Natural Computing

Lecture 14

J. Michael Herrmann
mherrman@inf.ed.ac.uk
phone: 0131 6 517177
Informatics Forum 1.42

4/11/2011

The canonical PSO algorithm

For each particle (for all members in the swarm) $i = 1 \ldots n$

- Create random vectors r_1, r_2 with components in $U[0,1]$
- update velocities

$$v_i \leftarrow \omega v_i + \alpha_1 r_1 \circ (\hat{x}_i - x_i) + \alpha_2 r_2 \circ (\hat{g} - x_i)$$

$\circ$: componentwise multiplication

- update positions

$$x_i \leftarrow x_i + v_i$$

- update local bests (for a minimisation problem)
  $\hat{x}_i \leftarrow x_i$ if $f(x_i) < f(\hat{x}_i)$
- update global best (for a minimisation problem)
  $\hat{g} \leftarrow x_i$ if $f(x_i) < f(\hat{g})$

Overview

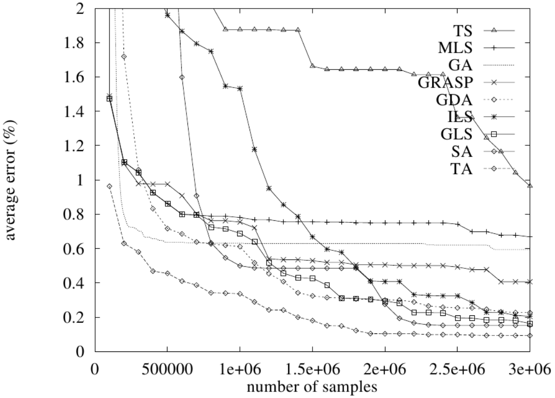
- Standard meta-heuristic algorithms
- Bees, frogs, fireflies, bats, cuckoos, eagles
- Comparison of metaheuristic algorithms
- Intensification and diversification
- A bit of genetics

Related optimisation methods

... not quite biologically inspired

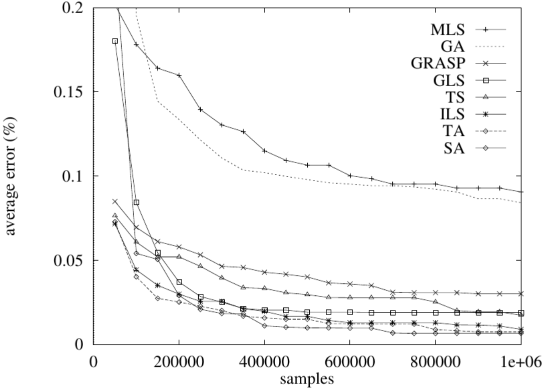
- Random multi-start local search (possibly with recorded search history)
- Simulated annealing: Random with a time-dependent acceptance rate of deteriorations,
  - acceptance probability $e^{-|\Delta|/T}$ where Δ is the fitness difference (always accept improvements), T follows a cooling schedule
  - variant threshold accepting: simplification of SA that accepts worsening if it is below a (time-dependent) threshold
- Greedy Randomized Adaptive Search Procedures (GRASP): Randomized greedy method to generate initial solutions for local search; iterated local search; variable neighbourhood search
- Taboo search: Modify fitness function such that previously found optima are removed, e.g. great deluge algorithm

Comparison on the single machine scheduling problem



M. YAGIURA and T. IBARAKI: On Metaheuristic Algorithms for Combinatorial Optimization Problems (2001).

Comparison on the MAXSAT problem



M. YAGIURA and T. IBARAKI: On Metaheuristic Algorithms for Combinatorial Optimization Problems (2001).

Comparison among ME algorithms

- Global bests in a standard set of benchmark problems based on a standard solution quality metrics (neither is agreed upon)
- Comparisons are not always meaningful
  - Standard data sets are simple,
  - Data sets are pragmatically selected
  - Even with best intentions one's own algorithm will be better tuned than the algorithm of a competitor
- Open competitions are an option
  - Preparation: Parameter adaptation on a given dataset
  - Competition: Test on a similar but unknown data set with manual readjustment of parameters
- Asymptotic space and time complexity (e.g. runtime growth rate)
- Dimension and sensitivity of the parameter space

J. Silberholz and B: Golden: Comparison of Metaheuristics. in Handbook of Metaheuristics 2010, Vol. 146, 625-640.

Principles for comparisons

First experimental principle: The problems used for assessing the performance of an algorithm cannot be used in the development of the algorithm itself.

Second experimental principle: The designer can take into account any available domain-specific knowledge as well as make use of pilot studies on similar problems.

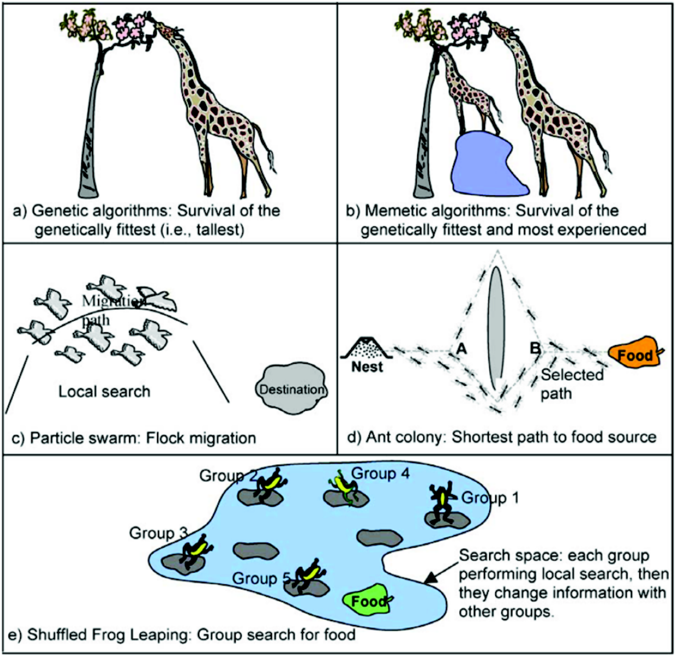
Third experimental principle: When comparing several algorithms, all the algorithms should make use of the available domain-specific knowledge, and equal computational effort should be invested in all the pilot studies. Similarly, in the test phase, all the algorithms should be compared on an equal computing time basis.

Mauro Birattari, Mark Zlochinand Marco Dorigo: Toward a theory of practice in metaheuristic design: A machine learning perspective. RAIRO-Inf. Theor. Appl. 40 (2006) 353-369.

Bees, frogs, fireflies, bats, cuckoos, eagles

- Honey bee algorithm: A bee directs others to nectar sources in dependence on its previous success (cf. ACO)
- Fireflies algorithm: Fireflies attract others by an inverse square law of the "light intensity" (i.e. fitness) (cf. ACO)
- Bat algorithm: Bats fly with a velocity that depends on their "wave length" (i.e. fitness), but can change also loudness and duration of the pulse etc. (cf. PSO)
- Frog leaping algorithm: Out of several subgroups of frogs the best ones are allowed to "jump", i.e. to exchange difference vectors (cf. DE)

see X-S Yang: Nature-inspired metaheuristic algorithms. Luniver Press 2010.



Emad Elbeltagi, Tarek Hegazy, Donald Grierson (2005) Advanced Comparison among five evolutionary-based optimization algorithms. Engineering Informatics 19, 43–53.

Advanced Comparison

Results of the continuous optimization problems

Comparison criteria	Algorithm	Number of variables						
		F8				*EF10*		
		10	20	50	100	10	20	50
% Success	GAs (Evolver)	50	30	10	0	20	0	0
	MAs	90	100	100	100	100	70	0
	PSO	30	80	100	100	100	80	60
	ACO	–	–	–	–	–	–	–
	SFL	50	70	90	100	80	20	0
Mean solution	GAs (Evolver)	0.06	0.097	0.161	0.432	0.455	1.128	5.951
	MAs	0.014	0.013	0.011	0.009	0.014	0.068	0.552
	PSO	0.093	0.081	0.011	0.011	0.009	0.075	2.895
	ACO	–	–	–	–	–	–	–
	SFL	0.08	0.063	0.049	0.019	0.058	2.252	6.469

Results of the discrete optimization problem

Algorithm	Minimum project duration (days)	Average project duration (days)	Minimum cost ($)	Average cost ($)	% Success rate	Processing time (s)
GAs	113	120	162,270	164,772	0	16
MAs	110	114	161,270	162,495	20	21
PSO	110	112	161,270	161,940	60	15
ACO	110	122	161,270	166,675	20	10
SFL	112	123	162,020	166,045	0	15

Emad Elbeltagi, Tarek Hegazy, Donald Grierson (2005) Advanced Comparison among five evolutionary-based optimization algorithms. Engineering Informatics 19, 43–53.

Result of the comparison

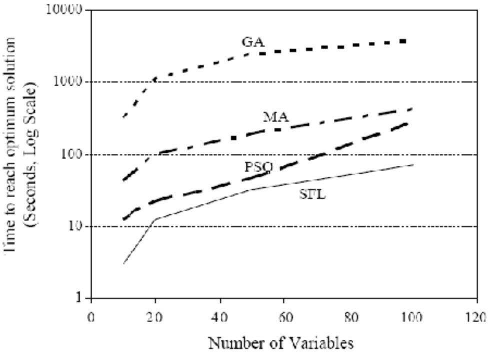


Fig. 5. Processing time to reach the optimum for $F8$ function.

. . . and the winner is

PSO

(check back for the next competition)

Example: Power Generation Expansion Planning

- Long-term behaviour of electricity markets
- Minimize the total investment and the operating cost of the generating units
- Meet the demand criteria, fuel mix ratio, and the reliability criteria
- Highly constrained, nonlinear, discrete optimization problem
- Solution through complete enumeration in the entire planning horizon
- System dynamics models for system behaviour: Detailed relationships between the main variables of the system with explicit recognition of feedbacks and delays.

S. Kannan, S. Mary Raja Slochanal, and Narayana Prasad Padhy: Application and Comparison of Metaheuristic Techniques to Generation Expansion Planning Problem. IEEE TRANSACTIONS ON POWER SYSTEMS 20:1, 2005.

Another competition

Algorithms:

Genetic algorithm
Differential evolution
Evolutionary programming
Evolution strategies
Ant colony optimization
Particle swarms
Taboo search
simulated annealing
Hybrid approach
(GA+direct search in linear span)

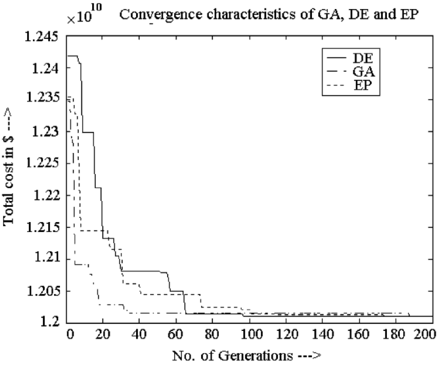
BEST PARAMETERS FOR 6-YEAR PLANNING HORIZON

Methods (Parameters)	Parameter values
GA (P_c, P_m, Crossover strategy)	(0.7, 0.15, stochastic crossover)
DE (P_c, F, Mutation strategy)	(0.5, 0.5, Rand/ Rand-Rand)
EP (β, Mutation strategy)	(0.02, Gaussian & Cauchy)
ES (∇)	Adaptive ∇
ACO (TotAnts)	25
PSO (w_{max}, C_1, C_2)	(0. 8, 1. 5, 1. 5)
TS (Neighbors, Tabu list)	(25, 5)
SA (Te, cooling schedule)	(400, 0. 95 × Te)
HA (P_c, P_m, Phase-2)	(0. 9, 0. 15, heuristic search)

RESULTS FOR 6-YEAR PLANNING HORIZON

Tech	Cost × 10^{10} \$		AFE	ANG	Error (%)	SR (%)	Exe. time (min)
	Best	Worst					
GA	1.2009	1.2024	10222	266	0-0.12	72	41.2
DE	1.2009	1.2009	12100	130	0	100	44
EP	1.2009	1.2012	9685	272	0-0.02	80	39.3
ES	1.2009	1.2024	7866	375	0-0.12	78	45.8
ACO	1.2009	1.2096	4560	165	0-0.74	18	90
PSO	1.2009	1.2014	9635	112	0-0.04	68	50
SA	1.2009	1.2086	5690	120	0-0.64	40	11.6
TS	1.2009	1.2024	7200	421	0-0.12	40	26.7
HA	1.2009	1.2012	14365	320	0-0.02	90	66
DP	1.2009				0	100	4.8

(Cost, # fitness evaluations, # generations, error range, success rate, execution time)



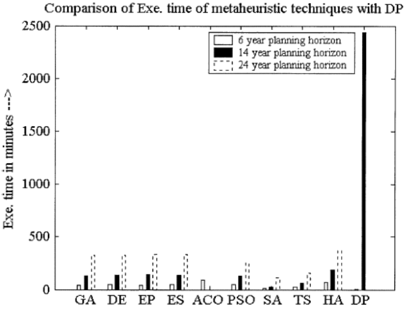
Long term results

Results for 14-Year Planning Horizon

Tech	Cost × 10^{10} \$		AFE	ANG	Error (%)	SR (%)	Exe. time (min)
	Best	Worst					
GA	2.1834	2.1979	12242	266	0.2-0.8	0	126.2
DE	2.1834	2.1957	14100	130	0.2-0.7	0	135.5
EP	2.1887	2.1988	10685	272	0.4-0.9	0	140.7
ES	2.1840	2.1957	8766	375	0.2-0.7	0	138.3
ACO	-	-	-	-	-	-	-
PSO	2.1859	2.1987	10635	112	0.3-0.9	0	125.3
SA	2.1924	2.2109	11690	120	0.6-1.4	0	23.6
TS	2.1858	2.2546	9200	421	0.3-3.4	0	64.6
HA	2.1797	2.1857	14365	320	0-0.3	12	185.7
DP	2.1797		-	-	0	100	2436

Results for 24-Year Planning Horizon

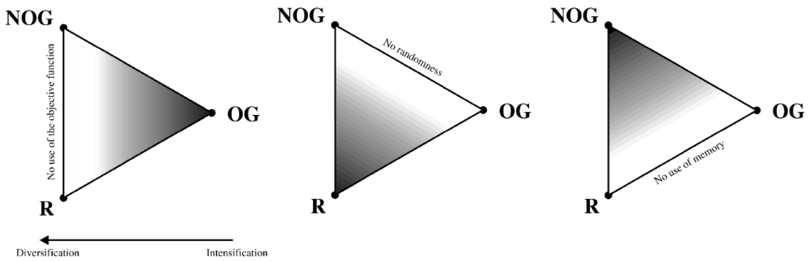
Tech	Cost × 10^{10} \$		AFE	ANG	Error (%)	SR (%)	Exe. time (Hrs)
	Best	Worst					
GA	2.9356	2.9970	14380	412	0.5-2.6	-	5.39
DE	2.9262	2.9527	15193	390	0-1.1	-	5.46
EP	2.9262	2.9556	14098	432	0.2-1.2	-	5.54
ES	2.9301	2.9671	11096	418	0.3-1.6	-	5.51
ACO	-	-	-	-	-	-	-
PSO	2.9279	2.9608	13147	256	0.2-1.4	-	3.15
SA	2.9262	2.9609	10110	187	0.2-1.4	-	0.98
TS	2.9825	3.0135	11223	449	2.1-3.2	-	2.68
HA	2.9206*	2.9334	16328	441	0- 0.43	-	3.42
DP	Unknown					-	-



Conclusion on Comparisons

- Tuning of all algorithm by generic methods
  - virtual mapping procedure (e.g. $n_1\times$ type A power station, $n_2\times$ type B: use variable n that Cantor enumerates the array formed by the pairs(n_1, n_2))
  - intelligent initial population generation (does not observe constraints but meets the demand plus a reserve margin)
  - penalty factor approach (constraints are penalised, but not deselected)
- Dynamic programming (DP) is optimal when computable
- Hybrid approach wins!
- Among the others: DE is best (perhaps because 4 vectors are "crossed over" instead of 2 in GA etc.)

Intensification and diversification



OG = solely guided by the objective function
NOG = solely guided by one or more function other than the objective function
R = solely guided by randomness

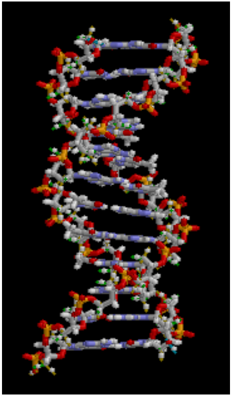
C. Blum & A. Roli: Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys* **35**:3, 2003, 268–308.

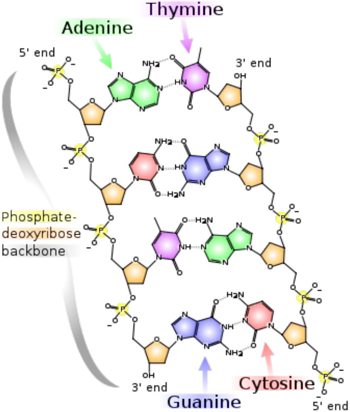
A conclusion on MHO: Intensification and diversification

"A metaheuristic will be successful on a given optimization problem if it can provide a balance between the exploitation of the accumulated search experience and the exploration of the search space to identify regions with high quality solutions in a problem specific, near optimal way."

T. Stuetzle: *Local Search Algorithms for Combinatorial Problems—Analysis, Algorithms and New Applications.* DISKI. infix, St. Augustin, 1999.

The Genetic Code





James Watson, Francis Crick, Maurice Wilkins and Rosalind Franklin:
DNA structure (hypothesis 1953, Nobel Prize 1962)

The Genetic Code

DNA = deoxyribonucleic acid

DNA is made up of a chain of simple molecular units. Each unit comprises a base, a sugar and a phosphate. The sugars and phosphates in many units link together in a chain with the bases sticking out. The bases in two chains attract one another resulting in a double helix structure.

There are just 4 kinds of **base** in DNA, labelled A, C, G and T (adenine, cytosine, guanine, thymine). C and G pair up, as do A and T.

. . . GATTACCA . . .
. . . CTAATGGT . . .

George Gamow (1950s): Triplets as the elementary units of the genetic code (codons) [he wrongly assumed ambiguity: GGAC=GGA+GAC (please do not memorize!)]

Encoding Proteins

How does this work?

Sections of chromosome contain the instructions for building chains of amino acids – proteins. The proteins are the building blocks, regulation units and manufacturing units of the body:

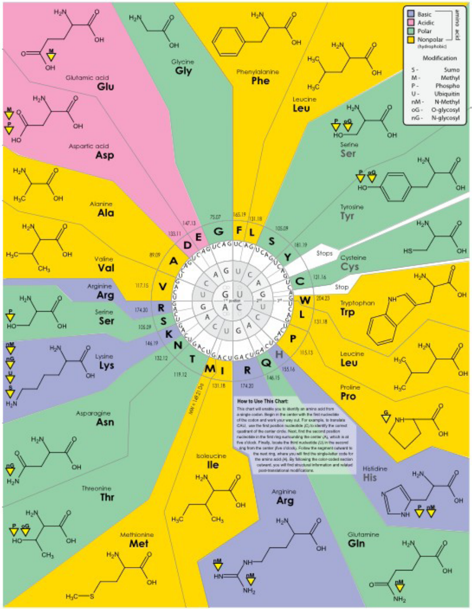
e.g. lactase (enzyme), collagen (structure), haemoglobin (oxygen transport), actin (muscle contractions), CLOCK protein (circadian rhythm regulation).

Encoding:	3 DNA bases	$\longrightarrow$	1 amino acid	AAA = lysine
	64 combinations	$\longrightarrow$	20 amino acids	CCC = proline
	– some redundancy			

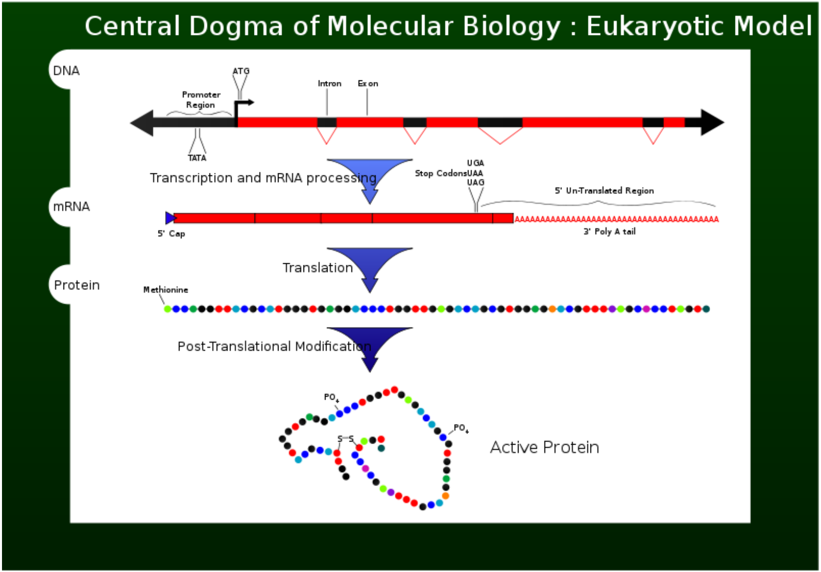
A protein is made up of many amino acids strung together and folded up.

Coding Principle

$4^3 = 64$
combinations
from
3 base pairs
Encoding
20 amino acids



Central Dogma of Molecular Biology



Original work by Mike Jones for wikipedia.

Central Dogma of Molecular Biology

- Enunciated by Francis Crick in 1958 (Nature 1970)
- “Information cannot be transferred back from protein to either protein or nucleic acid.”
- In other words, “once information gets into protein, it can’t flow back to nucleic acid.”

From: To	DNA	RNA	Protein
DNA	replication	reverse transcription	?
RNA	transcription	RNA replication	?
Protein	direct translation	translation	prions?

general transfer, special transfer, unknown

Outlook

- DNA Computing beyond metaphorics