Parameter Setting using Hyper-heuristics in Uncertain Environments

Ender Özcan, A. Şima Etaner-Uyar,
Berna Kiraz, Gönül Uludag, Muhammet Köle

Parallel Problem Solving from Nature - 2012

Outline

Outline

Sources of Uncertainty[1]

- Time-varying fitness functions (dynamic environments)
- Noise in the environment
- Tolerance/Range of values for the design variables and/or environmental parameters
- Fitness approximation

[1] Y. Jin, J. Branke, "Evolutionary optimization in uncertain environments-a survey", IEEE Trans. on Evolutionary Computation, 9:3, pp. 303-317, 2005.

Definition of Change

- dynamic ≡ changing ≡ stochastic environment
- fitness of an individual evaluated in different environments may be different

Definition

A change in the environment may mean:

- the objectives change
- the constraints change
- the problem instance changes

Properties of Change

- severity of changes
- frequency of changes
- predictability of changes
- cycle length / cycle accuracy

Motivation

- different approaches perform better with different types of changes
- may not be possible to know the properties of the change beforehand
- properties of the change may itself change over time
- better to adapt the approach based on the current properties of the changes
- hyper-heuristics (HH) adaptive in nature

Focus

- Case 1: parameter setting for function optimization in dynamic environments
- Case 2: parameter setting for a race car optimization (dynamic + noisy environment)
- Case 3: parameter setting for a statistical model in dynamic environments

Hyper-heuristics

- Perform search over the space of heuristics
- Two hyper-heuristic classes
  - Heuristics to choose heuristics
  - Heuristics to generate heuristics

A Selection Hyper-heuristic Framework
Single-point Search

Algorithm 1 A Selection Hyper-heuristic Framework

generate initial candidate solution p
while (termination criteria not satisfied) **do**
  select a heuristic h from $H_1, \ldots, H_n$
  generate a new solution $s = h(p)$ by applying h to p
  decide whether to accept s or not
  if (s is accepted) **then**
    $p = s$
  end if
end while

Heuristic Selection Mechanisms

- Simple Random (SR)
- Random Descent (RD)
- Random Permutation (RP)
- Random Permutation Descent (RPD)
- Greedy Selection (GR)
- Choice Function (CF)
- Reinforcement Learning (RL)

Move Acceptance Methods

- All Moves (AM)
- Only Improving (OI)
- Improving and Equal (IE)
- Great Deluge (GD)
- Monte Carlo based (EMCQ)
- Simulated annealing based (SA and SA+RH)

Outline

Scope of Study[2]

- thirty-five single point search based selection hyper-heuristics
  - 5 heuristic selection methods (SR, GR, CF, RL, RPD)
  - 7 move acceptance methods (AM, OI, IE, EMCQ, GD, SA, SA+RH)
- continuous dynamic environments
- various change dynamics

[2] Berna Kiraz, A. Şima Uyar, Ender Özcan, "An Investigation of Selection Hyper-heuristics in Dynamic Environments", EvoApplications 2011,Part I, LNCS vol. 6624, pp. 314-323, Springer, 2011.

The Approach

- candidate solutions: real-valued vectors representing the coordinates of a point in the multidimensional search space
- parameterized Gaussian mutation (seven different standard deviations: 0.5, 2, 7, 15, 20, 25, 30) used as low-level heuristics

Experimental Design

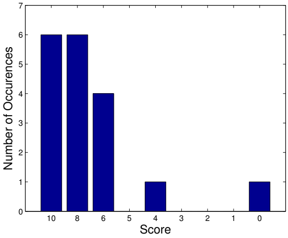
- Moving Peaks Benchmark (MPB) generator
- Performance Evaluation - Offline Error
- Change Dynamics
  - Change frequencies τ
  - Change severities ρ
- Performance comparison experiments
- Scalability

Results

Table: The overall Formula 1 based ranking scores for the top five approaches.

Approach	Overall Score
Choice Function–Improving and Equal	136
Choice Function–Only Improving	119
Choice Function–EMCQ	86
Random Permutation Descent–Only Improving	72
Random Permutation Descent–Improving and Equal	71

Histogram of scores for CF–IE over 18 dynamic environment cases



Discussion

- learning selection hyper-heuristics better
- acceptance criteria relying on some algorithmic parameter settings (e.g. SA) are not good
- AM is the worst strategy

Outline

Scope of Study[3]

- Explore the application of selection hyper-heuristics
- TORCS based Car Setup Optimisation Problem
- Hyper-heuristic study in a noisy environment

[3] Muhammet Köle, A. Şima Etaner-Uyar, Berna Kiraz, Ender Özcan, "Heuristics for Car Setup Optimisation in TORCS",UKCI 2012, the 12th Annual Workshop on Computational Intelligence, 2012.

Car setup optimisation problem

- TORCS
- Find an optimal setting for the parameters of a racing car
- A client-server architecture
- 22 real-valued parameters normalized in the interval $[0, 1]$
- Fuel usage can be used in the evaluations

Experimental Design

- A solution is represented as a real-valued vector of length 22
- Low-level heuristics
  - Gaussian mutation with $\sigma = \{0.01, 0.1, 0.3, 0.5, 0.75, 0.9, 1.0, 2.0\}$
  - H1 (0.01) to H8 (2.0)
- The total distance raced is used as fitness measure
- Heuristic Selection Methods
- Move Acceptance Methods

Results - Comparison of individual low-level heuristics with HH

Algorithm	*C-Speedway*		*CG-Track 2*		*E-Track 3*		*Dirt 6*	
	F_{on}	F_{off}	F_{on}	F_{off}	F_{on}	F_{off}	F_{on}	F_{off}
H1	3959.1916	3929.4184	3073.6408	2892.0696	2695.4984	2784.1938	2081.7148	2075.5434
H2	5363.7588	5437.5456	4042.6820	4047.4520	3529.3040	3608.8092	2471.1112	2448.5214
H3	5513.5218	5632.2624	**4246.4274**	4232.8706	3735.1758	3738.0818	2547.6020	2564.1836
H4	5588.3220	**5663.3742**	4208.1258	4155.1020	3736.1888	3732.7214	2553.6414	2538.0804
H5	5628.6702	5532.7586	4233.1604	**4233.1604**	3768.7794	3730.9820	2554.6182	**2584.7278**
H6	5594.7784	5637.9634	4198.4516	4219.9134	3782.5716	**3781.9954**	2544.9120	2551.3200
H7	**5667.5886**	5569.2184	4210.7988	4194.9276	3737.5020	3719.8992	**2561.9918**	2554.7114
H8	5591.9374	5537.2516	4201.9772	4213.2198	**3791.9290**	3769.9948	2557.9762	2539.8038
HH	5625.8342	5449.7374	4116.2248	4178.3028	3718.6228	3763.9754	2519.5432	2503.6368

Discussion

- SR-IE delivers a very good performance
- A random mutation hill climbing framework (H5) delivers a similar performance when compared to SR-IE

Outline

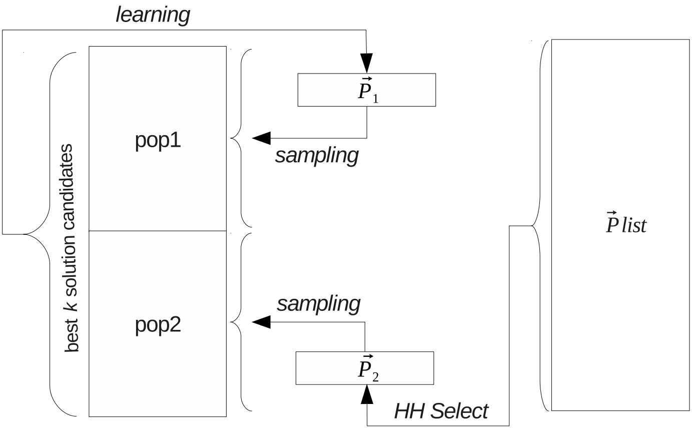
Scope of HH-PBIL2 Study[4],[5]

- A hybridisation of EDAs with hyper-heuristics in the form of a multi-phase framework
- Analyze and explain the behavior of such an EDA and hyper-heuristic hybrid
- Determine a heuristic selection method

[4] Gönül Uludag, Berna Kiraz, A. Şima Etaner-Uyar, Ender Özcan, "A Framework to Hybridize PBIL and a Hyper-heuristic for Dynamic Environments", PPSN 2012: 12th International Conference on Parallel Problem Solving from Nature, LNCS vol. 7492, pp. 358-367, Springer, 2012.

[5] Gönül Uludag, Berna Kiraz, A. Şima Etaner-Uyar, Ender Özcan, "Heuristic Selection in a Multi-phase Hybrid Approach for Dynamic Environments",UKCI 2012, the 12th Annual Workshop on Computational Intelligence, 2012.

The Framework of HH-PBIL2



Experimental Design

- XOR generator
- Decomposable Unitation-Based Functions (DUFs)
- Offline learning stage (SPBIL)
- Performance Evaluation - Offline Error
- Two types of dynamic environments
  - Randomly changing environments
  - Cyclic environments
- Change Dynamics
  - Change frequencies τ
  - Change severities ρ
  - Cycle lengths CL

Results - Overall

Table: Overall ($s+$, $s-$, $\geq$ and $\leq$) counts for the different hyper-heuristics incorporating different heuristic selection schemes.

Algorithms	$s+$	$s-$	$\geq$	$\leq$
RP	113	20	65	54
RPD	78	29	82	63
SR	57	55	59	81
RD	30	81	69	72
RL	18	111	59	64

Random Permutation is statistically significantly better

Discussion

- The approach is not very sensitive to the number of low-level heuristics and the mutation and learning rate
- The best performance is achieved when the change period is greater than or equal to the number of low-level heuristics.
- Restart schemes useful
- The method generates a very good performance particularly in cyclic environments

Outline

Conclusion

- HH successful in dynamic environments as a parameter setting approach
- HH approaches with learning successful
- HH approaches including a hillclimbing component successful
- Move acceptance is as important as heuristic selection
- noise is a problem!

Future Work

- new learning hyper-heuristics
- noise handling
- hybridisations ?
- real world problems (e.g. dynamic vehicle routing, dynamic community detection ...)