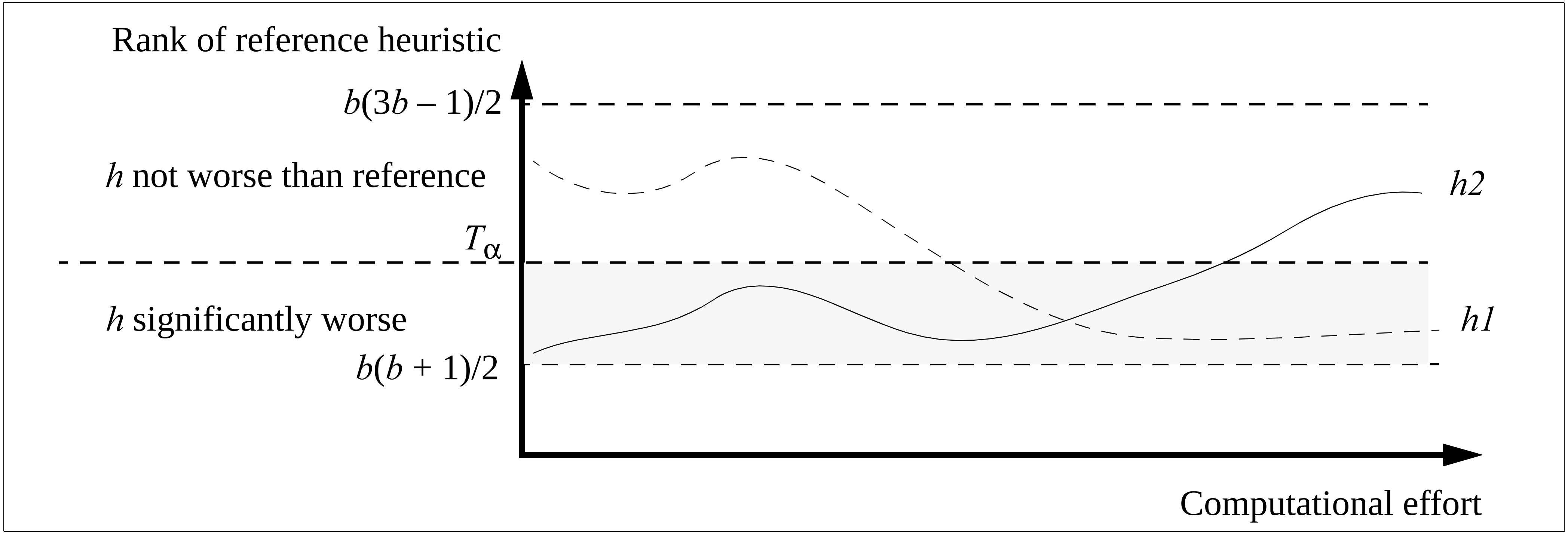
Reference heuristic (classical characterization in OR) : best average solution value.

Question : Is heuristic *h* significantly worse than reference heuristic ?

Answer : Mann-Whitney test.

Rank the $2b$ runs of reference + h heuristics by decreasing quality

If $\sum$ ranks of reference heuristic > $T_\alpha(b)$, reject hypothesis h is worse (α : confidence level).



COMPACT VIEW