Master's Thesis

From Parameter Tuning to Dynamic Heuristic Selection

submitted by

Yevhenii Semendiak

born 07.02.1995 in Izyaslav

Technische Universität Dresden

Fakultät Informatik
Institut für Software- und Multimediatechnik
Lehrstuhl Softwaretechnologie

Supervisor: MSc. Dmytro Pukhkaiev
Dr.-Ing. Sebastian Götz
Professor: Prof. Dr. rer. nat. habil. Uwe Aßmann

Submitted February 23, 2020

Technische Universität Dresden/Fakultät Informatik

Aufgabenstellung für die Masterarbeit

Name, Vorname: Semendiak, Yevhenii

Studiengang: Master DSE Matr. Nr.: 4733680

Thema:

From Parameter Tuning to Dynamic Heuristic Selection

Zielstellung :

Metaheuristic-based solvers are widely used in solving combinatorial optimization problems. A choice of an underlying metaheuristic is crucial to achieve high quality of the solution and performance. A combination of several metaheuristics in a single hybrid heuristic proved to be a successful design decision. State-of-the-art hybridization approaches consider it as a design time problem, whilst leaving a choice of an optimal heuristics combination and its parameter settings to parameter tuning approaches. The goal of this thesis is to extend a software product line for parameter tuning with dynamic heuristic selection; thus, allowing to adapt heuristics at runtime. The research objective is to investigate whether dynamic selection of an optimization heuristic can positively effect performance and scalability of a metaheuristic-based solver.

For this thesis, the following tasks have to be fulfilled:

- Literature analysis covering closely related work.
- Development of a strategy for online heuristic selection.
- Implementation of the developed strategy.
- Evaluation of the developed approach based on a synthetic benchmark.
- (Optional) Evaluation of the developed approach with a problem of software variant selection and hardware resource allocation.

Betreuer: M.Sc. Dmytro Pukhkaiev, Dr.-Ing. Sebastian Götz

Verantwortlicher Hochschullehrer: Prof. Dr. rer. nat. habil. Uwe Aßmann

Institut: Software- und Multimediatechnik

Beginn am : 01.10.2019

Einzureichen am : 09.03.2020

Unterschrift des verantwortlichen Hochschullehrers

Verteiler: 1 x SCIS, 1x HSL, 1x Betreuer, 1x Student

Statement of Authorship

I hereby certify that I have authored this Master Thesis entitled “From Parameter Tuning to Dynamic Heuristic Selection“ independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

Dresden, February 23, 2020

Yevhenii Semendiak

Yevhenii: should I add a disclaimer like this?

Contents

0.1 abstract

Abstract will be available in final versions of thesis.

1 Introduction

Intent and content of chapter. This chapter is an self-descriptive, shorten version of thesis.

1.1 Motivation

Structure:

- optimization problem(OP) → exact or approximate (+description to both) → motivation to use **approximate solvers** →
- impact of parameters, their tuning on solvers → motivation of **parameter control** (for on-line solver) →
- but what if we want to solve a class of problems (CoP) → algorithms performance is different →
- user could not determine it [1] → exploration-exploitation balance
- no-free-lunch (NFL) theorem [2] → motivation of the thesis

thesis motivation The most related research field is Hyper-heuristics optimizations [3], that are designed to intelligently choose the right low lewel heuristics (LLH) while solving the problem. But the weak side of hyper-heuristics is the luck of parameter tuning of those LLHs [links]. In the other hand, meta-heuristics often utilize parameter control approaches [links], but they do not select among underlying LLHs. The goal of this thesis is to get the best of both worlds - algorithm selection from the hyper-heuristics and parameter control from the meta-heuristics.

1.2 Research objective

The following steps should be completed in order to reach the desired goal:

Analysis of existing studies of algorithm selection. *(find a problem definition, maybe this will do [1])*

Analysis of existing studies in field of parameter control and algorithm configuration problems *(find a problem definition)* [4]

Formulation and development of combined approach for LLH selection and parameter control.

Evaluation of the developed approach with

Yevhenii: family of problems??? since it is a HH, maybe we should think about it...

.

Research Questions At this point we define a Research Questions (RQ) of the Master thesis.

- **RQ 1** Is it possible to select an algorithm and it hyper-parameters while solving an optimization problem *on-line*?
- **RQ 2** What is the gain of selecting and tuning algorithm while solving an optimization problem?
- **RQ 3?** How to solve the problem of algorithm selection and configuration simultaneously?

1.3 Solution overview

- described problems solved by HH, highlight problems of existing HHs(off-line, solving a set of homogeneous problems in parallel)
- create / find portfolio of MHs (Low level Heuristics)
- define a search space as combination of LLH and their hyper-parameters (highlight as a contribution)
- solve a problem on-line selecting LLH and tuning hyper-parameters on the fly. (highlight as a contribution? need to analyze it.)

Thesis structure The description of this thesis is organized as follows. First, in chapter **??** we refresh readers background knowledge in the field of problem solving and heuristics. In this chapter we also define the scope of thesis. Afterwards, in chapter **??** we describe the related work and existing systems in defined scope. In Chapter 4 one will find the concept description of dynamic heuristics selection. Chapter 5 contains more detailed information about approach implementation and embedding it to BRISE. The evaluation results and analysis could be found in Chapter 6. Finally, Chapter 7 concludes the thesis and Chapter 8 describe the future work.

2 Background and related work

The structure is the same the beginning of introduction, but way more detailed.

2.1 Optimization problems and solvers

2.1.1 Definition of optimization problems

Yevhenii: Need to find a classification of optimization problems.

2.1.2 Optimization problems solvers

Some (but not only) literature: [5]

Exact solvers

Approximate solvers

Comparison:

Pros and cons of both [6]

2.2 Approximate solvers of optimization problems

TSP as the running example. I guess, I will introduce it as an example of pertrubation problems in previous section.2.1

2.2.1 Heurustics

Definition

examples

Conclution

Heuristics are strictly problem dependent and each time require adaptations.

2.2.2 Meta-heuristics

Definition

classification

examples

We distinguish following examples among all existing meta -heuristics, since later we use them as the LLH in developed hyper -heuristic.

GA

SA

ES

No-free-lunch theorem

NFL is the problem of heuristics[2]

Exploration-explotation balance

Conclution

Proper assignment of hyper -parameters has great impact on exploration-explotation balance and those on (meta) -heuristic performance.

2.2.3 Hybrid -heuristics

definition

examples

Guided Loca Search (GLS) + Fast Local Search [7]

Direct Global + Local search [8]

Simulated Annealing + Local Search [9]

2.2.4 Conclution

2.2.5 Hyper -heuristics

definition

Classification

Search space: heuristic selection, heuristic generation

Learning time: on-line learning hyper-heuristics, off-line learning hyper-heuristics, no-learning hyper-heuristics

other classification characteristics from [1], [10], mb smth else. For instance, hyperparameter tuning

examples

[11] (Online algorithm selection at page 27); [1]

conclution

they usually (need to check it) have lack of parameter control

2.2.6 Conclution on approximate solvers

Pros and cons of heuristics - too problem dependent

Pros and cons of meta -heuristics - no LLH selection, strict to one problem

Pros and cons of hybrid -heuristics - no LLH selection, strict to one problem ?

Pros and cons of hyper -heuristics - no parameter control?

2.3 Parameter tuning

Yevhenii: The goal of section: analysis of existing systems for hyper -parameter optimization (tuning), weaknesses and strength of each of the system

Yevhenii: Should I include the analysis from the code-basis point of view? If no, I do not see what should I conclude from this section exect ”there are numer of parameter tuning systems each of them has pros and cons..

2.3.1 problem definition, approaches (grid, random, mh..)

2.3.2 Systems for parameter tuning

IRACE

approach [12]

pros and cons

SMAC

approach description

BOHB

approach description

AUTO-SKLEARN

CASH (Combined Algorithm Selection and Hyperparameter optimization) problem

pros and cons (on-line or off-line, problems to solve, extensibility) [13]

BRISEv2

approach description

Yevhenii: Other systems?

conclution

depends on the resolved question in todo...

2.4 Parameter control

definition, approaches

system examples

conclution

impact of parameter control based on other's evaluation

2.5 Conclution

The meta-heuristic systems designers reported positive impact of parameter control embedding. However, as the outcome of the no-free-lunch theorem, those systems can not tolerate broad range of problems, for instance, problem classes. In other hand, hyper -heuristics are designed with an aim to select the low level heuristics and those propose a possible solution of problem, stated in no-free-lunch theorem, but the lack of parameter control could dramatically decrease the performance of LLH (probably, I need to find a prove of this, or rephrase).

Scope of work defined. In this thesis we try to achieve the best of both worlds applying the best fitting LLH and tuning it's parameters while solving the problem on-line.

3 Concept description

Reviewer: It is your concept section, do not put related work here. How do you solve a defined problem?

3.1 Considered approaches

Approaches to derive an information about configurations:

- ”SPIRE” or heuristics (IRACE approach, all possible combinations are defined beforehand)
- switching (from one to another, which switch is better)
- picking (just, ”run this particular heuristic now”)

3.2 What is the problem with the Search Space?

Yevhenii: unofficial style? (not much, but at least to make some statements / headers catchy)

Reviewer: why is search space important? it just appears without an explanation

- tree-shaped search space
- heterogeneous data
- different algorithms (model) to traverse and optimize such a search space could require different information
– generality
- top-down approach of optimization
– different views of same Configuration (level-dependent) - filtering, transformation
– problem features to consider while selecting meta-heuristic [1] page 6

Reviewer: more sections?

4 Implementation

4.1 Analysis of required changes in system

4.1.1 Description of BRISE v2.2.0

Data preprocessing

Model compatibilities

Search space compatibilities

4.1.2 Meta-heuristics repository

-table and short comparison of checked repositories

- Solid
- mlrose
- jMetalPy
- or-tools
- pyTSP
- LocalSolver

jMetalPy in more details

Available Meta-heuristics

Required features - any-time (or predictable) termination
- warm-startup

opened PR

4.1.3 Development planning

use-case definition

generalization with team

requirements engineering - interfaces

class diagrams

data flow

5 Evaluation

5.1 Problem

which problem I wanna solve with hyper-heuristic

- TSP
- n-Queens
- knapsack

Reviewer: what will you compare?

6 Conclusion

Reviewer: answer research questions

7 Future work

- more sophisticated models?
- dependencies / constraints in search space
- wider evaluation?

Reviewer: consider merging with conclusion, if too short

Yevhenii: is it correct style?

Reviewer: better the one with first letters of family names and year

Bibliography

[1] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, “Automated algorithm selection: Survey and perspectives,” *Evolutionary computation*, vol. 27, no. 1, pp. 3–45, 2019.

[2] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[3] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, “Hyper-heuristics: An emerging direction in modern search technology,” in *Handbook of metaheuristics*. Springer, 2003, pp. 457–474.

[4] N. Lavesson and P. Davidsson, “Quantifying the impact of learning algorithm parameter tuning,” in *AAAI*, vol. 6, 2006, pp. 395–400.

[5] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[6] J. Hromkovič, *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*. Springer Science & Business Media, 2013.

[7] E. Tsang and C. Voudouris, “Fast local search and guided local search and their application to british telecom’s workforce scheduling problem,” *Operations Research Letters*, vol. 20, no. 3, pp. 119–127, 1997.

[8] M. Syrjakow and H. Szczerbicka, “Efficient parameter optimization based on combination of direct global and local search methods,” in *Evolutionary Algorithms*. Springer, 1999, pp. 227–249.

[9] O. C. Martin and S. W. Otto, “Combining simulated annealing with local search heuristics,” *Annals of Operations Research*, vol. 63, no. 1, pp. 57–75, 1996.

[10] E. K. Burke, M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, “A classification of hyper-heuristic approaches: revisited,” in *Handbook of Metaheuristics*. Springer, 2019, pp. 453–477.

[11] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, “Recent advances in selection hyper-heuristics,” *European Journal of Operational Research*, 2019.

[12] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, “The irace package: Iterated racing for automatic algorithm configuration,” *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[13] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, “Efficient and robust automated machine learning,” in *Advances in neural information processing systems*, 2015, pp. 2962–2970.

A Appendix

A.1 Additional Information

A.2 More Important Information

Confirmation

I confirm that I independently prepared the thesis and that I used only the references and auxiliary means indicated in the thesis.

Dresden, February 23, 2020