BABEȘ-BOLYAI UNIVERSITY

DOCTORAL THESIS SUMMARY

Optimization in complex systems with computational intelligence models

---

Author:
Nándor Bándi

Supervisor:
Noémi Gaskó, assoc. prof. PhD

2026

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Contributions 2

1.4 List of publications 4

2 Theoretical background **6**

2.1 Optimization methods 7

2.1.1 Iterative search operator classification 7

2.1.2 Challenges of optimization techniques 8

2.2 Hyper-heuristics 9

2.2.1 Motivation 9

2.2.2 Classification of hyper-heuristics 9

3 Proposed hyper-heuristic framework **11**

3.1 Nested Markov Chain Hyper-Heuristic 11

3.1.1 Proposed model 12

3.1.2 Numerical experiments 13

3.1.3 Conclusions and further work 14

3.2 Sequential probability ratio testing in hyper-heuristics 15

3.2.1 Proposed framework 15

3.2.2 Numerical experiments 15

3.2.3 Results 16

3.2.4 Conclusion 16

4 Applications **17**

4.1 Application To Hyperparameter Tuning 17

4.1.1 Problem statement 17

4.1.2 Adapted framework 18

4.1.3 Numerical experiments 19

4.1.4 Results and conclusion 20

4.2 Hyper level search hybridization. A clustering application. 21

4.2.1 Improved framework 21

4.2.2 Numerical experiments 21
4.2.3 A clustering application 22
4.2.4 Conclusion 22
4.3 Addressing the sensor network localisation problem 24
4.3.1 Numerical experiments and results 24
4.3.2 Conclusion 24

5 Conclusions **25**
5.1 Contributions and results 25

Bibliography **28**

Chapter 1

Introduction

In this thesis we present research work in the field of optimization techniques. The thesis is focused on presenting our results in the field of hyper-heuristics. We present a novel hyper-heuristic model, various improvements and some applications to well known problems in machine learning and localisation. The model evaluation methodology matches that of the literature, numerical experiments were conducted on well known benchmark problems in such a way as to ensure statistical significance and fairness. The achieved results highlight both the potential and limitations of the proposed approach. The experimental results suggest that the proposed framework is able adapt to specific problems and that the solution quality is limited by the quality of the low-level heuristics and the number of available iterations.

1.1 Motivation

The process of optimization is crucial to many fields. The reduction of problems to a mathematical formulation enables the solution of many problems. The solution quality is not limited only by the problem complexity and available computational resources but also by the effectiveness of the optimization algorithm chosen for the given problem. Choosing the appropriate approach to complex problems usually requires deep domain knowledge and understanding of the problem structure in order to reduce the search space to a feasible size. This work is motivated by the fact that any improvement of the optimization process can impact the solution quality of many problems. An improvement direction would be the automatic design of optimizers which can diminish the need for deep domain expertise. Finding the right optimization approach is often done by trial and error and using prior experience. Hyper-heuristic methods emerged as the research direction aimed to alleviate this issue. These methods search the optimization algorithm space in an attempt to improve optimization results. The work has also been motivated by the fact that hyper-heuristics can be effective even in the light of the no free lunch theorem in the special case of considering a restricted set of objective functions, as detailed in [48]. An important limitation of hyper-heuristic methods is that they can require many

objective function evaluations in order to find a well-performing heuristic configuration. This can be problematic in scenarios where function evaluations are costly, such as computationally intensive simulations. Another challenge is finding efficient search strategies in algorithm space, and the definition of this space such that it remains explorable but highly expressive.

1.2 Objectives

The research objectives that have been set out were motivated by the following factors: the field of optimization is vast, the research process can only be carried out effectively after a thorough overview and understanding of the existing literature. Another factor was the identification of research gaps in the existing literature and the selection of a relevant research direction. The most important factor was the contribution of novel research in the identified direction supported by correct evaluation methodologies and applications. The identified objectives are:

i) Identification of research gaps in the hyper-heuristics literature. Identify problem domains that have not been addressed yet by hyper-heuristic methods. Search for aspects of current hyper-heuristic frameworks that can be improved.

ii) Introduction of a novel hyper-heuristic based on the identified research gaps and improvement possibilities.

iii) The carrying out of experiments and comparisons with state-of-the-art methods. Review the hyper-heuristics evaluation methodology used in the literature and perform numerical experiments based on that methodology.

iv) Improvement of the introduced method based on the experiments. Reflect on the results and identify potential improvement opportunities.

v) Identification of problems that can benefit from applying the hyper-heuristic optimization approach. Application of the novel method to problems such as clustering, hyperparameter tuning and localization problems.

1.3 Contributions

This work contains contributions to both fundamental and applicative research in the field of hyper-heuristics. Contributions to fundamental research include the proposal of a novel hyper-heuristic framework aimed at improving the expressivity and efficiency of existing frameworks and various improvements to the heuristic design process. These contributions have been formalised independently of any problem domain which enables their general application. The proposed framework and the

improvements have been evaluated via numerical experiments and according to the evaluation methodology suggested by the hyper-heuristic literature.

- Research of novel hyper-heuristic [16, 6]
  - Novel hyper-heuristic: Nested Markov Chain Hyper-heuristic (NMHH)
  - Numerical experiments and comparison with other methods on benchmark functions
  - Analysis of the adaptability and efficiency of proposed method
- Research of effective sampling in hyper-heuristics [14]
  - Integration of sequential probability ratio testing into NMHH
  - Numerical experiments performed on extended set of benchmark functions
  - Comparison with the regular sampling approach, another hyper-heuristic and meta-heuristics

This work also contributes to applicative research in order to assess the validity of the proposed models in real problem settings and to study their limitations. The proposed framework has been applied to problems such as clustering, hyperparameter tuning and sensor network localisation.

- Applicative research on hyperparameter tuning [15]
  - Adaptation of NMHH to hyperparameter tuning of classifiers
  - Numerical experiments on well known datasets, comparison with well known tuning methods
  - Analysis of resulting efficiency frontier
- Applicative research on clustering [5]
  - Hybridization of the hyper level search
  - Application of the hybrid version to clustering problems
  - Numerical experiments on well known clustering problems and benchmark problems
  - Comparison with well known clustering methods
- Applicative research on the sensor network localisation problem [4]
  - Application of NMHH to the SNLP
  - Numerical experiments on synthetic localisation problems of varying complexity

- Comparison with well known SNLP solving methods
- Comparison with manually fine-tuned heuristic approach

These contributions are based on original research publications [16, 6, 14, 15, 5, 4].

1.4 List of publications

The publication rankings are based on the guideline of the National Council for the Recognition of University Degrees, Diplomas and Certificates (CNATDCU). The conference papers have been ranked according to the appropriate CORE ranking, and the articles by the UEFISCDI article awards list.

- **N. Bándi**, N. Gaskó. 2024. Nested Markov chain hyper-heuristic (NMHH): a hybrid hyper-heuristic framework for single-objective continuous problems. PeerJ Computer Science, volume 10, 10:e1785 doi: 10.7717/peerj-cs.1785
  Indexed by Web Of Science, ***Rank A*** *(2024 UEFISCDI article awards list),* ***8 points***

- **N. Bándi**, N. Gaskó, Hyperparameter Tuning of Classification Algorithms with a Hyper-heuristic Framework - the Nested Markov Hyper-Tuner, 37th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI 2025), Athens, Greece, November 3-5, pp. 951-957. doi: 10.1109/ICTAI66417.2025.00139
  Indexed by IEEE, ***Rank B*** *(ICORE 2026),* ***4 points***

- **N. Bándi**, N. Gaskó. A Hybrid Differential-Evolution-Based Approach to the Sensor Network Localisation Problem. Intelligent Systems Design and Applications, (ISDA 2023). Lecture Notes in Networks and Systems, vol 1048. pp. 50-61. Springer Nature Switzerland, doi: 10.1007/978-3-031-64650-8_5
  Indexed by SCOPUS, ***Rank C*** *(CORE 2023),* ***2 points***

- **N. Bándi**, N. Gaskó, Solving Continuous Optimization Problems with a New Hyperheuristic Framework. Machine Learning, Optimization, and Data Science, (LOD 2023). Lecture Notes in Computer Science, vol 14505, pp. 116-130. Springer Nature Switzerland, doi: 10.1007/978-3-031-53969-5_10 Print ISBN:978-3-031-53968-8
  Indexed by Web Of Science, ***Rank C****,* ***2 points***

- **N. Bándi**, N. Gaskó, Improved Nested Markov Chain Hyper-heuristic Framework. A Clustering Application. Artificial Intelligence Applications and Innovations, (AIAI 2025). IFIP Advances in Information and Communication Technology, vol 755, pp. 204-220. Springer Nature Switzerland, doi: 10.1007/978-3-031-96239-4_15
  Indexed by SCOPUS, ***Rank D*** *(ICORE 2026),* ***1 point***

- **N. Bándi**, Improving the Nested Markov Chain Hyper-Heuristic (NMHH) Framework Efficiency Through Sequential Probability Ratio Testing, 2024 26th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 53-60, doi: 10.1109/SYNASC65383.2024.00022.
  Indexed by IEEE, ***Rank D***, ***1 point***

- **N. Bándi**, R. -B. Tunyogi, Z. Szabó, E. Farkas and C. Sulyok, Image-Based Volume Estimation Using Stereo Vision, 2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2020, pp. 000055-000060, doi: 10.1109/SISY50555.2020.9217089.
  Indexed by IEEE, ***Rank D***, ***0 points*** *(third publication in* ***rank D****)*

Total publication points: ***18***

Submitted papers

- N. Bándi, N. Gaskó, Z. Tasnádi, Generalisations of the node infiltration problem.

Chapter 2

Theoretical background

The process of finding the option that is considered best among a set of available choices can be encountered in many places. The field of optimization aims to formulate an exact mathematical representation of such processes. The exact representation is problem dependent, but the general approach consists of defining the set of feasible solutions (search space) and a "better-than" relation over the elements. Each optimization algorithm defines a process that is able to search for the best solution within the search space.

Classification of optimization problems

Optimization problems can be classified based on multiple criteria, such as their type of search space, solution constraints, type of objective function and determinism.

Discrete - continuous Considering their search space, problems can be considered to be either continuous, discrete or a mixture of both. Discrete problems can be further classified according to the nature of the elements within the discrete search space. Some problems are bound to have integer solutions, other problems are restricted to a binary search space. Combinatorial problems often have a large but discrete search space.

Convex - non-convex The subset of optimization problems that are considered convex have the property that their local minima coincide with the global minimum and are solved in an easier manner compared to non-convex problems.

Linear - non-linear Linear optimization (Linear programming) problems contain only linear relationships within the objective and constraints [22]. Linear optimization problems are convex and can be solved with approaches taking advantage of this property, such as simplex or interior point methods [22, 43]. Non-linear problems contain at least one non-linear term in the objective or constraints.

Deterministic - stochastic Problems can also be classified based on the nature of their inputs. Stochastic problems contain a degree of uncertainty within the parameters, while deterministic problems have a fixed domain and objective. Examples of approaches solving non-deterministic optimization problems include robust optimization [8] and stochastic programming [10].

2.1 Optimization methods

The approach to solving optimization problems varies according to the nature of the objective, the dimensionality of the problem, and the available computational resources.

Deriving the exact optimum directly is feasible in the case of a small number of problems that have analytical solutions. If the direct solution approach is not feasible due to either the size of the problem or the lack of an analytical solution, the iterative approximation of the optimum is necessary.

Complex non-convex, non-linear problems can be solved by iterative improvement of an initial solution [45]. The starting points are chosen randomly or on the basis of prior knowledge of the search space. Methods based on local search are sensitive to the initial point, as it defines the local minimum towards which the solutions converge. Multiple restarts of the optimization process can improve the range of final solutions [39].

The optimization process is stopped when at least one condition of the stopping conditions is satisfied.

2.1.1 Iterative search operator classification

Iterative optimization algorithms can be classified according to the search strategy they use. The most important strategies are local, metaheuristic search and hybrid.

Local search Local search operators iteratively explore the immediate neighboring region of the solution in each iteration.

Some operators evaluate neighbouring points and choose the best one as the next step, others model the shape of the objective around the region of the current solution and select the most promising neighboring value [51].

Metaheuristic search Local search operators perform well when the initial point lies within the region of the global minimum. In every other case they will not be able to escape the local minimum in order to reach the global optimum. Metaheuristic search operators aim to explore the whole search space without getting stuck in

local minima [41]. This operator category is diverse. Some approaches (such as genetic algorithm, genetic programming, differential evolution) rely on the evolution of a population of solutions [59, 55]. Other approaches are inspired by physical phenomena, such as simulated annealing or gravitational search algorithm [36, 50].

Hybrid search Hybrid techniques combine both local and metaheuristic search elements in order to benefit from their advantages [56].

2.1.2 Challenges of optimization techniques

The design and process of optimization is riddled with challenges. The problem dimensionality, non-determinism, resource constraints all need to be taken into consideration in order to solve a problem. Some algorithms may perform well on certain distributions of problems and not for others. Finding the right algorithm and algorithm parameters for a given problem is often problematic and subject to trial and error.

No free lunch theorem The no free lunch theorem (NFL) [60] states that there is no algorithm that outperforms any other algorithm when considering all possible optimization problems.

Basins of attraction Some objective functions have many local minima which can trap the optimization process in a suboptimal solution.

Curse of dimensionality The size of the search space can increase exponentially with each extra dimension [7]. If the problem structure does not allow the effective exclusion of suboptimal subsets of the search space at each iteration then exploring the search space can become problematic.

Resource limitations The optimization process is often constrained by limitations on the available resources. Large optimization problems require many computational resources and computational time, which can be bounded by available hardware and time constraints.

Solution constraints Many problems impose complex constraints on the problem solutions [37]. Satisfying these constraints can be hard if the generation of these solutions cannot be done in a systematic way.

Exploration-exploitation tradeoff At each iteration within the optimization process the search operator can either try to escape the attraction base of the current solution, or it can improve the current solution.

Finding the right balance for a problem is difficult without prior knowledge of the structure of the problem [41].

2.2 Hyper-heuristics

The literature contains many optimization algorithms and heuristics, some inspired by nature, others inspired by physics or other phenomena. Finding the appropriate approach is often problem-specific and can be tedious.

Recently, hyper-heuristic algorithms have been of huge interest, as they provide an automatic way of selecting or generating heuristics for unseen problems (see [52] for a review). These approaches can be thought of as the optimization of the optimization process. The selection or generation of heuristics yields a problem-specific optimization algorithm that in many cases performs better than a single standard heuristic.

2.2.1 Motivation

The motivation for the creation of hyper-heuristics appeared in the 1960s. According to [28] the combination of different search algorithms produced better results than only simple algorithms. These optimization techniques aim to automate the selection, generation and evolution of heuristics in order to improve the optimization process. As hyper-heuristics seemed to be an emerging area in algorithm design, several surveys captured the history of hyper-heuristics and the most important research directions [11, 13, 24].

The automation of heuristic design and adaptation is motivated by the lack of efficiency of designing problem-specific heuristics and the dependence on domain expertise. Hyper-heuristics can adapt more easily to changes in the problem definition.

These methods aim to automatically adapt to each specific problem instance. Hyper-heuristics usually leverage multiple heuristics which can improve the overall results compared to relying on a single heuristic.

2.2.2 Classification of hyper-heuristics

Hyper-heuristics can be classified according to multiple criteria. Considering the way they adapt and apply the heuristics they can be either selection based or generation based methods as presented in [12]. Selection based hyper-heuristics choose

the most appropriate heuristic at each iteration step. These methods leverage either a learning mechanism or an evolutionary strategy to select the most appropriate operators. In [40] a new taxonomy for interactive optimization, where an interaction between the user and the system exists is proposed.

Four categories of heuristic selection are mentioned in [18]. Meta-heuristic-based approaches employ Genetic Algorithms [21], Simulated Annealing [3], Tabu search [35] or some other metaheuristic for the selection process. Other approaches use reinforcement learning for adaptive selection. [34] uses the Conditional Markov Chain Search (CMCS) algorithm for the Bipartite Boolean Quadratic Programming Problem (BBQP). Greedy selection methods preliminarily evaluate all heuristics and choose the one that performs best at each step [21]. Learning based methods leverage models such as neural networks or reinforcement learning in order to exploit the identified heuristic performance patterns [38].

Generation based hyper-heuristics create problem specific heuristics for each problem instance. These methods rely on genetic programming techniques to derive new heuristics. Heuristcs are expressed as executable programs which can be adapted and evolved to fit the problem instance.

In [25] a new classification of hyper-heuristics that extends the classification proposed in [12] is presented. In this new taxonomy, six properties of hyper-heuristics are mentioned: based on feedback, low-level heuristics, objectives, solutions, move acceptance, and parameter setting.

Hyper-heuristics can use either constructive or perturbative heuristics. Constructive heuristics are applied iteratively to incrementally generate solutions and perturbation heuristics start from and improve existing solutions.

Hyper-heuristics can be classified based on the nature of their adaptation or learning mechanism. Online learning approaches adapt their strategy during the problem solving process. These methods rely on tracking the performance of the individual heuristics during optimization. They can employ reinforcement-learning or adaptive memory.

Offline hyper-heuristics can rely on information pertaining to other problem instances and historical performance of heuristics and configurations. These methods can leverage problem instance related features.

Chapter 3

Proposed hyper-heuristic framework

This chapter introduces our contribution to the field of hyper-heuristics.

Section 3.1 introduces the Nested Markov Chain Hyper-heuristic (NMHH) framework, the models depicted in this section are associated to original research publications [16, 6].

Section 3.2 describes our introduction of sequential probability ratio testing into the hyper-heuristic acceptance process. The model depicted in this section is associated to the original research publication [14].

3.1 Nested Markov Chain Hyper-Heuristic

We introduce a hyper-heuristic model based on nested Markov chains.

Although several hyper-heuristic frameworks have been proposed, most of them are concerned with specific combinatorial optimization problems; only a few are designed to solve continuous numerical optimization problems. A research gap exists regarding hyper-heuristic frameworks that balance the exploration-exploitation rate and tune the operator parameters in an online fashion. As another research gap we can mention the lack of generality of the proposed hyper-heuristic frameworks, the majority of them are incorporating domain specific knowledge about a specific problem (for example [30]).

We propose a new hyper-heuristic framework and present its advantages for single-objective continuous problems and the comparison with a well-known hyper-heuristic and six state-of-the-art metaheuristics. The novelty of our approach consists of introducing a nested Markov chain to the base level for the search for the best-performing heuristic operators and their sequences. Simulated annealing is used on the hyper level, which evolves the chain and the operator parameters. In our approach, the upper level of the Markov chain expressing the hybridization of global and local search operators and the lower level automatically selecting the best-performing operator sequences for the problem. The general formulation of

the model allows the usage of other arbitrary operators, as well. Our model can be used to achieve good optimization results without the user having deep domain (problem specific) knowledge. The limitations of our approach are similar to other hyper-heuristics, finding the right operator configurations and balance can require many function evaluations.

3.1.1 Proposed model

The structure of the proposed hyper-heuristic framework is presented in Figure 3.1. The approach is based on two levels. The base level optimizes the problem, starting with a population of candidate solutions and a limited number of function evaluations for each candidate. The hyper level guides and improves the base level. The hyper level searches the algorithm space by finding the best-performing operator sequences and operator parameters via Simulated Annealing. The base level performs the optimization according to the operator sequence modelled by a nested Markov chain. The first layer models the transitions between perturb, selection, and refinement operators, and the second layer models the operator sequences of each category.

Our approach contains a genetic algorithm (GA) and a differential evolution (DE) operator in the perturb category and an elitist selector (ES) in the select category. It incorporates a refiner category containing the gradient descent (GD) and Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (LBFGS) operators that perform the local search. In this way, the base level is parameterized so that it can express a continuum between exploration and exploitation.

At each step, a statistically significant number of base-level evaluations are performed and the performance metric is the median plus interquartile range of the costs. The next step in the design space is taken by perturbing the previous point by a normally distributed noise factor scaled according to the parameter bounds. The advantage of this formalization lies in the expressiveness of the base level as it allows the selection and combination of sets of operators that have different roles (exploration, exploitation, selection).

The model can perform well in high-dimensional settings, as it can iteratively find the equilibria between exploration and exploitation operators.

Population evolution All operators in the perturb category generate a new population of candidates. The operators in the refine category perform an iterated local search starting with these candidate solutions. The elitist selection operator then selects the best-performing points from the old and new populations to become the

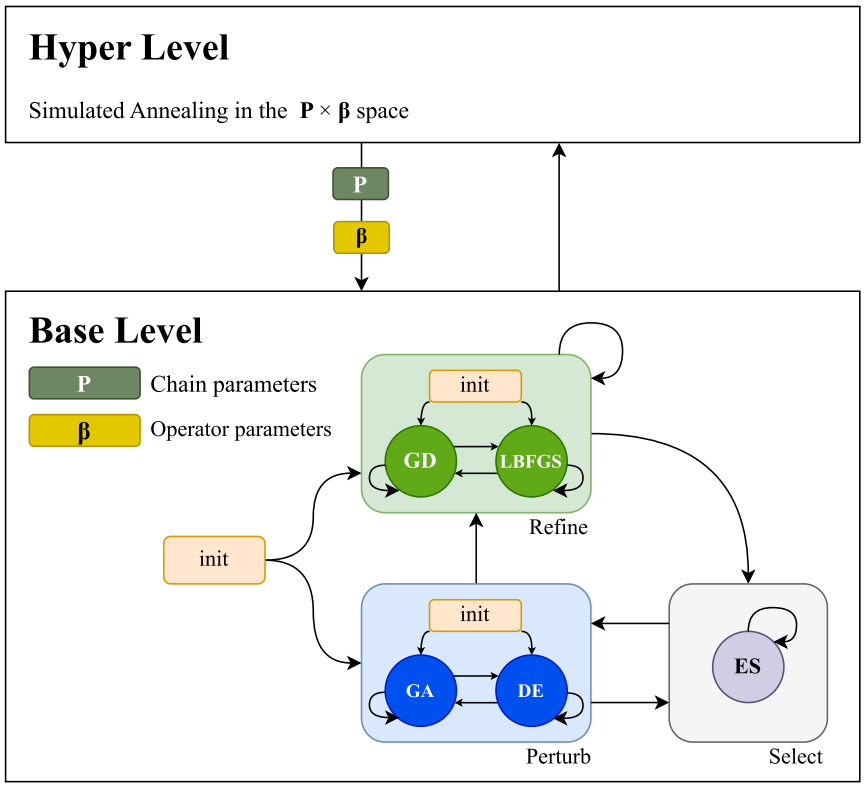


FIGURE 3.1: The structure of the hyper-heuristic framework. The base level is used to optimize the problem using a given number of function evaluations. On this level a genetic algorithm (GA) and differential evolution (DE) operator is used in the perturb category, an elitist selector (ES) in the select category, and in the refiner category the gradient descent (GD) and Limited-memory Broyden, Fletcher, Goldfarb, Shanno algorithm (LBFGS) operators are used. The hyper level is used to guide and improve the base level.

next generation. All perturbed points landing in the attraction basin of a better solution are selected into the next generation when the refiner operators iterate the points closer to these attractors and the ES operator selects them.

3.1.2 Numerical experiments

The results of the numerical experiments conducted were compared to state-of-the-art metaheuristics and another recent hyper-heuristic that incorporates several state-of-the-art metaheuristic operators.

Benchmarks

For the numerical experiments, we used six well-known continuous benchmark functions: Rastrigin, Rosenbrock, Styblinski Tang, Schweffel 2.23, Trid, and Qing. The dimensionality, convexity, separability, and multimodality of the functions were varied to assess performance in different settings.

Results and discussion

The results point to a considerable difference between the hyper-heuristic and standard metaheuristic approaches.

For the majority of problems (Qing, Rosenbrock, Styblinski Tang, Trid), NMHH outperformed all metaheuristics and CUSTOMHyS. In the case of Rastrigin and Schwefel 2.23 which have their optimum in the origin, the SMA, AEO and BRO found the global minimum in all the test cases suggesting a bias towards solutions in the origin. The experiments were repeated with the optimum shifted out of the origin all other experimental settings kept the same. In the shifted case SMA, AEO and BRO are not able to find the shifted optimum and their results are worse. The results of NMHH are not impacted by this shift pointing to the fact that the method is stable. NMHH found the best solution in 65.6% of test cases. These results indicate that in many cases our approach can outperform state-of-the-art metaheuristics and the investigated selection hyper-heuristic.

Ablation test In order to assess the importance of each category within the framework an ablation test was carried out. Two ablated versions were analyzed. SA-REFINE represents the NMHH version that shifts the exploration-exploitation balance entirely towards exploitation by removing the perturb category. Similarly SA-PERTURB represents the NMHH version that shifts this balance entirely towards exploration by removing the refine category. The results point to the fact that both operator categories have a positive impact on the overall performance of NMHH. The refinement category has a bigger role compared to the perturb category because removing it results in a bigger performance decline. Regarding computational time, SA-REFINE proves to be the slowest, as it relies on gradient calculations. NMHH is faster and is more stable than SA-REFINE, but it is slower than SA-PERTURB across dimensions. These results show that finding the right exploration-exploitation balance is indeed necessary to improve optimization performance.

3.1.3 Conclusions and further work

In this section we proposed a new hyper-heuristic framework (NMHH) with two main innovations: the use of a nested Markov chain to model complex distributions of operators and the search for the equilibrium between exploration and exploitation in this space by balancing the category of iterated local search against metaheuristic exploratory operators. Numerical experiments conducted on continuous benchmark problems in high dimensions confirm the effectiveness of the proposed approach. The results show that NMHH evolved operator sequences and found the exploration-exploitation rate that outperformed state-of-the-art metaheuristics and

the CUSTOMHyS hyper-heuristic in 65.6% of the cases in the high-dimensional setting.

3.2 Sequential probability ratio testing in hyper-heuristics

The effectiveness of hyper-heuristic methods is limited by the number of base level design configurations that can be assessed. The assessments are usually limited by computational constraints such as the number of available function evaluations or computational time. It is important that these resources are utilized effectivelly in order to improve results. One way of reducing computational costs is implementing an efficient sampling method that reduces the number of function evaluations.

Sequential testing is a well-known technique with a rich history of applications in many fields [31, 58].

The idea of stopping the sampling process at the moment of sufficient confidence for the acceptance of a hypothesis can prove to be useful when sampling is costly.

This section presents results of introducing sequential sampling to the performance sampling step of a hyper-heuristic method, an approach which can be generalized to other hyper-heuristic methods leveraging performance sampling as well. In order to show the possible performance improvement that can be gained using the proposed approach, the nested Markov chain hyper-heuristic (NMHH) framework is used [16]. This section proposes a reduction in the number of function evaluations required to find this balance and the right heuristic design.

3.2.1 Proposed framework

The proposed model is based on NMHH, a model that utilizes a nested Markov chain in the heuristic design process. The number of samples used can be decreased by using sequential testing and early stopping when a hypothesis can be accepted with a reasonable amount of confidence. The sampling process is stopped as soon as there is sufficient evidence that the current design point is not better than the best one yet. This approach allows the testing of more design points with the same function evaluation budget.

3.2.2 Numerical experiments

The numerical experiments were performed by comparing the hyper-heuristic results on 12 standard continuous benchmark problems. These differ with respect to convexity, separability an multimodality.

3.2.3 Results

The proposed approach (SPRT-NMHH) had the highest score in 93% of test cases. The second best performing approach is NMHH, which outperformed all the metaheuristics and CUSTOMHyS in 74% of dimensions. The AEO, SMA, BRO algorithms have all performed remarkably well independently of the number of dimensions in the case of Rastrigin, Schwefel223, Sphere, and SumSquares.

These results are due to the fact that these objectives have their optimum in the origin as shown in Subsection 3.1.2. Both NMHH and SPRT-NMHH perform well on Sphere. This is explainable by the fact that the LBFGS operator is able to model the objective accurately and is able to minimize at every step.

3.2.4 Conclusion

This section proposes the improvement of the hyper-heuristic performance sampling process. The fixed sample sized hypothesis testing is replaced with sequential sampling (SPRT) in order to reject worse performing heuristic configurations as fast as possible and reduce the total number of performance samples required. Numerical testing was carried out on well-known benchmark functions and the proposed method (SPRT-NMHH) was compared to the fixed sample size version (NMHH), another hyper-heuristic (CUSTOMHyS), and to six metaheuristics. Both low and high dimensional problems were considered. The Wilcoxon rank-sum statistical test was used to compare the methods. The proposed method (SPRT- NMHH) outperformed all other methods considering all problems in 93% of the test cases considered. These results point to the fact that sequential sampling can reduce the number of performance samples needed and improve the overall results. The investigation of the effect of the significance rate α, β and maximum sample size max_s on overall performance is proposed as a future research direction.

Chapter 4

Applications

This chapter presents applications of the proposed framework to real problem settings.

Section 4.1 depicts the adaptation of the Nested Markov Chain Hyper-Heuristic to the problem of hyperparameter tuning. The results shown in this section are associated to the original research publication [15]. Section 4.2 presents a hybrid hyper level search strategy and the application of the framework to clustering. The results presented in this section are associated to the original research publication [5].

Section 4.3 addresses the sensor network localisation problem with a manually configured and automatically designed heuristic approach. The results presented in this section are associated to the original research publication [4].

4.1 Application To Hyperparameter Tuning

This section presents results associated to the adaptation of the proposed hyper-heuristic framework to the problem of hyperparameter tuning.

This direction constitutes an exploration of the power of hyper-heuristics in the context of machine learning, mainly in classification.

Although there exists a wide literature in this topic, due to the complexity of the problem and the different nature of the classification algorithms, new hyperparameter optimization algorithms can still improve the accuracy of the existing classification algorithms.

The main goal of this section is to study an adaptation of the proposed hyper-heuristic framework (NMHH), that is called Nested Markov Hyper-Tuner (NMHT).

4.1.1 Problem statement

Hyperparameter optimization (or tuning) is an essential task in machine learning, having huge importance in increasing the performance of models for specific problem instances. The behavior and performance of most machine learning models

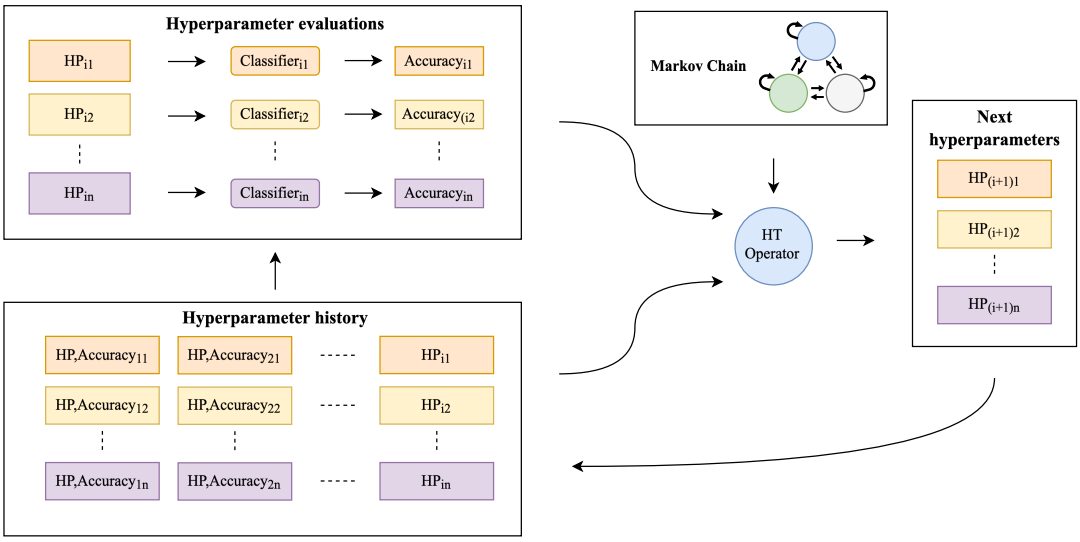


FIGURE 4.1: The hyperparameter optimization flow within the base level. The Markov chain defines the hyperparameter operator sequence. Each operator derives the next hyperparameter population leveraging the parameter history.

depend on certain parameters and choices defined before training. Finding the right model configuration is essential but quite expensive as each evaluation requires the training of the model.

4.1.2 Adapted framework

The studied framework is an adaptation of NMHH, the hyper-heuristic framework that we have introduced in Section 3.1. Due to the high complexity of the search space (non-differentiable, black-box) the gradient based operators in the exploitation category were replaced with Bayesian optimization based operators.

Architecture

The framework follows the general architecture of hyper-heuristic frameworks. The hyper level starts with a random configuration within $\beta \times P$ and iteratively improves it via Simulated Annealing. The Simulated Annealing operator perturbs the configuration in order to iteratively improve it, initially through exploration and later through exploitation. The base level maintains a population of solutions that are randomly initiated at each hyperheuristic step. Each operator operates on the population and produces the next generation of solutions. It applies the operators according to the Markov Chains configured in the hyper level for a restricted number of function evaluations as depicted in Figure 4.1. The hyperparameter search history is maintained and available to the Bayesian optimization based operators in order to create and improve the surrogate model. The performance of the configuration is given as feedback to the upper level and recorded in the hyperparameter

TABLE 4.1: Datasets used for the numerical experiments and basic properties

Name	No. instances	No. features	Feature type	No. Classes	Ref.
Audit Risk	777	26	Integer,Real	2	Nishtha Hooda, UCI ML repository
Cervical cancer	858	36	Integer,Real	2	[27]
Wine	178	13	Integer, Real	3	[1]
Gallstone	320	38	Real	2	[26]
Digits	1797	64	Integer	10	[23]
HeartFailure	299	12	Integer,Real	2	[19]

history. The transition matrices are constrained to avoid invalid operations, such as multiple consecutive selections. The optimization formulation allows for the expression of both population based algorithms and single solution algorithms. The search space in the hyper level enables the search for the optimal balance between operator categories (such as the balance between exploration and exploitation) and the right operator parameters.

Objective function, search space In the case of hyperparameter tuning of classification models, the objective of the base level was defined as the search for the hyperparameters that maximize the accuracy cross-validation score. The numerical hyperparameters are constrained to be within their feasible domain, the categorical parameters are represented as integers during the optimization process.

4.1.3 Numerical experiments

The proposed method was tested on four classification models and six datasets. The method is compared to five other hyperparameter tuning methods.

For experiments we used four state-of-the-art classifiers: Decision Tree classifier [49], Random Forest [46], Gradient Boosting [9], and Support Vector Machine [17].

The compared methods tuned the hyperparameters of these four classifiers. The implementations provided by the scikit-learn package were used [47].

Six small-medium sized datasets were chosen for testing the proposed approach in order to limit the computational time required for the experiments. The six datasets used were Audit Risk - classifying suspicious firms collected from Auditor office of India in years 2015 and 2016[1], cervical cancer prediction [27], gallstone prediction [26], heart failure prediction [19], Wine dataset [1] - which contains chemical analysis of different cultivates, and Digits classification from MNIST [23]. Basic properties of the used datasets are presented in Table 4.1.

[1]Downloaded from https://archive.ics.uci.edu/dataset/475/audit+data, last accessed at 24/08/2025

4.1.4 Results and conclusion

In most cases our proposed method (NMHT) and the BayesGP seemed to have best performances taking into account all four used classifiers. In the case of the AuditRisk dataset almost all hyperparameter tuning methods had similar results, for this dataset the default parameter setting was sufficient, as well. For all other five datasets, the tuning approach heavily impacted the final results.

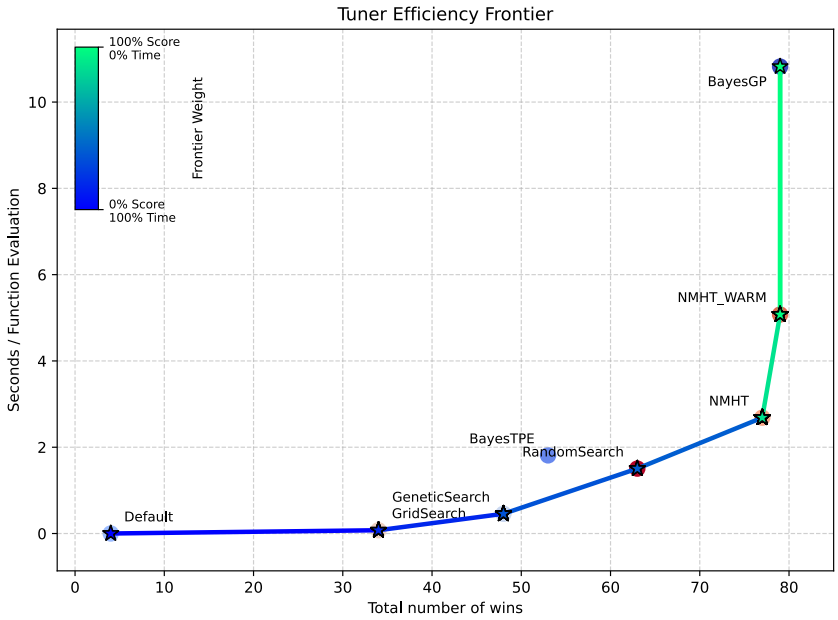


FIGURE 4.2: Efficiency frontier of the hyperparameter tuning methods. The blue-green gradient represents different tradeoffs between speed and performance.

Figure 4.2 presents the efficiency frontier of the compared methods when taking computational time into consideration as well. The default parameter approach had the worst performance, as expected. Grid search, genetic search, and BayesTPE outperformed the default settings with BayesTPE in the lead. The random approach outperformed these, pointing to the fact that under noisy conditions and limited training budget, grid search, genetic search, and BayesTPE might not be effective. BayesGP was able to significantly outperform random search as it was able to efficiently sample the search space and adapt to the cost landscape. When considering the total number of wins, NMHT_WARM matched the performance of the BayesGP approach. The simple NMHT was outperformed by the warm initialized version highlighting the importance of the consideration of many base level configurations. The highest computational cost was incurred by BayesGP, due to its cubic complexity. Our approaches had a significantly smaller computational cost compared to the BayesGP, but higher than the other methods, representing tradeoffs between speed and performance. The results point to the fact finding the right hedge of tuning

approaches can have a positive impact on results and computational cost. Each approach has some advantages and limitations, their right combination can reduce the limitations and complement their strengths.

4.2 Hyper level search hybridization. A clustering application.

The process of Simulated Annealing in the hyper level can be improved by leveraging search operators that can take parameter dependencies into consideration. This section presents work regarding a hybrid variant of the Nested Markov Chain Hyper Heuristic (NMHH). The application of the resulting variant to a clustering problem is also presented.

4.2.1 Improved framework

The Simulated Annealing operator used in the hyper level has the drawback of not considering previous points and the relationship between variables. Other operators such as the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [32], or Mesh Adaptive Direct Search (MADS) [2] attempt to address this limitation. These operators are not using any derivative information, making them suitable for the black box search space that is the base level design space.

The studied framework is an improvement of the hyper-heuristic framework that we have introduced (NMHH). The studied framework improves the search procedure of NMHH. We study the hybridization of SA with MADS and CMA-ES. The SA operator represents an initialization strategy and MADS, CMA-ES a refinement strategy.

Hybridization process In the improved model, the hyper level combines SA with either MADS or CMA-ES. The search process starts with SA to perform a global search and reach a reasonably good solution. In the next step, this solution is refined by MADS or CMA-ES. Depending on whether global or local search is emphasised, two hybridizations are proposed. In the first, the SA operator is applied for 33% of steps (SA-MADS, SA-CMA-ES); in the second, it is applied for 66% of the steps (BIGSA-MADS, BIGSA-CMA-ES). For each hybridized method, two variants were studied: on the base level with GWO (mentioned in the name of the algorithm) and without.

4.2.2 Numerical experiments

For numerical experiments, 12 well-known test functions are used.

Results and discussion

Different variants of the hybridization on the hyper level were studied.

The best-performing algorithm is the BIGSA-CMA-ES-GWO, both in low and high dimensions. The second-best-performing variant for the low dimensions is the BIGSA-CMA-ES, while in the case of high dimensions the BIGSA-MADS-GWO is second.

4.2.3 A clustering application

Data clustering is an emerging area in data analysis techniques, meaning the grouping of data entities in different groups, data in the same cluster having greater similarity than data in different clusters.

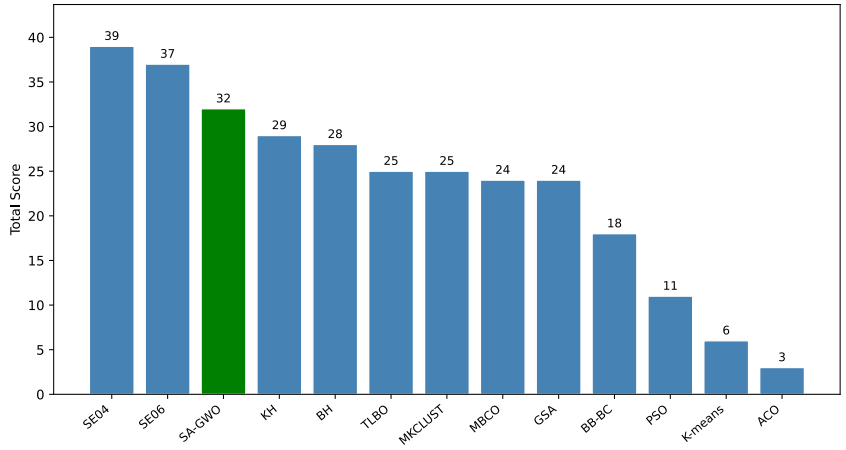
For numerical experiments, four datasets were used: the Iris dataset[2] with 150 data instances, four features and three clusters (classes) [57], the Wine dataset[3] with 178 instances, 13 features and three clusters, the Glass dataset[4] with 214 instances, nine features and six clusters, and the CMC dataset[5] with 1473 instances, nine features and three clusters. The SA-GWO hyper-heuristic was compared to 12 other well known clustering methods from the literature.

In the case of the Glass dataset, our method outperformed all other 12 algorithms. In the CMC and Wine datasets it occupied second and third place, having differences with the first place only in the second decimal, while in the case of the Iris dataset it held the eighth place. The overall scoring is depicted in Figure 4.3 plot (a). Overall our approach ranked third place out of thirteen. The experiment setting in which our method (BIGSA-CMA-ES-GWO) was left to converge was also tested. These results are depicted in Figure 4.3 plot (b). The method matched the performance of the best clustering method among those compared. This result highlights the trade-off between having better results and having less function evaluations.

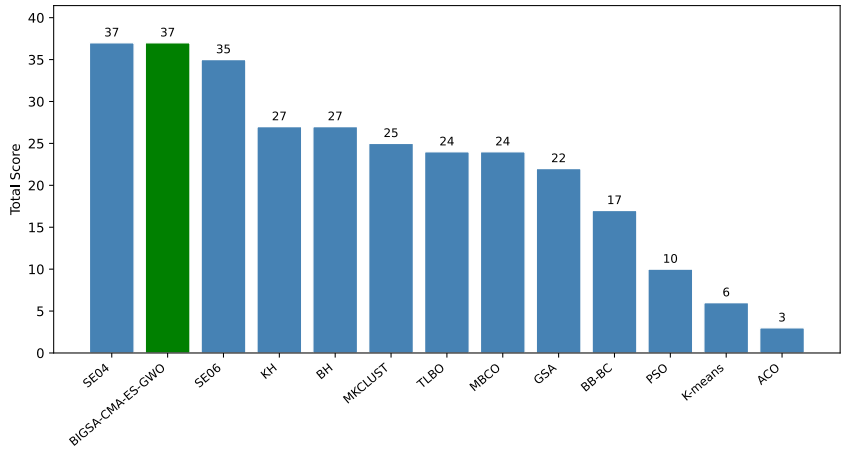
4.2.4 Conclusion

In this section, we studied an improved hyper-heuristic framework for continuous optimization problems, with significant improvements on the hyper level, hybridized SA variants were tested. Numerical experiments conducted up to 750 dimensions for continuous benchmark problems prove the potential of the framework.

[2]https://archive.ics.uci.edu/dataset/53/iris
[3]https://archive.ics.uci.edu/dataset/109/wine
[4]https://archive.ics.uci.edu/dataset/42/glass+identification
[5]https://archive.ics.uci.edu/dataset/30/contraceptive+method+choice



(a) Test setting where our method (SA-GWO) had limited hyper-heuristic steps.



(b) Test setting where our method (BIGSA-CMA-ES-GWO) was left to converge.

FIGURE 4.3: Total score for each clustering method. Each score represents how many times a given method outperformed another method.

Based on the comparisons with other methods, the new variants in most cases outperform other algorithms on the benchmark problems considered. As an application, we present a clustering problem, where our hyper-heuristic framework is competitive with existing state-of-the-art clustering techniques without using any domain specific search operators. When allowing for a higher computational budget, our method matches the performance of the best clustering method among those compared.

4.3 Addressing the sensor network localisation problem

Sensor networks have gained a great deal of attention in recent years. Although they are widely used to collect metrics of the environment, this data is useful only with the positional information of the sensors [29]. Since equipping each sensor with a GPS tracking system is costly, there is a need for other solutions.

This section compares a manually finetuned hybrid algorithm based on differential evolution to the NMHH approach. The hybrid approach combines local search algorithms (GD and a LBFGS) and the well-known differential evolution (DE) algorithm [4]. The automatic design approach is also investigated, leveraging the Nested Markov Chain Hyper-Heuristic for the SNLP. Both approaches are compared to domain specific state-of-the-art algorithms as well: DV-Hop [44], Amorphous [42], APIT [33], BoundingBox [54], MDS-MAP [53].

A widely encountered variant consists of the knowledge of the true position of only a few sensors, and the distance measurements between a subset of the others [20]. The problem can be mathematically formulated as a graph realization problem, where the positions of the nodes have to be estimated in either two or three-dimensions.

4.3.1 Numerical experiments and results

The methods were tested on randomly generated problems of varying complexity. The number of anchor points, the communications radius and the anchor point location precision were varied. The results show that the hybrid method is more precise than DV-Hop, Amorphous, Bounding-Box and APIT by an order of magnitude. MDS-Map has similar precision to the hybrid approach and the configurations automatically designed by NMHH were performing slightly better than the hybrid approach. The automatically designed configuration was in the top two positions in each test case.

4.3.2 Conclusion

This section highlighted the advantages of the automatic design capabilities compared to manual fine-tuning of heuristic operators. The SNLP was used as a case study to compare the performance of various domain specific algorithms, manually configured hybrid heuristics and automatically designed algorithms.

The results show that better performance comes at the cost of higher computational time. The non-iterative methods were the fastest but ranked lower overall. NMHH had the highest computational cost but ranked the highest overall.

Chapter 5

Conclusions

Optimization plays a key role in many fields, ranging from operations research and machine learning to finance and physics. These problems are of varying complexity and there have been many solution approaches presented in the literature. Domain experts can address computationally challenging problems using heuristics and metaheuristic algorithms. In order to reduce dependence on domain expertise, another research area called hyper-heuristics emerged in the field. These methods aim to automate the search for the best heuristic or set of heuristics for a particular problem. Higher level heuristics are used to explore the space of lower level heuristics, resulting in a more general approach available to non-experts. This work is motivated by the fact that the field is relatively understudied and there are many research directions that have not been explored. Another reason was that the potential impact of these methods is high due to their broad applicability. Hyper-heuristics present their own set of trade-offs, which need to be explored and addressed. Better performance and generality usually come at the price of higher computational cost and having appropriate building blocks. Among the objectives of the thesis were: identifying research gaps in the literature, proposing a novel hyper-heuristic method, validating it using the methodology accepted by the research community, and demonstrating its applicability to real-world problems.

5.1 Contributions and results

The contributions of the thesis align with the objectives that were set out. The first objective was the survey of the literature and the identification of research gaps and aspects of existing approaches that can be improved. The theoretical background, a recent survey of the literature and related work have been presented in Chapter 2. As for the identified research gaps we can mention the underexploration of methods that can alleviate the requirement for many function evaluations such as effective sampling. Another identified research gap was that most existing methods adapt to choose only the next best heuristic based on past performance which limits the expression of complex operator sequence distributions. Most methods consider the

low level heuristic set as homogeneous. Approaches that impose structure such as categorization to explicitly control the exploration and exploitation rate are understudied. The second objective was the contribution of novel hyper-heuristic methods in order to fill in the identified research gaps. Chapter 3 presents contributions in this direction. A novel method is proposed that searches in the space of complex operator sequence distributions. This hyper-heuristic framework aims to improve on existing frameworks via the introduction of nested Markov chains to model complex distributions. The Nested Markov Chain Hyper-Heuristic (NMHH) introduces structure into the lower level heuristic set partitioning it into distinct categories. This structure enables the search for the exploration-exploitation equilibrium. Another contribution aimed to reduce the number of function evaluations needed to find a good heuristic design. The integration of sequential probability ratio testing into the hyper-level search process proved to be beneficial. The third objective was the carrying out of experiments and comparisons with state-of-the-art methods. The validation needs to be carried out according to the evaluation methodology accepted by the literature. The proposed models were tested using numerical experiments and compared with both a state-of-the-art hyper-heuristic and recent metaheuristics according to the literature methodology. The literature suggests using a wide range of benchmark functions to assess performance and adaptability to different cost landscapes. Furthermore the comparisons are to be made using statistical tests, and in order for the comparison to be considered fair, all methods are limited to the same number of total function evaluations. Both the numerical results and the computational times are interpreted. Ablation tests were carried out in order to assess the importance of different components of our method. The fourth objective was reflecting on the experimental results, and the identification of improvement possibilities. The results show that our method managed to outperform the other methods considered in most of the test cases on the considered benchmark functions, having a similar computational cost as the other hyper-heuristic and larger than the metaheuristics. The cost effective sampling method proved to be beneficial, as it enabled the consideration of a much larger set of heuristic configurations. An expanded experimental setting covering a larger set of problems and dimensionalities was used for validation. As for the potential improvement possibility, the hybridization of the hyper-level search process was identified. Simulated Annealing does not consider historical results, even though it could be beneficial. In Section 4.2 a hybrid hyper-level variant is presented. In this model Simulated Annealing is used to find a good initial point, and afterwards MADS or CMA-ES operators are used for fine-tuning. The method is compared to another state-of-the-art hyper-heuristic, recent metaheuristics, and to the non-hybrid version. The hybrid variants

consistently scored significantly higher than the other hyper-heuristic and the metaheuristics, and slightly higher than the non-hybrid variant. The last objective was the application of the novel methods to relevant real-world problems. Chapter 4 presents three applications. In the first, the problem of hyperparameter tuning is addressed. The NMHH framework is adapted to the problem with the introduction of hyperparameter tuning specific operators. The adapted method was compared to state-of-the-art approaches such as Bayesian optimization, grid, random and evolutionary search on a set of classification problems. The results show that hedging the right hyperparameter tuning operators can provide a tradeoff between accuracy and computational time. The results also point to an important limitation of hyper-heuristics, namely that they are highly impacted by the available number of objective function evaluations. The second application was to the problem of data clustering. In this case our method was not adapted to this particular problem, that is it did not have any clustering specific heuristic operators but kept the original generic optimization operators. Numerical experiments were carried out on four datasets, and the results compared to the results of twelve other clustering algorithms. Our method managed to reach the third place out of thirteen. When given a higher computational budget, our method was able to match the performance of the best performing algorithm. These results further underline the fact that hyper-heuristics are able to provide a means to reach better results to non-experts, enabling them to trade a higher computational budget for better performance in a straightforward way. The third application was to the problem of sensor network localization. Our method was not adapted to this particular problem either. Numerical experiments were carried out on several simulated network environments under noisy conditions. The method was compared to five domain-specific algorithms and two manually configured hybrid metaheuristics. The results show that the automatically designed heuristic configuration had the highest score among all algorithms but at the price of a higher computational time. Considering all applications the experimental results show that the proposed hyper-heuristic framework variants are able to reach relatively good results but are not constantly the absolute best performers. When given a higher computational budget, they can improve their performance which is useful. Overall, it can be said that hyper-heuristics can be an important first step to non-experts to handle challenging optimization problems and they can serve as a baseline for further problem specific heuristics.

Bibliography

[1] Stefan Aeberhard, Danny Coomans, and Olivier Y. de Vel. "Comparative analysis of statistical pattern recognition methods in high dimensional settings". In: *Pattern Recognit.* 27 (1994), pp. 1065–1077.

[2] Charles Audet and J. E. Dennis. "Mesh Adaptive Direct Search Algorithms for Constrained Optimization". In: *SIAM Journal on Optimization* 17.1 (2006), pp. 188–217.

[3] Ruibin Bai and Graham Kendall. "An Investigation of Automated Planograms Using a Simulated Annealing Based Hyper-Heuristic". In: *Metaheuristics: Progress as Real Problem Solvers*. Vol. 32. Boston, MA: Springer US, 2005, pp. 87–108. ISBN: 978-0-387-25383-1.

[4] Nándor Bándi and Noémi Gaskó. "A Hybrid Differential-Evolution-Based Approach to the Sensor Network Localisation Problem". In: *Intelligent Systems Design and Applications*. Ed. by Ajith Abraham et al. Cham: Springer Nature Switzerland, 2024, pp. 50–61. ISBN: 978-3-031-64650-8.

[5] Nándor Bándi and Noémi Gaskó. "Improved Nested Markov Chain Hyperheuristic Framework. A Clustering Application". In: *Artificial Intelligence Applications and Innovations*. Ed. by Ilias Maglogiannis et al. Cham: Springer Nature Switzerland, 2025, pp. 204–220. ISBN: 978-3-031-96239-4.

[6] Nándor Bándi and Noémi Gaskó. "Solving Continuous Optimization Problems with a New Hyperheuristic Framework". In: *Machine Learning, Optimization, and Data Science*. Ed. by Giuseppe Nicosia et al. Cham: Springer Nature Switzerland, 2024, pp. 116–130. ISBN: 978-3-031-53969-5.

[7] Richard Bellman. *Dynamic Programming*. Dover Publications, 1957. ISBN: 9780486428093.

[8] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[9] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms". In: *Artificial Intelligence Review* 54 (2021), pp. 1937–1967.

[10] John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, July 1997. ISBN: 0387982175.

[11] Edmund Burke et al. "Hyper-heuristics: An emerging direction in modern search technology". In: *Handbook of metaheuristics* (2003), pp. 457–474.

[12] Edmund K Burke et al. "A classification of hyper-heuristic approaches". In: *Handbook of metaheuristics* (2010), pp. 449–468.

[13] Edmund K Burke et al. "Hyper-heuristics: A survey of the state of the art". In: *Journal of the Operational Research Society* 64 (2013), pp. 1695–1724.

[14] Nándor Bándi. "Improving the Nested Markov Chain Hyper-Heuristic (NMHH) Framework Efficiency Through Sequential Probability Ratio Testing". In: *2024 26th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. 2024, pp. 53–60.

[15] Nándor Bándi and Noémi Gaskó. "Hyperparameter Tuning of Classification Algorithms with a Hyper-Heuristic Framework - the Nested Markov Hyper-Tuner". In: *2025 IEEE 37th International Conference on Tools with Artificial Intelligence (ICTAI)*. 2025, pp. 951–957.

[16] Nándor Bándi and Noémi Gaskó. "Nested Markov chain Hyper-Heuristic (NMHH): A hybrid hyper-heuristic framework for single-objective continuous problems". In: *PeerJ Computer Science* 10 (2024).

[17] Jair Cervantes et al. "A comprehensive survey on support vector machine classification: Applications, challenges and trends". In: *Neurocomputing* 408 (2020), pp. 189–215.

[18] Konstantin Chakhlevitch and Peter Cowling. "Hyperheuristics: Recent Developments". In: *Adaptive and Multilevel Metaheuristics*. Vol. 136. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 3–29. ISBN: 978-3-540-79438-7.

[19] Davide Chicco and Giuseppe Jurman. "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone". In: *BMC Medical Informatics and Decision Making* 20 (2020).

[20] Chee-Yee Chong and Srikanta P Kumar. "Sensor networks: evolution, opportunities, and challenges". In: *Proceedings of the IEEE* 91.8 (2003), pp. 1247–1256.

[21] Peter Cowling, Graham Kendall, and Eric Soubeiga. "A Hyperheuristic Approach to Scheduling a Sales Summit". In: *Practice and Theory of Automated Timetabling III*. Vol. 2079. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 176–190. ISBN: 978-3-540-44629-3.

[22] George B. Dantzig. *Linear Programming and Extensions*. United States Air Force Project RAND. The RAND Corporation, 1963.

[23] Li Deng. "The MNIST database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[24] Tansel Dokeroglu, Tayfun Kucukyilmaz, and El-Ghazali Talbi. "Hyper-heuristics: A survey and taxonomy". In: *Computers & Industrial Engineering* (2023), p. 109815.

[25] John H Drake et al. "Recent advances in selection hyper-heuristics". In: *European Journal of Operational Research* 285.2 (2020), pp. 405–428.

[26] İrfan Esen et al. "Early prediction of gallstone disease with a machine learning-based method from bioimpedance and laboratory data". In: *Medicine* 103.8 (2024), e37258.

[27] Kelwin Fernandes, Jaime S. Cardoso, and Jessica C. Fernandes. "Transfer Learning with Partial Observability Applied to Cervical Cancer Screening". In: *Iberian Conference on Pattern Recognition and Image Analysis*. 2017.

[28] H Fisher and G Thompson. "Probabilistic learning combination of local job-shop scheduling rules. Prentice-Hall, Englewood Cliffs". In: *Industrial Scheduling* (1963).

[29] Christodoulos A Floudas and P M Pardalos. *Encyclopedia of optimization. Vol. 5*. Springer, 2009.

[30] Francesca Guerriero and Francesco Paolo Saccomanno. "A hierarchical hyper-heuristic for the bin packing problem". In: *Soft Computing* 27.18 (2023), pp. 12997–13010.

[31] J. Hajnal. "A Two-Sample Sequential t-Test". In: *Biometrika* 48.1/2 (1961), pp. 65–75. ISSN: 00063444.

[32] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)". In: *Evolutionary Computation* 11.1 (2003), pp. 1–18.

[33] Tian He et al. "Range-free localization schemes for large scale sensor networks". In: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*. MobiCom '03. San Diego, CA, USA: Association for Computing Machinery, 2003, 81–95. ISBN: 1581137532.

[34] Daniel Karapetyan, Abraham P. Punnen, and Andrew J. Parkes. "Markov Chain methods for the Bipartite Boolean Quadratic Programming Problem". In: *European Journal of Operational Research* 260.2 (2017), pp. 494–506. ISSN: 0377-2217.

[35] Graham Kendall and Naimah Mohd Hussin. "An investigation of a tabu-search-based hyper-heuristic for examination timetabling". In: *Multidisciplinary Scheduling: Theory and Applications: 1 st International Conference, MISTA'03 Nottingham, UK, 13–15 August 2003 Selected Papers*. Springer. 2005, pp. 309–328.

[36] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing". In: *Science* 220.4598 (1983), pp. 671–680.

[37] Harold W. Kuhn and Albert W. Tucker. "Nonlinear Programming". In: *Traces and Emergence of Nonlinear Programming*. Ed. by Giorgio Giorgi and Tinne Hoff Kjeldsen. Basel: Springer Basel, 2014, pp. 247–258. ISBN: 978-3-0348-0439-4.

[38] Michail G Lagoudakis, Michael L Littman, et al. "Algorithm Selection using Reinforcement Learning." In: *ICML*. 2000, pp. 511–518.

[39] Helena Lourenço, Olivier Martin, and Thomas Stützle. "Iterated Local Search". In: Jan. 2003, pp. 321–353. ISBN: 978-1-4020-7263-5.

[40] David Meignan et al. "A review and taxonomy of interactive optimization methods in operations research". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.3 (2015), pp. 1–43.

[41] Zbigniew Michalewicz. *How to solve it : modern heuristics*. Berlin New York: Springer, 2004. ISBN: 3540224947.

[42] Radhika Nagpal, Howard Shrobe, and Jonathan Bachrach. "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network". In: *Information Processing in Sensor Networks*. Ed. by Feng Zhao and Leonidas Guibas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 333–348. ISBN: 978-3-540-36978-3.

[43] Yurii E. Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. Vol. 13. Siam studies in applied mathematics. SIAM, 1994, pp. I–VII, 1–405. ISBN: 978-0-89871-319-0.

[44] D. Niculescu and B. Nath. "Ad hoc positioning system (APS)". In: *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*. Vol. 5. 2001, 2926–2931 vol.5.

[45] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[46] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. "A review on random forest: An ensemble classifier". In: *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*. Springer. 2019, pp. 758–763.

[47] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[48] Riccardo Poli and Mario Graff. "There Is a Free Lunch for Hyper-Heuristics, Genetic Programming and Computer Scientists". In: *Genetic Programming*. Ed. by Leonardo Vanneschi et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 195–207.

[49] Priyanka and Dharmender Kumar. "Decision tree classifier: a detailed survey". In: *International Journal of Information and Decision Sciences* 12.3 (2020), pp. 246–269.

[50] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi. "GSA: A gravitational search algorithm". In: *Information Sciences* 179.13 (2009), 2232–2248.

[51] Stuart Jonathan Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Third. Pearson Education, 2016. ISBN: 9781292153964.

[52] Patricia Ryser-Welch and Julian F Miller. "A review of hyper-heuristic frameworks". In: *Proceedings of the evo20 workshop, aisb*. Vol. 2014. 2014.

[53] Yi Shang et al. "Localization from mere connectivity". In: *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*. MobiHoc '03. Annapolis, Maryland, USA: Association for Computing Machinery, 2003, 201–212. ISBN: 1581136846.

[54] Slobodan Simić and S. Shankar Sastry. *Distributed Localization in Wireless Ad Hoc Networks*. Tech. rep. UCB/ERL M02/26. 2002.

[55] Rainer Storn and Kenneth Price. "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of global optimization* 11.4 (1997), p. 341.

[56] E.-G. Talbi. "A Taxonomy of Hybrid Metaheuristics". In: *Journal of Heuristics* 8.5 (Sept. 2002), 541–564. ISSN: 1381-1231.

[57] Antony Unwin and Kim Kleinman. "The iris data set: In search of the source of virginica". In: *Significance* 18.6 (2021), pp. 26–29.

[58] Abraham Wald. *Sequential Analysis*. 1st. John Wiley and Sons, 1947.

[59] Darrell Whitley. "A genetic algorithm tutorial". In: *Statistics and Computing* 4.2 (1994).

[60] D. H. Wolpert and W. G. Macready. "No Free Lunch Theorems for Optimization". In: *IEEE Transactions on Evolutionary Computation* 1.67 (1997), pp. 67–82.