

Automatic design of metaheuristic algorithms with learning

FONDECYT Project 1230365

Elizabeth Montero

Departamento de Informática, Universidad Técnica Federico Santa María
Valparaíso, Chile.

Journées Scientifiques Inria Chile 2024
December 04, 2024



Agenda

1 Context

- Problem solving
- Metaheuristic algorithms
- Tuning processes
- Automatic selection approaches

2 Work in progress

3 Results

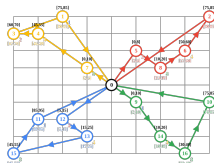
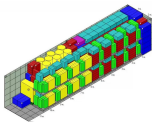
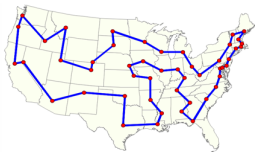
4 Conclusions & Future work

Problem Solving

- Artificial Intelligence: A Modern Approach, 4th US ed., Russell & Norving

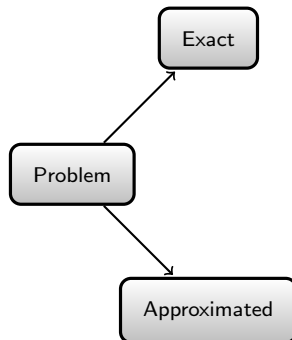


- Solving Problems by Searching - Combinatorial optimization
- Search in Complex Environments - NP-hard problems



Metaheuristic algorithms

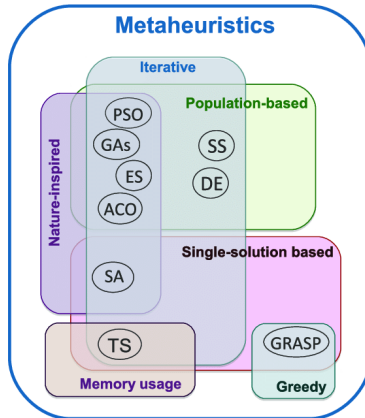
- Local search strategies/aproximatte
- Local optima solutions
- Computationally efficient
- Stochastic nature



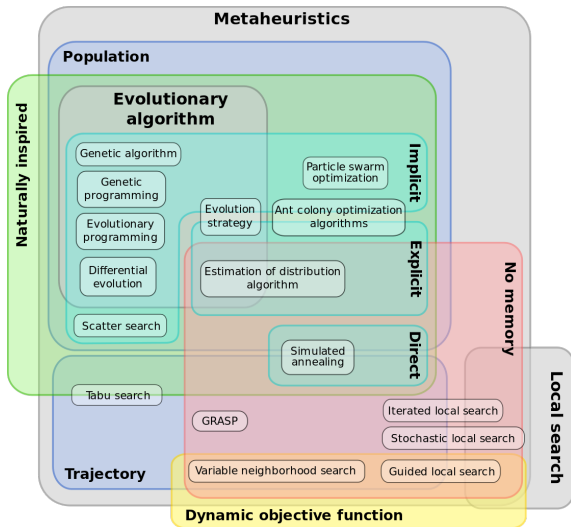
References

- Zbigniew Michalewicz and David B. Fogel, *How to Solve It: Modern Heuristics*, Enlarged 2nd Edition, 2004.
- Emile Aarts and Jan K. Lenstra, *Local Search in Combinatorial Optimization*, 2003-08-03.

Classification



Classification



Difficulties with metaheuristics

- Definition of critical components

- Neighborhood structures → local search
- Crossover and mutation operators
- Disturbation/Shacking/Restart procedures

categorical parameters

- Setting of critical parameter values

- Temperature schedule
- Crossover and mutation probabilities
- Population sizes.

numerical parameters



- Different instance-features (size, distribution)
- Stochastic nature
- Combinatorial problem itself.

Difficulties with metaheuristics

- Definition of critical components
 - Neighborhood structures → local search
 - Crossover and mutation operators
 - Disturbation/Shacking/Restart procedures

categorical parameters

- Setting of critical parameter values
 - Temperature schedule
 - Crossover and mutation probabilities
 - Population sizes.

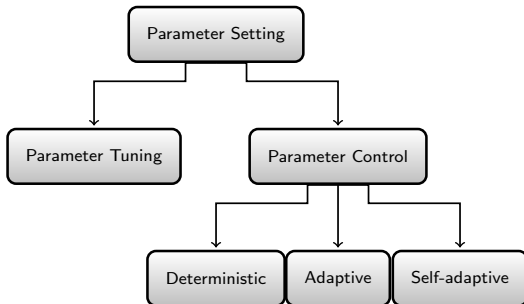
numerical parameters



- Different instance-features (size, distribution)
- Stochastic nature
- **Combinatorial problem itself.**

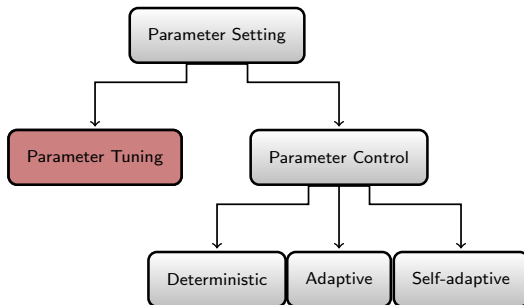
Parameter setting

- Approaches: [ES08]
 - Tuning
 - Control
 - Deterministic
 - Adaptive
 - Self-adaptive



Parameter setting

- Approaches: [ES08]
 - Tuning
 - Control
 - Deterministic
 - Adaptive
 - Self-adaptive

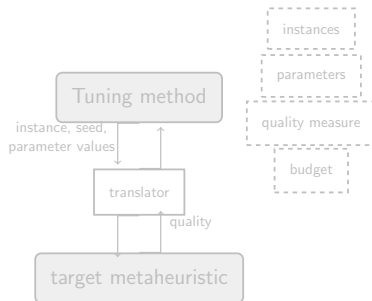


Tuning processes

Several tuning methods have been proposed in literature [MRN14].

Four categories [ES11]:

- **Sampling methods:** Calibra [ADL06] and Myers and Hancock in [MH01].
- **Model-based methods:** SPO [BBLP05] and SMAC [HHLB11]
- **Screening methods:** F-Race [BSPV02] and I-Race [BBS07] its extension.
- **Meta-evolutionary/Search-based methods:** ParamILS [HHLBS09] and Evoca [RM13].

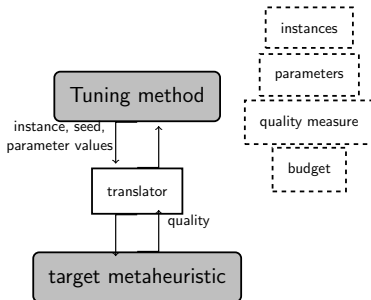


Tuning processes

Several tuning methods have been proposed in literature [MRN14].

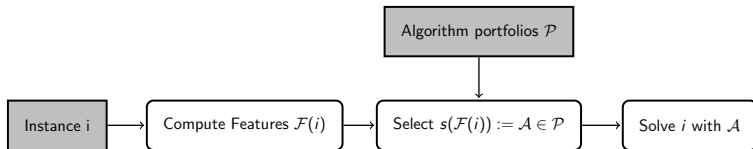
Four categories [ES11]:

- **Sampling methods:** Calibra [ADL06] and Myers and Hancock in [MH01].
- **Model-based methods:** SPO [BBLP05] and SMAC [HHLB11]
- **Screening methods:** F-Race [BSPV02] and I-Race [BBS07] its extension.
- **Meta-evolutionary/Search-based methods:** ParamILS [HHLBS09] and Evoca [RM13].



Automatic selection approaches

- Non-Free Lunch Theorem [DJ05].
- *per-instance and per-set algorithm selection, algorithm configuration, algorithm portfolios and hyper-heuristic methods*
- SATzilla [XHHLB08] and AutoFolio [LHHS15], implement combinations of algorithms and per-instance selection.



- Feature extraction:
 - Satisfiability (SAT) problems [XHHLB08, XHHLB12].
 - Traveling Salesman problem [KKB⁺18, KKHT15]
 - Mixed integer programming problem instances [HXHLB14].
 - Traveling thief problem [WLM⁺18].

Framework

- ① Instance feature extraction,
 - Extraction of problem instances features.
- ② Parameter tuning,
 - Determine the best parameter values for each problem instance.
- ③ Training of machine learning algorithms, and
 - Input: Features (1) + parameters (2)
- ④ Parameter prediction for novel instances.
- ⑤ Explainability analysis

- Problem: Vehicle Routing Problems.
 - Capacitated Vehicle Routing Problem with Time Windows.
- Metaheuristic: Genetic Algorithm
 - Hybrid-Genetic Search (HGS) [VCG⁺12] as implemented in [Vid22] framework: multi-depot VRP, periodic VRP and multi-depot periodic VRP with capacitated vehicles and constrained route duration.
 - In [VCGP13], was extended to incorporate a large class of time-constrained VRPs: VRP with time windows, multi-depot VRP with time windows and vehicle-site dependencies VRP with time windows.
- Problem instances:
 - Well-known CVRPTW Homberger set of 300 instances [Geh99].
Client distribution = {C1: Densely Clustered, C2: Sparsely Clustered, R1: Random (200), R2: Random (1000), RC1: Semi Clustered (200), RC2: Semi Clustered (1000)}

Work in progress

First Step - Instance feature extraction

- Classic metrics [RKM16]:
 - 1 Number of clients
 - 2 Distance from centroid to depot
 - 3 Average distance between depot and clients
 - 4 Average distance between centroid and clients
 - 5 Coefficient of Variation (CV) of distance between centroid and clients
 - 6 Ratio of average client demand to capacity
 - 7 Ratio of the standard deviation of client demands to capacity
 - 8 Ratio of clients to vehicles
 - 9 Average of distances to the nearest neighbor
 - 10 CV of distances to the nearest neighbor

Work in progress

First Step - Instance feature extraction

- Clustering metrics using dynamicOPTICS [ABKS99].
 - 1 Optimal min_samples value: Measures clusterability
 - 2 Cluster Ratio: Number of clusters
 - 3 Outlier Ratio: Number of non-clustered clients
 - 4 Average clients per cluster: Measures cluster size (number of clients)
 - 5 CV clients per cluster: Cluster size heterogeneity
 - 6 Average of intra-cluster distances: Cluster density
 - 7 Average of inter-cluster distances: Cluster spread

Work in progress

First Step - Instance feature extraction

- New time windows features.
 - 1 Ratio of the highest number of overlapping windows to total: Measures tightest time interval.
 - 2 Ratio of average number of overlapping windows to total: Measures average time tightness.
 - 3 Ratio of average window length to the longest window: Measures normalized time window length.
 - 4 Ratio of the standard deviation of window length to the longest time window: Measures normalized time window heterogeneity.

Work in progress

Second Step - Parameter tuning

- Parameter Iterated Local Search (ParamILS) [HHS07].
- State-of-the-art VRP solver: Hybrid-Genetic Search (HGS) [VCG⁺12].
- *PyVRP* implementation [Vid22].
- Three scenarios: 10[s], 30[s] and 60[s]. 10 runs per scenario per instance.
- Training (240/300) Homberger instances (80%)
- Test (60/300) (20%)

Work in progress

Second Step - Parameter tuning

- μ : **Minimum population size**; lower bound of the number of solutions that are allowed to exist in either the feasible or unfeasible subpopulation.
- λ : **Generation size**; the number of solutions over μ allowed to exist. λ solutions are eliminated through a survivor selection phase once the maximum number is reached.
- n_{Elite} : **Number of elite solutions** considered in the fitness calculation.
- $n_{Closest}$: **Number of closest solutions** considered in the diversity-contribution measure of the fitness solution.
- ξ^{REF} : **Target proportion of feasible individuals to total population size.**

Parameter	Values				
μ	5	15	25	35	45
λ	10	20	40	70	100
n_{Elite}	0.1	0.2	0.4	0.6	0.8
$n_{Closest}$	0.1	0.2	0.4	0.6	0.8
ξ^{REF}	0.1	0.2	0.4	0.6	0.8

Work in progress

Second Step - Parameter tuning

- μ : **Minimum population size**; lower bound of the number of solutions that are allowed to exist in either the feasible or unfeasible subpopulation.
- λ : **Generation size**; the number of solutions over μ allowed to exist. λ solutions are eliminated through a survivor selection phase once the maximum number is reached.
- n_{Elite} : **Number of elite solutions** considered in the fitness calculation.
- $n_{Closest}$: **Number of closest solutions** considered in the diversity-contribution measure of the fitness solution.
- ξ^{REF} : **Target proportion of feasible individuals to total population size.**

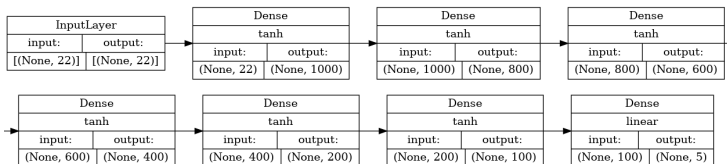
Parameter	Values				
μ	5	15	25	35	45
λ	10	20	40	70	100
n_{Elite}	0.1	0.2	0.4	0.6	0.8
$n_{Closest}$	0.1	0.2	0.4	0.6	0.8
ξ^{REF}	0.1	0.2	0.4	0.6	0.8

Work in progress

Third Step - Training of machine learning algorithms

- Algorithms:

- K Nearest Neighbours [FH89, KE11]: $K \in \{1, 3, 5, 7, 9, 11\}$.
- Artificial Neural Networks: 22-neuron input layer, six hidden layers, and a five-neuron output layer.



- Loss function: *Mean Squared Error*
- Optimizer: *AdamW* with $\alpha = 0.0002$
- Epochs: 1000
- Batch size: 30
- Patience: 200
- Activation function for all layers: *tanh*
- Activation function for the output layer: *linear*

Work in progress

Fourth Step - Performance prediction

- A. Parameter values predicted by KNN with $K \in \{1, 3, 5, 7, 9, 11\}$
- B. Parameter values predicted by the artificial neural network.
- C. Base parameter values for the HGS metaheuristic (as defined in [Vid22]).
- D. Parameter values samples from a random distribution.

Work in progress

Fourth Step - Performance prediction

Parameter Set	Instances	10 [s]	30 [s]	60 [s]	Avg. 10 [s]	Avg. 30 [s]	Avg. 60 [s]
KNN1	C1	14728	14572	14543	15645	15231	15228*
	C2	11100	10532	10505			
	R1	26005	25216	25529*			
	R2	14850	14377	14144			
	RC1	16278	16076	16048			
	RC2	10908	10610	10597			
KNN3	C1	14755	14579	14547	15768	15630	15031
	C2	11202	10550	10448			
	R1	26515	27641	24542			
	R2	14904	14349	14123			
	RC1	16307	16083	16051			
	RC2	10925	10576	10473			
KNN5	C1	14732	14575	14535	15648	15620	15094
	C2	11061*	10533	10418			
	R1	26054*	27620	25028			
	R2	14816	14335	14128			
	RC1	16303	16093	16031			
	RC2	10920	10563	10422			
KNN7	C1	14727	14585	14531	15882	15631*	15121
	C2	11018	10522	10440			
	R1	27485	27654*	25092			
	R2	14875	14348	14175			
	RC1	16295	16108	16023			
	RC2	10892	10566	10466			
KNN9	C1	14763	14588	14512	15924	15233	15016
	C2	11205	10529	10432			
	R1	27493	25242	24576			
	R2	14868	14379	14140			
	RC1	16312	16100	15982			
	RC2	10905	10559	10452			

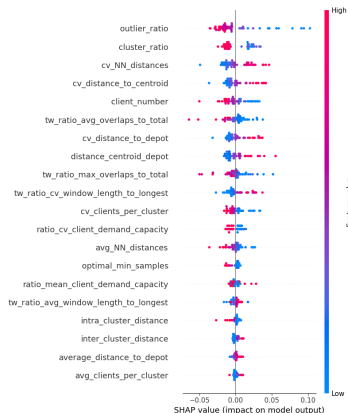
Work in progress

Fourth Step - Performance prediction

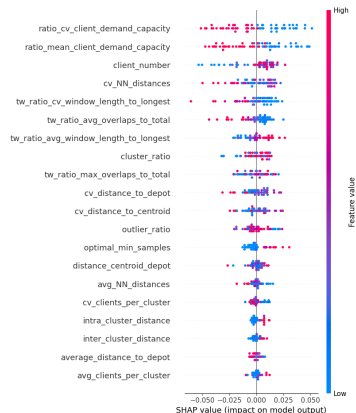
Parameter Set	Instances	10 [s]	30 [s]	60 [s]	Avg. 10 [s]	Avg. 30 [s]	Avg. 60 [s]
KNN11	C1	14771	14595	14523	16017*	15249	14938
	C2	11176	10536	10425			
	R1	27951	25254	24082			
	R2	15042	14454	14163			
	RC1	16349	16113	16003			
	RC2	10815	10544	10434			
Neural Network	C1	14742	14591	14526	15755	15414	14964
	C2	11149	10560	10431			
	R1	26482	26196	24064			
	R2	14854	14394	14305			
	RC1	16337	16125	16021			
	RC2	10967	10618	10436			
Base	C1	14934*	14686	14608	16005	15366	14920
	C2	11761	10876*	10514			
	R1	26336	24993	23344			
	R2	15220	14638	14363			
	RC1	16523*	16240	16144			
	RC2	11255	10762	10548			
Random	C1	14886	14705*	14617*	15769	15392	14988
	C2	11716	10952*	10587*			
	R1	24951	24664	23415			
	R2	15271*	14667*	14493*			
	RC1	16456	16299*	16173*			
	RC2	11333*	11065*	10641*			

Work in progress

Fifth Step - Explainability analysis (1/3)



(a) Generation size (λ)

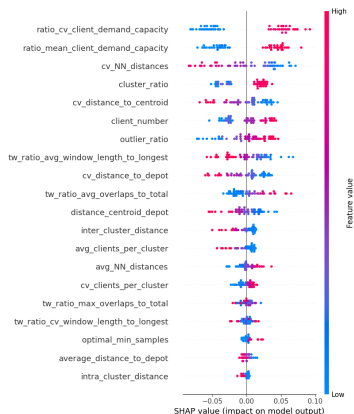


(b) Number of close solutions ($n_{Closest}$)

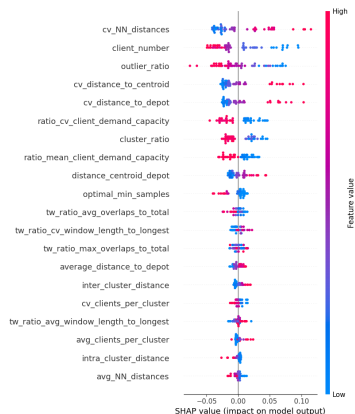
Figure: SHAP values.

Work in progress

Fifth Step - Explainability analysis (2/3)



(a) Number of elite solutions (n_{Elite})



(b) Population size (μ)

Figure: SHAP values.

Work in progress

Fifth Step - Explainability analysis (3/3)



Figure: SHAP values for the target proportion of feasible solutions (ξ^{REF})

Conclusions & Future work

- Parameter prediction was successful, reducing the average travel time in Homberger instances compared to the base parameters proposed in [Vid22].
- More accurate predictions would require instances with a higher level of diversity.
- Client clustering, client number, and vehicle capacity are the most relevant features when predicting HGS' optimal parameterization.
- Time windows-related features were considered the top ten relevant ones for 4/5 parameters of HGS.

Future work we are interested in:

- (A) Construct a wider spanning set of CVRPTW instances using an evolutionary algorithm as mentioned in [SMvH11], to maximize the variance of extracted features in the instance population.
- (B) Apply this parameter prediction pipeline to tune the parameters of a metaheuristic for dynamically changing instances.

Conclusions & Future work

- Parameter prediction was successful, reducing the average travel time in Homberger instances compared to the base parameters proposed in [Vid22].
- More accurate predictions would require instances with a higher level of diversity.
- Client clustering, client number, and vehicle capacity are the most relevant features when predicting HGS' optimal parameterization.
- Time windows-related features were considered the top ten relevant ones for 4/5 parameters of HGS.

Future work we are interested in:

- (A) Construct a wider spanning set of CVRPTW instances using an evolutionary algorithm as mentioned in [SMvH11], to maximize the variance of extracted features in the instance population.
- (B) Apply this parameter prediction pipeline to tune the parameters of a metaheuristic for dynamically changing instances.

Extended future work

- Search of algorithms for different classes of problems can be seen as a MOP.
- MO tuning methods appear as a clear way to perform the best identification of these configurations.
- Use these non-dominated parameter configurations to perform transitions between different problem instances classes and algorithms.

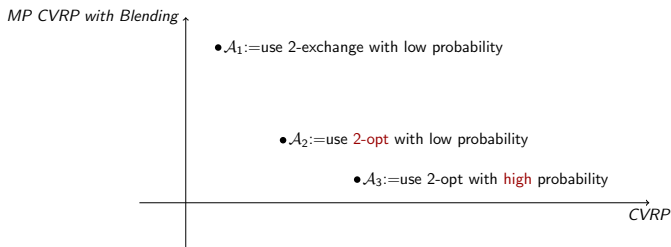
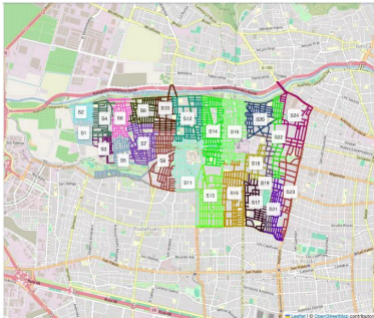


Figure: Algorithm Pareto Front

- Design of collaborative collection of household solid waste in urban areas.
- INRIA Internship: Ricardo Barriga
Dissimilarity clustering for stochastic optimization and machine learning.



References I



Mihael Ankerst, Markus Breunig, Peer Kröger, and Joerg Sander.
Optics: Ordering points to identify the clustering structure.
volume 28, pages 49–60, 06 1999.



B. Adenso-Diaz and M. Laguna.
Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search.
Operations Research, 54(1):99–114, 2006.



T. Bartz-Beielstein, C. W. G. Lasarczyk, and M. Preuss.
Sequential Parameter Optimization.
In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, pages 773–780, 2005.



P. Balaprakash, M. Birattari, and T. Stützle.
Improvement strategies for the f-race algorithm: Sampling design and iterative refinement.
In Thomas Bartz-Beielstein, María José Blesa Aguilera, Christian Blum, Boris Naujoks, Andrea Roli, Günter Rudolph, and Michael Sampels, editors, *Hybrid Metaheuristics: 4th International Workshop, HM 2007, Dortmund, Germany, October 8-9, 2007. Proceedings*, pages 108–122, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.



Mauro Birattari, Thomas Stützle, Luis Paquete, and Klaus Varrentrapp.
A racing algorithm for configuring metaheuristics.
In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '02*, pages 11–18, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.



Whitley D. and Watson J.P.
Complexity Theory and the No Free Lunch Theorem.
Springer, Boston, MA, 2005.

References II



A.E. Eiben and M.C. Schut.

Advances in Metaheuristics for Hard Optimization, chapter New Ways to Calibrate Evolutionary Algorithms, pages 153–177.

Natural Computing. Springer, 2008.



A.E. Eiben and S.K. Smit.

Parameter tuning for configuring and analyzing evolutionary algorithms.

Swarm and Evolutionary Computation, 1(1):19–31, 2011.



Evelyn Fix and Joseph L. Hodges.

Discriminatory analysis - nonparametric discrimination: Consistency properties.

International Statistical Review, 57:238, 1989.



Hermann Gehring.

A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. 1999.



Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown.

Sequential model-based optimization for general algorithm configuration.

In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, LION'05, pages 507–523, Berlin, Heidelberg, 2011. Springer-Verlag.



Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, and Thomas Stützle.

Paramils: An automatic algorithm configuration framework.

J. Artif. Int. Res., 36(1):267–306, September 2009.

References III



Frank Hutter, Holger H. Hoos, and Thomas Stützle.
Automatic algorithm configuration based on local search.
Proceedings of the twenty-second conference on artificial intelligence, 2007.



Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown.
Algorithm runtime prediction: Methods & evaluation.
Artif. Intell., 206:79–111, 2014.



Nikolaos Kouiroukidis and Georgios Evangelidis.
The effects of dimensionality curse in high dimensional knn search.
In 2011 15th Panhellenic Conference on Informatics, pages 41–45, 2011.



P. Kerschke, L. Kotthoff, J. Bossek, H. H. Hoos, and H. Trautmann.
Leveraging tsp solver complementarity through machine learning.
Evolutionary Computation, 26(4):597–620, 2018.



Lars Kotthoff, Pascal Kerschke, Holger Hoos, and Heike Trautmann.
Improving the state of the art in inexact tsp solving using per-instance algorithm selection.
In Clarisse Dhaenens, Laetitia Jourdan, and Marie-Éléonore Marmion, editors, Learning and Intelligent Optimization, pages 202–217, Cham, 2015. Springer International Publishing.



Marius Lindauer, Holger H. Hoos, Frank Hutter, and Torsten Schaub.
Autofolio: An automatically configured algorithm selector.
Journal of Artificial Intelligence Research, 53, 2015.

References IV



R. Myers and E. R. Hancock.

Empirical Modelling of Genetic Algorithms.
Evolutionary Computation, 9(4):461–493, 2001.



Elizabeth Montero, María-Cristina Riff, and Bertrand Neveu.

A beginner's guide to tuning methods.
Appl. Soft Comput., 17:39–51, April 2014.



Jussi Rasku, Tommi J. Kärkkäinen, and Nysret Musliu.

Feature extractors for describing vehicle routing problem instances.
In *Student Conference on Operational Research*, 2016.



María-Cristina Riff and Elizabeth Montero.

A New Algorithm for Reducing Metaheuristic Design Effort.
In *IEEE Congress on Evolutionary Computation (CEC 2013)*, pages 3283–3290, Cancún, México, June 2013.



Kate Smith-Miles and Jano van Hemert.

Discovering the suitability of optimisation algorithms by learning from evolved instances.
Annals of Mathematics and Artificial Intelligence, 04 2011.



Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, and Walter Rei.

A hybrid genetic algorithm for multidepot and periodic vehicle routing problems.
Operations Research, 60:611–624, 06 2012.



Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins.

A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows.

Computers & Operations Research, 2013.



Thibaut Vidal.

Hybrid genetic search for the cvrp: Open-source implementation and swap* neighborhood.

Computers & Operations Research, 140:105643, 2022.



Markus Wagner, Marius Lindauer, Mustafa Mundefinedsundefinedr, Samadhi Nallaperuma, and Frank Hutter.

A case study of algorithm selection for the traveling thief problem.

Journal of Heuristics, 24(3):295–320, 2018.



Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown.

Satzilla: Portfolio-based algorithm selection for sat.

J. Artif. Int. Res., 32(1):565–606, 2008.



Lin Xu, Frank Hutter, Holger Hoos, and Kevin Leyton-Brown.

Evaluating component solver contributions to portfolio-based algorithm selectors.

In Alessandro Cimatti and Roberto Sebastiani, editors, *Theory and Applications of Satisfiability Testing – SAT 2012*, pages 228–241, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

Automatic design of metaheuristic algorithms with learning

FONDECYT Project 1230365

Elizabeth Montero

Departamento de Informática, Universidad Técnica Federico Santa María
Valparaíso, Chile.

Journées Scientifiques Inria Chile 2024
December 04, 2024

