Course: Natural Computing

7. Hybrid Metaheuristics and Hyperheuristics

J. Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

- Memetics
- Hybrid metaheuristics
- Hyperheuristics

What is Metaheuristics?

Given:

- a set S of potential solutions of a problem (search space)
- a fitness function $f : S \mapsto \mathbb{R}$ (bounded from above or below)

Find:

- a sampling procedure $G\left(\{s_i\}_{i=1}^{T-1}\right) \mapsto s_T$ such that $f(s_T)$ is likely to be near-optimal for $T = T_{\text{termination}}$

Metaheuristics focuses on problems where words as 'likely' or 'near-optimal' cannot be specified without unreasonable effort, i.e. for

- small data due to peculiarity, availability, or non-stationarity
- fuzzy problem formulation, unspecified data properties
- multi-objective problems without clear weights or preferences

Beyond Metaheuristics

What can be inferred from the set $\{s_i\}_{i=1}^{T-1}$ of previous samples?

- Search direction (based on a temporal coherence prior)
- Step width (based on a spatial smoothness prior)
- Prediction of new samples (temporal coherence and smoothness)
- Search dimensions (based on a compositionality prior)
- Search space size and granularity (non-occurrence of certain samples)

Some priors can be seen as implied by an anthropic bias.

Higher-level 'Memetic' algorithms (Chen, Ong, & Lim, 2010)

- Heuristics (single problem)
- Metaheuristics (problems of a certain type)
- Memetic Algorithms (1st level MA)
- Hybrid algorithms (compositional MA)
- Hyperheuristic (2nd level MA)
- Co-evolution and self-generating (3rd level MA)

[Co-evolutionary free lunch?]

Memetic algorithms (Moscato, 1989)

- Metaphor based on social evolution → *cultural algorithm*
- Includes both genetic and individual learning (similar to the Baldwin effect and Lamarckian evolution)
- Can be as simple as GA with ES for local search
- In principle, the memetic component of the MH needs to be developed in a social context (different from our representation of Baldwin and Lamarck)
- Can be considered a type of *hyperheuristic algorithms*, see below.

see e.g. Neri & Cotta (2012) Memetic algorithms and memetic computing optimization: A literature review.

Memetic algorithms (Moscato, 1989)

Approaches

- Lamarckian: Individual learning changes original representation
- Baldwinian: Individual learning is part of the fitness evaluation, and can support a genetic drift towards a typical result of learning
- Haeckelian: Social protection for some individuals
- Fisherian: "The rate of increase in the mean fitness of any organism at any time ascribable to natural selection acting through changes in gene frequencies is exactly equal to its genetic variance in fitness at that time." (Fundamental theorem of natural selection) [Edwards, 1994]
- Kauffmanian: Embracing a culture of diversity
- Wildtype vs. culturally adapted individuals (see islands, elites)
- Competition for learning ability

Memetic algorithms

More generally we can ask:

- What type of local search?
- Editing of the genome?
- Choice of offspring for local search?
- Frequency of local search?
- How intensive is the locale search?

W. Jakob, Memetic Comp., 2010

Hybrid metaheuristics

What is an (MHO) algorithm?

- A sequence of instructions that has a justification of purpose
- Algorithms can be variants or compositions of existing algorithms

Approaches to hybrid metaheuristics include

- Local search (as in memetic algorithms)
- Incorporation of elements from existing algorithms (see above)
- subpopulations or "seasons" where evolution follows different algorithms
- See also hybrid (biology) in Wikipedia

There is no clear distinction between hybrid metaheuristics and memetics, i.e. either one can be said to include the other: Memetics is hybridisation specifically by local search, whereas hybridisation can be seen as 1st level memetics.

Outlook

- As the "Banal" MH (Lect. 4s) suggests, it is not clear whether we talk about a hybrid of existing algorithms or about a new algorithm.
- We need some classification of hybridisation approaches.
- This will enable both a creative approach to design of new algorithms and also automatic search within the space of algorithms ($\Rightarrow$ hyperheuristics).

Hybridisation (Talbi: Metaheuristics, 2009)

Combining metaheuristics with

- Greedy heuristics
- Local search
- (complementary) metaheuristics
- Exact methods from mathematical programming
- Constraint programming approaches
- Machine learning and data mining techniques

Classification of hybrid metaheuristics

Many combinations are possible in a hybrid metaheuristics.
Talbi (2009) gives the following criteria

- Level
  - high
  - low
- Mode
  - relay
  - teamwork
- Type
  - homogeneous
  - heterogeneous
- Domain
  - global
  - partial
- Function
  - generalist
  - specialist
- Interaction
  - static
  - adaptive

Combining metaheuristics with (complementary) metaheuristics

- Level: A function of one MH is replaced by another MH (low) or two complete algorithms are cooperating (high)
- Mode: Two or more MHs are applied sequentially (relay) or is there a direct cooperation (teamwork)
- Type: Always the same metaheuristics is used (homogeneous) or a choice among several MHs is made (heterogeneous)
- Domain: All algorithms work on the same search space (global) or are working on different aspects of the problem, e.g. single objectives in MOO (partial)
- Function: All algorithms work on the full problem (generalist) or just on aspects such as diversification (specialist)
- Interaction: The combination is fixed (static) or depends on the runtime properties of the algorithm (adaptive)

Structure of hybrid metaheuristics

- Using local search can be seen as a low-level relay hybrid; e.g. in GP, a continuous metaheuristic can be used to find numerical constants
- GAs can include teamwork with taboo search (used in mutation to avoid already visited states) and a greedy heuristic (for crossover to improve off-spring)
- In heterogeneous hybrids a blackboard architecture can be used for communication among the algorithms
- Classical memetic algorithm can be considered as low-level teamwork hybrids

Examples: hybrid metaheuristics

- In partial heterogeneous hybrids, various diversifying agents receive solutions from regions explored by various intensifying agents to increase diversity while they in turn send suggestions to the intensifying agents
- Adaptation can mean to use a second MH (e.g. random search) when stagnation is detected
- Island algorithms allow for heterogeneous approaches
- Elitism invites different algorithms for the elite and the rest of the population

Examples: Hybrid metaheuristics

- Combining metaheuristics: e.g. a trajectory-based method (e.g. SA) with a population-based method (e.g. PSO)
- Constraint optimisation problem: limit search for MHO, guide search in AI
- Large neighbourhood search: Adapt range of neighbourhood
- Identify promising schemas in GA (and find specific mutation/crossover operators)
- Finding a decomposition of a dynamic programming problem

See C. Blum (2010) Hybrid Metaheuristics. Presentation at BIOMA, Ljubljana, Slovenia

Combining metaheuristics with AI and ML techniques

- Gradient-based algorithms need good starting values that can be provided by a diversifying MH
- Local search can be performed by (local) exact techniques
- Known strict bounds on the fitness can be used as admissibility criteria
- Use path finding algorithms to improve representation of a Pareto front found by a population
- Parameters of algorithms can be adapted by greedy search methods
- Mixed continuous integer optimisation

A unified view

Algorithm Pool Template

Initialize pool P by an external procedure;
while termination=FALSE **do**
  $S \leftarrow OF(P)$;
  if $|S| > 1$ **then**
    $S' \leftarrow SCM(S)$
  else
    $S' \leftarrow S$;
  $S'' \leftarrow IM(S')$;
  $P \leftarrow IF(S'')$;
Apply a post-optimizing procedure to P.

- P: Pool of solutions (S)
- IF: Input function
- OF: Output function
- IM: Improvement method
- SCM: Solution combination method

S. Voss (2006) Hybridizing metaheuristics. 6th Europ. Conf. Evol. Comput. in Comb. Optim.
G. R. Raidl (2006) A unified view on hybrid metaheuristics. Int. Workshop Hybrid Metaheuristics.

Hyperheuristics

The fundamental difference between metaheuristics and hyperheuristics is that most implementations of metaheuristics search within a search space of problem solutions, whereas hyperheuristics always search within a search space of heuristics.

https://en.wikipedia.org/wiki/Hyper-heuristic

In contrast to a hybrid metaheuristics, hyperheuristics can find a hybridisation automatically, but can adapt the algorithm also in a more general sense.

What about GP?

Hyperheuristics

- There are several hundreds of metaheuristics which can be modified by parameter settings, used mutually as "local" search algorithms, and combined into hybrids: Why search among solutions, if we can search among search algorithms?
- Hyperheuristics can adapt, select or combine metaheuristics, i.e. automatically designs metaheuristics on case-by-case basis
  - Use several MHO algorithms for a fraction of the fitness evaluation budget, then select the best one
  - Use a high-level algorithm to decide about low-level algorithms and their parameters narrowing the ensemble but continue to use them in parallel
  - Use a high-level algorithm to generate or design low-level heuristics
- "A hyper-heuristic is an automated methodology for selecting or generating heuristics to solve computational search problems." (Burke et al., 2019, Handbook of Metaheuristics. Ch. 14) or: "Heuristics to choose heuristics" (Cowling et al., 2000; see also Burke et al., 2013)

Hyperheuristics

- It does not escape from the NFL theorem. Bias depends on formulation, e.g. the first data (selecting MHO) are representative for later data (used by the MHO)
- Potential problems:
  - Is the size of the new search space reasonable?
  - Is the original fitness functions good enough?
  - Is the metaheuristics diverse enough?

If you give a man a fish he is hungry again in an hour. If you teach him to catch a fish you do him a good turn.

Anne Isabella Thackeray Ritchie

Hyperheuristics

Another dimension is related to the use of fitness values

- On-line learning HH: Request new fitnesses from the problem
  - costly, but flexible and potentially accurate
- Offline learning HH: Learn from a set of training examples
  - requires control of generalisation, large data or a model
  - fails at non-stationarity

GA as a Hyperheuristic

- E.g. consider PSO with many terms (repulsion, alignment, velocity control etc.)

  $$v_i = g_0 \omega v_i + g_1 \alpha_1 R_1 \text{Expression}_1 + g_2 \alpha_2 R_2 \text{Expression}_2 + \dots$$

  the $g_i$terms are included here to switch on or off any of these terms. Now, use GA to find optimal g_i.
- Construct various cross-over operators and mutations operators, use another GA to choose which ones to use.
- Write programs for many different MHO algorithms that operate on a joint problem representation. Now use a GA to select what combination of these algorithms to use.

GP as a Hyperheuristic

The suggestive idea to produce metaheuristic algorithms by GP has been studied already for more than a decade, but so far there is not too much progress.

See e.g.:

Keller and Poli (2007) Cost-benefit investigation of a genetic-programming hyper-heuristic.

Stützle and Manuel López-Ibáñez (2019) Automated design of metaheuristic algorithms. Handbook of Metaheuristics, Ch. 17.

Case study

Jorge M Cruz-Duarte et al. (2021) Hyper-Heuristics to customise metaheuristics for continuous optimisation. *Swarm and Evolutionary Computation* **66**, 100935.

What algorithms win the competitions?

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.

What algorithms win the competitions?

IEEE Congress on Evolutionary Computation: Competition winners

- Previously, single objective real-parameter optimization competitions with evaluation criteria specified beforehand
- Criteria change every year: Is there progress in the field?
- Properties of successful algorithms in the years 2013-2015 (after this mostly multi-objective optimisation), e.g.
  - ES + restart when collapsed with increasing population size
  - GA + multiple parent crossover
  - ES with covariance matrix adaptation (CMA-ES)
  - GA with CMA as local search
  - GA, DE and CMA-ES in parallel for first half of time
  - Mean-Variance Mapping Optimisation (MVMO)
  - Success History-based Adaptive DE (SHADE) and variants
- Result: If previous winners had taken part again, they may have won. Although this is not a significant result, it suggests that improvement is not very fast

Molina, Moreno-García, Herrera (2017) Analysis among winners of different IEEE CEC competitions on Real-parameters Optimization: Is there always improvement?

Conclusions

- Employ a problem-oriented approach
  - what has been attempted previously, where are strengths and weaknesses
  - in what case synergies can be expected
- Keep in mind that
  - hybrid MHO algorithms as well as hyperheuristics are still MHO algorithms
  - they may potentially provide a good adaptation to the problem domain
  - hyperheuristics are computationally very demanding
  - hyperheuristics remain restricted to applications with low-cost fitness functions (and competitions)
- Avoid "Frankenstein methods", i.e. overly intricate methods with many different operators (Michalewicz and Fogel, 2004), where the contribution of each component is hard to evaluate