Tómas Philip Rúnarsson

Data Driven Design of Fast Approximate Algorithms

The Icelandic Research Fund 2014
Project Grant - New proposal
Appendix A (Detailed project description)

Reykjavík, 3/6/2013

a.) State of the art and proficiency

Designing algorithms to solve problems approximately, for tasks that cannot be solved exactly, is a time consuming trial and error process requiring problem specific insights. The alternative to hand-crafting heuristics, is a data driven approach to automate, in part, this design process. Typically this data consists of numerous problem instances and when possible their exact solution. The first successful application of this idea was presented by Minton back in 1996 [13]. There the idea was to automatically design problem specific versions of constraint satisfaction algorithms. Using data generated from different instance distributions programs were designed that were on par with hand-crafted programs.

Depending on the underlying data distribution of problem instances, different heuristic perform differently. This is due to the fact that any algorithm which has superior performance in one class of problems is inevitably inferior over another class, c.f. the *no free lunch* theorems [22]. The success of a heuristic is how it manages to deal with and manipulate the characteristics of its given problem instance. An attempt to understand the heuristic's performance is to look at the relationship between problem structure and heuristic effectiveness. Attempt to classify heuristic performance over instance space have been termed "landmarking" or "footprints" in instance space [6, 16]. These studies involve clustering problem instances using problem specific features. As an example, a recent study shows that a linear combination of features for bin packing problems are related to heuristic performance [11]. In this study principle component analysis is used to perform the cluster analysis. In the scope of scheduling this has been explored by the authors for jobshop scheduling by [8] and timetable scheduling [19, 20]. Clustering with self-organizing maps indicates that real-world timetable problems can have a very different underlying parameter distribution that synthetically generated instances [19]. This has also been observed in scheduling [21].

Exact solutions can in some cases demonstrate how an optimal solution to a problem instance may be constructed. Building solutions, however, requires problem specific assignments which must be predefined for any prob-

lem type. For example, in bin packing this would be to assign an item to a bin and similarly for scheduling assigning a job to a machine. Typically, one would design assignments that guarantee the construction of legal solutions. For example, we would not assign an item to a full bin, or a job to a busy machine. Improvement or local search operators would also require the design of problem specific "assignments". Deciding on an assignment requires a policy and it is this decision policy we aim to design and learn. In some cases the number of possible assignments is very large or a complex procedure is needed to create legal assignments. The common practice then is to design a number of problem specific procedures or heuristic for this purpose. An example of such a heuristic for the packing of a large number of items would be to assign the smallest one to a particular bin. The policy for selecting among these different assignment heuristics is now the learning problem. The most common approach to representing policies is through utility functions and simple one step look-ahead procedures. In this case all possible assignments or heuristics used for this purpose are performed and the partial solutions evaluated by the utility function. The assignment resulting in the highest utility is then the one chosen. Using genetic programming this approach has been quite successful for bin-packing problems [17, 5]. In some sense we can think of this as a classification problem where we must determine which partial solution should be selected. From this perspective the utility function corresponds to a classification function. The alternative to performing a look-ahead would be to associate the current partial solution to a particular assignment or assignment heuristic. This would seem to be a harder problem to solve, but a few examples of this approach exist for the design of heuristics. For example, in [7, 1] a genetic program, with a tree like structure, inputs properties of the current partial solution and all leaf nodes correspond to some predefined assignment heuristic.

The idea of using algorithms to automatically design or tune search algorithms using hyper-heuristics was prosed in [2]. The hyper-heuristic framework presented operates at a high level of abstraction and often has no knowledge of the domain. Typically an evolutionary algorithm will be applied to search the policy space directly. In this case the exact solution to generated

problem instances need not be know in advance. The evolutionary algorithm simply searches for policies that perform on average the best over the set of training problem instances.

In [10] data is generated from a known heuristic, for job-shop scheduling, and a decision tree used to rediscover the heuristic from the data. However, such a technique is unable to outperform the heuristic that generated the training data used. This drawback was confronted in [12, 18, 15] by using data generated from an optimal scheduler, computed off-line. Preferring simple to complex models, the resulting dispatching rules gave significantly better schedules than using popular heuristics in that field, and a lower worst-case factor from optimality. A similar approach is taken for timetable scheduling in [3], using case based reasoning. Training data is guided by the two best heuristics for timetable scheduling. The authors point out that in order for their framework to be successful, problem features need to be sufficiently explanatory and training data need to be selected carefully so they can suggest the appropriate solution for a specific range of new cases.

Using the hype-heuristic framework HyFlex [14].

A good overview on hyper-heuristics may be found in [4].

the reinforcement learning approach, Zhang and Runarsson [9] point out that meta learning can be very fruitful in reinforcement learning, and in their experiments they discovered some key discriminants between competing algorithms for their particular problem instances, which provided them with a hybrid algorithm which combines the strengths of the algorithms.

Monte Carlo approach, Runarsson x2

Missing items:

- Surrogate models.
- Handling large volumed of training data.
- ... other ...

b.) Objectives of the project and originality

Things not yet done in the literature:

- Optimizing for speed and accuracy, dual-objective problem.
- Design of experiment: how to sample of training data to achieve the goal of automatically generating algorithms, often just generated in an ad-hoc manner.
-

How can you speed up an algorithm? What can the data be used for within the design procedure? Data can be used for modelling objectives, but also for the design og heuristics to create solutions, and?

The broad objective of this research programme is to deepen our understanding of how data may be applied to the design of fast and approximate algorithms.

To achieve this goal we have put together a research team with expertise in statistics, computing and computational intelligence. There are in principle three main aspects which we For this we believe a diverse set of applications will help tease out common ... in ...

1. **Data generation**: to understand the properties and characteristics of the data needed to design efficient and effective algorithms. We will try to achieve this objective by looking at specific sub-goals. They are:
   (a) **Problem instance generator**: developing numerous new problem instances from a few examples, understand to what extend one can expect the algorithm designed can generalize beyond these instance distributions. When many examples exist, as is the case in our real world problem tackled by this project, how and which instances are sufficient to design our algorithms.
   (b) **Sampling**: within each instance a large amount of data may be generated. The data may be generated from optimal as well as suboptimal solutions to the instance. sub-sampling this space is also an issue. and data sampling within a problem instance.

2. **Parallel computing framework**: Develop a generic framework for the development of data driven designed algorithms. Implement numerous machine learning for large data sets on a parallel framework.

3. **Applications**: a diverse set of applications will be applied using the parallel framework, they can be put into two different categories:

   (a) Designing heuristics:

   (b) Approximating objectives:

   In each case their one data driven design will be based on read world data, whereas the second will be based on theoretically derived data.

c.) Methodology, work plan and timescale

At the University of Iceland the following personnel will be working on the project: .... , Communication with external collaborators will be conducted through short visits, meetings at conferences, e-mail and phone.

Work-Package 1 – Instance generation and data sampling

Responsibility: T. P. Rúnarsson, B. Hrafnkelsson, and PhD student.
Time-frame: 01.01.14–31.12.14.
Milestones: Framework for generating problem instances and sampling training data.

Work-Package 2 – Computational framework

Responsibility: T. P. Rúnarsson, Morris Riedel and PhD student.
Time-frame: 01.01.14–31.12.14.
Milestones: High performance computing framework for data driven design of algorithms.

Work-Package 3 – Fast approximate algorithms for streaming computing (STC)

Responsibility: P. Melsted, MSc student
Time-frame: 01.01.14–31.12.14.
Milestones: .

Missing description

Work-Package 4 – Two dimensional free form bin packing

Responsibility: H. Ingimundardttir, MSc student
Time-frame: 01.01.14–31.12.14.
Milestones: .

Valka ehf. is currently working on intelligent portioning solutions for fish. Current solutions are constrained to either fixed weights or fixed shape, and the machinery is generally inspecting only one scheme at a time. However the diversity of fish within a catch can be quite varying (both respect to weight, size, and quality) and thus it would be appropriate to have a self-adapting algorithm that implements the best cutting pattern per fish fillet based on its X-Ray imagery. Moreover, the algorithm should be able to report whether gaping (. los) is present for quality control purposes.

It is possible to interpret the fish portioning as two-dimensional free-form bin packing problem (2D-FBP), where approximately rectangular portions are being cut from irregular shape, e.g. fish fillet. Unfortunately, 2D-FBP is generally put forth as ir/regular shapes cut from regular shapes (not vice versa). Portioning combines several combinations of traditional bin-packing, namely,

- trim-loss: determination of cutting pattern to minimize waste
- assortment: determination of ?best? fillet for cutting predefined portions in order to minimise number of fillets needed to fulfil an order
- knapsack: determination of ?best? portion for cutting, since each portion has a value based on market demand making the portioning prob-

lem a multi-objective optimization problem subject to some technical constraints, such as,

- material properties: tail portions can only be taken from tail region, etc.
- cutting processes: a minimum distance between portions is required or order not to damage to cutters
- serious time constraints: the algorithm has to report the cutting paths before the fillet is situated near the water cutters

The project can be divided into the following subtasks:

- Literature study and establishment of the state of the art involving cutting stock problems of irregular shapes, especially feature selection.
- Mathematical description of methods/models of fast approximate optimisation for 2D-FBP, involving implementation in e.g. C++ or similar software.
- Comparison of method on real-world-data, with respect to accuracy and speed.
- Case study: Implementation for a X-Ray guided cutting machine designed by Valka ehf., RapidPinbone, which is being developed with HB Grandi. An extensive data set is available for two problem distributions, namely Atlantic red fish (. karfi) and cod (. orskur) suitable for data-driven learning approach
- Dissemination in the form of a M.Sc. thesis and a journal paper.

Work-Package 5 – Fast computation of thermodynamic and transport properties of fluids in relation to geothermal reservoir modeling

Responsibility: Halldr Plsson, Matthildur Mara Gumundsdttir, MSc Student

Time-frame: 01.01.14–31.12.14.
Milestones: .

Complex three dimensional models of energy and fluid flow often require accurate representations of fluid properties, especially if model simulations are performed with substantial changes in temperature and pressure. Additionally, these properties must be evaluated for different phases and in regions close to the critical point, where variations of properties can be great. Typical properties involved in computational fluid dynamics and heat transfer are pressure, temperature, density, enthalpy, entropy, viscosity and thermal conductivity.

Currently, several computer implementations exists for property calculations, mostly focused on the properties of water (in both liquid and vapor phase). A well known implementation is the REFPROP package, available from National Institute of Standards and Technology in USA, where a large number of fluids has been included in a comprehensive model. These fluids are usually based on the best available data, such as from the IAPWS (International Association for the Properties of Water and Steam) and evaluations of properties are implemented in several Fortran programs. Even though the REFPROP implementation is accurate, it requires an evaluation of long series of sometimes complex functions as well as root finding of nonlinear equations. This makes the use of REFPROP in CFD models problematic since millions of evaluations are often required, and as a consequence the property evaluation becomes a serious bottleneck in the computations.

The purpose of this work package is to examine possibilities of fast approximate calculation of fluid properties in general. This involves using table lookups and simple function evaluations of low degree polynomials (splines) or computational kernels that require minimal calculation effort. The REFPROP database will be used as a reference to produce accurate data for comparison. The product of the projects is a framework for generating fast algorithms or programs that can calculate thermodynamic and transport properties of chose fluids, as well as their derivatives, with user selectable accuracy. The algorithms should be automatically generated as C, C++ or Fortran subroutines that can be linked to CFD codes.

The project can be divided into the following subtasks:

- Literature study and establishment of the state of the art, involving possible contact to H.J. Kretzschmar who is currently working on a similar project involving water and steam.
- Mathematical description of methods/models for data interpolation, involving implementation in e.g. Matlab or similar software.
- Data fitting of models to REFPROP data, evaluation of errors.
- Comparison of methods, with respect to accuracy and speed.
- Design of scripts for automatic code generation.
- Case study: Implementation into a geothermal reservoir model.
- Dissemination in the form of a M.Sc. thesis and a journal paper.

Work-Package 6 – Fast approximate evaluation for PARAMIN

Responsibility: Gunnar Stefnsson and Gumundur Einarsson MSc student in statistics
Time-frame: 01.01.14–31.12.14.
Milestones: .

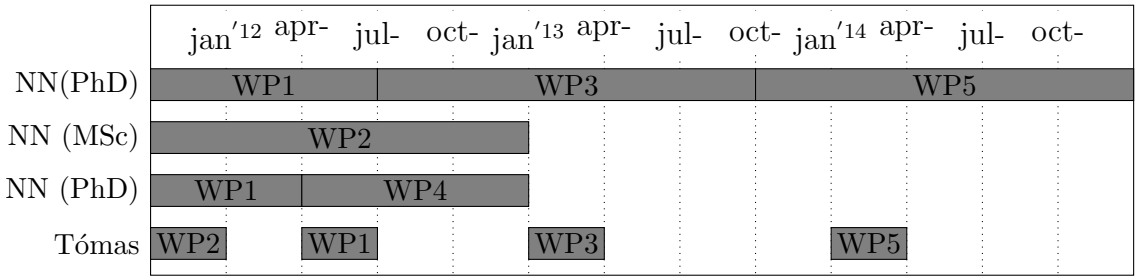

Figure 1: The six work-packages (WP) illustrated for the project schedule over the 3 years.

d. Co-operation (domestic/foreign)

e. Contribution of doctoral and master's degree students

The contribution of the doctoral and masters student to the projects has been listed in the work-packages in section c. and shown in figure 1.

f. Proposed deliverables and impact

The general aim of this research programme is to deepen our understanding of the ... customizing for problem specific applications is both of considerable scientific and commercial value. The research is expected to produce deliverables listed in the work-packages in section c. In summary:

1. 4 MSc theses with the University of Iceland.
2. 2 PhD thesis from the University of Iceland (one is already under way),
3. ??? journal and conference papers.
4. Various open source software packages will be developed.

g. Proposed publication of results

The project will produce two PhD thesis and four masters thesis which will be available to the public. The research papers will be published in peer-reviewed well-established international conferences and journals. We aim to publish with the Journal of Heuristics and IEEE Transaction of Evolutionary Computation, but other Operations Research Journals and Machine Learning Journals will also be targeted. The conferences where we aim to present our results are: PPSN, LION and Computational Intelligence conferences such as GECCO, CEC and WCCI. ???

References

[1] Mohamed Bader-El-Den, Riccardo Poli, and Shaheen Fatima. Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework. *Memetic Computing*, 1(3):205–219, 2009.

[2] Edmund Burke, Graham Kendall, Jim Newall, Emma Hart, Peter Ross, and Sonia Schulenburg. Hyper-heuristics: An emerging direction in modern search technology. *International series in operations research and management science*, pages 457–474, 2003.

[3] Edmund Burke, Sanja Petrovic, and Rong Qu. Case-based heuristic selection for timetabling problems. *Journal of Scheduling*, 9:115–132, 2006.

[4] Edmund K Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and Rong Qu. Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society (to appear)*, 2010.

[5] Edmund K Burke, Matthew R Hyde, Graham Kendall, and John Woodward. Automating the packing heuristic design process with genetic programming. *Evolutionary computation*, 20(1):63–89, 2012.

[6] David Corne and Alan Reynolds. Optimisation and generalisation: Footprints in instance space. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Gnter Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pages 22–31. Springer Berlin, Heidelberg, 2010.

[7] Alex S Fukunaga. Automated discovery of local search heuristics for satisfiability testing. *Evolutionary Computation*, 16(1):31–61, 2008.

[8] Helga Ingimundardottir and Thomas Philip Runarsson. Determining the Characteristic of Difficult Job Shop Scheduling Instances for a Heuristic Solution Method. In Marc Schoenauer, editor, *Learning and Intelligent Optimization, 6th International Conference, LION 6*, Paris, 2012. Springer Lecture Notes in Computer Science.

[9] Shivaram Kalyanakrishnan and Peter Stone. Characterizing reinforcement learning methods through parameterized learning problems. *Machine Learning*, 84(1-2):205–247, June 2011.

[10] Xiaonan Li and Sigurdur Olafsson. Discovering dispatching rules using data mining. *Journal of Scheduling*, 8:515–527, 2005.

[11] Eunice López-Camacho, Hugo Terashima-Marín, Gabriela Ochoa, and Santiago Enrique Conant-Pablos. Understanding the structure of bin packing problems through principal component analysis. *International Journal of Production Economics*, (in press):–, 2013.

[12] Abid M. Malik, Tyrel Russell, Michael Chase, and Peter Beek. Learning heuristics for basic block instruction scheduling. *Journal of Heuristics*, 14(6):549–569, December 2008.

[13] Steven Minton. Automatically configuring constraint satisfaction programs: A case study. *Constraints*, 1(1-2):7–43, 1996.

[14] Gabriela Ochoa, Matthew Hyde, Tim Curtois, Jose A Vazquez-Rodriguez, James Walker, Michel Gendreau, Graham Kendall, Barry McCollum, Andrew J Parkes, Sanja Petrovic, et al. Hyflex: a benchmark framework for cross-domain heuristic search. In *Evolutionary Computation in Combinatorial Optimization*, pages 136–147. Springer, 2012.

[15] Sigurdur Olafsson and Xiaonan Li. Learning effective new single machine dispatching rules from optimal scheduling data. *International Journal of Production Economics*, 128(1):118–126, 2010.

[16] Bernhard Pfahringer, Hilan Bensusan, and Christophe Giraud-carrier. Meta-learning by landmarking various learning algorithms. In *in Proceedings of the 17th International Conference on Machine Learning, ICML'2000*, pages 743–750. Morgan Kaufmann, 2000.

[17] Riccardo Poli, John Woodward, and Edmund K Burke. A histogram-matching approach to the evolution of bin-packing strategies. In *Evolu-*

tionary Computation, 2007. CEC 2007. IEEE Congress on, pages 3500–3507. IEEE, 2007.

[18] Tyrel Russell, Abid M. Malik, Michael Chase, and Peter van Beek. Learning heuristics for the superblock instruction scheduling problem. *IEEE Trans. on Knowl. and Data Eng.*, 21(10):1489–1502, October 2009.

[19] Kate Smith-Miles and Leo Lopes. Generalising algorithm performance in instance space: A timetabling case study. In Carlos Coello, editor, *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, pages 524–538. Springer Berlin, Heidelberg, 2011.

[20] Kate Smith-Miles and Thomas T. Tan. Measuring algorithm footprints in instance space. *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012.

[21] Jean-Paul Watson, Laura Barbulescu, L. Darrell Whitley, and Adele E. Howe. Contrasting structured and random permutation flow-shop scheduling problems: Search-space topology and algorithm performance. *INFORMS Journal on Computing*, 14:98–123, 2002.

[22] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.