citation	domain	features	predict what	predict how	predict when	portfolio
[Langley; Langley]	search	past performance	algorithm	hand-crafted and learned rules	offline and online	dynamic
[Carbonell et al.]	planning	problem domain features, search statistics	control rules	explanation-based rule construction	online	dynamic
[Gratch and DeJong]	planning	problem domain features, search statistics	control rules	probabilistic rule construction	online	dynamic
[Smith and Setliff]	software design	features of abstract representation	algorithms and data structures	simulated annealing	offline	static
[Aha]	machine learning	instance features	algorithm	learned rules	offline	static
[Brodley]	machine learning	instance and algorithm features	algorithm	hand-crafted rules	offline	static
[Kamel et al.]	differential equations	past performance, instance features	algorithm	hand-crafted rules	offline	static
[Minton; Minton; Minton]	constraints	runtime performance	algorithm	hand-crafted and learned rules	offline	dynamic
[Cahill]	software design	instance features	algorithms and data structures	frame-based knowledge base	offline	static
[Tsang et al.]	constraints	instance features	-	-	-	static
[Brewer]	software design	runtime performance	algorithms, data structures and their parameters	statistical model	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Weerawarana et al.; Joshi et al.]	differential equations	instance features	runtime performance	Bayesian belief propagation, neural nets	offline	static
[Borrett et al.]	constraints	search statistics	switch algorithm?	hand-crafted rules	online	static, static order
[Allen and Minton]	SAT, constraints	probing	runtime performance	hand-crafted rules	online	static
[Sakkout et al.]	constraints	search statistics	switch algorithm?	hand-crafted rules	online	static
[Huberman et al.]	graph colouring	past performance	resource allocation	statistical model	offline	static
[Gomes and Selman; Gomes and Selman]	constraints	problem size and past performance	algorithm	statistical model	offline	static
[Cook and Varnell]	parallel search	probing	set of search strategies	decision trees, Bayesian classifier, nearest neighbour, neural net	online	static
[Fink; Fink]	planning	past performance	resource allocation	statistical model, regression	offline	static
[Lobjois and Lemaître]	branch and bound	probing	runtime performance	hand-crafted rules	online	static
[Caseau et al.]	vehicle routing problem	runtime performance	algorithm	genetic algorithms	offline	static
[Howe et al.]	planning	instance features	resource allocation	linear regression	offline	static
[Terashima-Marín et al.]	scheduling	instance and search features	algorithm	genetic algorithms	offline	dynamic

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Wilson et al.]	software design	instance features	data structures	nearest neighbour	offline	static
[Beck and Fox]	job shop scheduling	instance feature changes during search	algorithm scheduling policy	hand-crafted rules	online	static
[Brazdil and Soares]	classification	past performance	ranking	distribution model	offline	static
[Lagoudakis and Littman]	order selection, sorting	instance features	remaining cost for each sub-problem	MDP	online	static
[Sillito]	constraints	probing	cost of solving problem	statistical model	offline	static
[Pfahringer et al.]	classification	instance features, probing	algorithm	9 different classifiers	offline	static
[Fukunaga]	TSP	past performance	resource allocation	performance simulation for different allocations	offline	static
[Soares and Brazdil]	machine learning	instance features	ranking	nearest neighbour	offline	static
[Gomes and Selman]	constraints, mixed integer programming	past performance	algorithm	statistical model	offline	dynamic
[Cowling et al.]	scheduling	instance features	algorithm	hand-crafted rules, weights	online	static
[Epstein and Freuder; Epstein et al.; Epstein et al.; Epstein and Petrovic]	constraints	variable characteristics	algorithm	weights, hand-crafted rules	offline and online	dynamic
[Lagoudakis and Littman]	DPLL branching rules	instance features	remaining cost for each sub-problem	MDP	online	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Nareyek]	optimisation	search statistics	expected utility of algorithm	reinforcement learning	offline and online	static
[Horvitz et al.]	constraints	instance and instance generator features, search statistics	runtime performance, restart parameters	Bayesian model	offline and online	static
[Borrett and Tsang]	constraints	instance features, search statistics	redundant constraints to add	hand-crafted rules	offline	-
[Little et al.]	logic puzzles	instance graph features	instance model transformations for runtime performance	nearest neighbour	offline	-
[Petrovic and Qu]	scheduling	instance features	algorithm	case-based reasoning	offline	static
[Leyton-Brown et al.]	winner determination problem	instance features	instance hardness	several forms of regression	offline	static
[Fukunaga; Fukunaga]	SAT	variable characteristics	algorithm	genetic algorithms	offline	dynamic
[Yu et al.; Yu et al.; Yu and Rauchwerger]	parallel reduction algorithms	instance features	algorithm	decision trees, general linear regression	offline and online	static
[Ruan et al.]	SAT	instance features	restart policy	dynamic programming	offline	static
[Vrakas et al.]	planning	instance features	parameters	classification association rules	offline	dynamic
[Guo]	sorting, probabilistic inference	instance features	algorithm	decision tree, naïve Bayes, Bayesian network, meta-learning	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Watson]	job shop scheduling	instance features, search statistics	local search algorithm	statistical model	offline and online	static
[Brazdil et al.]	machine learning	instance features	ranking	nearest neighbour	offline	static
[Gebruers et al.]	bid evaluation problem	instance and instance graph features	solution method	nearest neighbour	offline	static
[Guerri and Milano]	bid evaluation problem	instance and instance graph features	solution method, algorithm	decision trees	offline	static
[Beck and Freuder]	scheduling	probing	algorithm	hand-crafted rules	offline	static
[Nudelman et al.; Xu et al.; Xu et al.]	SAT	instance features, probing	runtime performance	ridge regression, lasso regression, SVMs, Gaussian processes	offline	static
[Carchrae and Beck; Carchrae and Beck]	job shop scheduling	probing, search statistics	length of exploration phase, switch algorithm?	Bayesian classifier, reinforcement learning	offline and online	static
[Soares et al.]	machine learning	instance features	ranking of SVM kernel widths	nearest neighbour	offline	static
[Guo and Hsu]	most probable explanation problem	instance features	algorithm	decision trees, naïve Bayes rules, Bayes networks, meta-learning techniques	offline	static
[Gagliolo et al.]	search problems	past performance	resource allocation	linear model	online	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Prudêncio and Ludermir]	machine learning	instance features	ranking	decision trees and neural networks	offline	static
[Demmel et al.]	linear algebra	instance features	algorithm	multivariate Bayesian decision rule	offline	static
[Gebruers et al.]	constraints	instance features	problem model, solution strategy	nearest neighbour, decision trees, statistical model	offline	static
[Petrik]	SAT	past performance	resource allocation	analytic model, MDP	offline and online	static
[Cicirello and Smith]	scheduling	past performance	algorithm	reinforcement learning	online	static
[Gagliolo and Schmidhuber]	-	past performance	resource allocation	neural nets	online	static
[Armstrong et al.]	procedure calls	runtime performance	switch algorithm?	reinforcement learning	online	static
[Gagliolo and Schmidhuber]	SAT, auction winner determination problem	past performance	resource allocation	reinforcement learning	online	static
[Roberts and Howe]	planning	instance features	resource allocation	decision trees	offline	static
[Hough and Williams]	optimisation	instance, algorithm and environment features	algorithm	ensembles of decision trees, SVMs	offline	static
[Bhowmick et al.]	linear systems	instance features	algorithm	boosting, alternating decision trees	offline	static
[Hutter et al.]	stochastic local search	instance features	runtime performance	ridge regression	offline	dynamic
[Sayag et al.]	SAT	past performance	resource allocation	static model, probabilistic model	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Ali and Smith]	classification	instance features	algorithm	decision rules	offline	static
[Cavazos and O'Boyle]	software design	instance features	algorithm	logistic regression	offline	static
[Xu et al.]	SAT	instance features	satisfiability and runtime performance	sparse multinomial logistic regression, ridge regression	offline	static
[Pulina and Tacchella; Pulina and Tacchella; Pulina and Tacchella]	QBF	instance features	resource allocation	decision trees, decision rules, logistic regression, nearest neighbour	offline and online	static
[Samulowitz and Memisevic]	QBF	instance features	algorithm, confidence values	multinomial logistic regression	offline and online	static
[Wu and van Beek]	scheduling	-	portfolio	case-based reasoning	offline	dynamic
[Streeter et al.]	planning	past performance	resource allocation	statistical model	offline and online	static
[Wang and Tropper]	simulation algorithms	past performance	control parameter	reinforcement learning	online	static
[Roberts and Howe; Roberts et al.]	planning	instance features	runtime, probability of success	32 different algorithms	offline	static
[de la Rosa et al.; de la Rosa et al.; de la Rosa et al.]	planning	instance features	algorithm	case-based reasoning	online	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Steer et al.]	-	fitness landscape features	algorithm	-	offline	static
[Streeter and Smith]	SAT, integer programming, planning	instance features	resource allocation	statistical model	offline and online	static
[O'Mahony et al.; Bridge et al.]	constraints	instance features, probing	resource allocation	nearest neighbour	offline	static
[Kuefler and Chen]	linear systems	instance features, search statistics	algorithm	reinforcement learning	online	static
[Wei et al.]	SAT	search statistics	algorithm	hand-crafted rules	online	static
[Gagliolo and Schmidhuber]	SAT	past performance	resource allocation	reinforcement learning	online	static
[Smith-Miles]	Quadratic Assignment Problem	instance features, probing	algorithm, runtime performance	neural networks and self-organising maps	offline	static
[Stergiou; Stergiou; Paparrizou and Stergiou]	constraints	search statistics	propagation method	clustering	online	static
[de la Rosa et al.; de la Rosa et al.]	planning	instance features	algorithm	decision tree	online	static
[Nikolić et al.]	SAT	instance features	search strategy	nearest neighbour	offline	static
[Stamatatos and Stergiou]	constraints	probing	propagation method	clustering	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Arbelaez et al.; Arbelaez et al.]	constraints	instance features, search statistics	search strategy	SVM	online	static
[Haim and Walsh]	SAT	instance features	restart strategy and satisfiability	ridge regression, logistic regression	offline	static
[Bhowmick et al.]	linear systems	instance features	algorithm	nearest-neighbour, alternating decision trees, naïve Bayes, SVM	offline	static
[Gerevini et al.]	planning	past performance	macro actions, resource allocation	performance simulations for different allocations	offline	static
[Xu et al.]	constraints	instance features	algorithm	reinforcement learning	online	static
[Bougeret et al.]	SAT	past performance	resource allocation	static model	offline	static
[Smith-Miles et al.]	scheduling	instance features	algorithm	decision tree, neural networks, self-organizing maps	offline	static
[Leite et al.]	machine learning	past performance, probing	ranking of classification algorithms	statistical model	offline and online	static
[Silverthorn and Miikkulainen]	SAT	past performance	runtime performance	latent class models	offline	static
[Stern et al.]	QBF, combinatorial auctions	instance and algorithm features	algorithm	Bayesian model	offline and online	static
[Garrido and Riff]	dynamic vehicle routing problem	runtime performance	combination of low-level heuristics	genetic algorithms	online	dynamic

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Domshlak et al.]	planning	state variables	algorithm	naïve Bayes classifier	online	static
[Kadioglu et al.]	SAT, mixed integer programming, set covering	instance features	algorithm	clustering	offline	dynamic
[Gent et al.]	constraints	instance features, probing	algorithm	decision trees	offline	static
[Gent et al.]	software design	instance features	implementation	19 different classifiers	offline	static
[Kotthoff et al.]	constraints	instance features, probing	algorithm	ensembles of classifiers	offline	static
[Ewald et al.]	simulation algorithms	past performance	portfolio	genetic algorithms	offline	dynamic
[Elsayed and Michel; Elsayed and Michel]	constraints	instance features	search strategy	hand-crafted rules	online	dynamic
[Valenzano et al.]	search problems	-	algorithm	round-robin	online	static
[Leite and Brazdil]	classification	past performance	ranking	statistical model	offline	static
[Aiguzhinov et al.]	classification	past performance	ranking	naïve Bayes	offline	static
[Kanda et al.; Kanda et al.]	TSP	instance features	algorithms	nearest neighbour, decision tree, SVM, naïve Bayes	offline	static
[Peng et al.]	numerical optimisation	past performance	resource allocation	optimisation	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Graff and Poli]	program induction	fitness function	runtime performance	regression	offline	static
[Fialho et al.]	genetic algorithms	past performance	algorithm	aggregation	online	static
[Tolpin and Shimony]	constraints	search statistics	algorithm	hand-crafted rules	online	static
[Malitsky et al.]	SAT	instance features	algorithm	nearest neighbour	offline	static
[Kadioglu et al.]	SAT	instance features	resource allocation	nearest neighbour	offline	static
[Kroer and Malitsky]	SAT, constraints	instance features	algorithm	clustering	offline	dynamic
[Kotthoff et al.; Kotthoff et al.]	SAT, QBF, constraints	instance features, probing	algorithm, runtime performance, ranking	31 different machine learning algorithms	offline	static
[Gagliolo and Schmidhuber; Gagliolo and Schmidhuber]	SAT, QBF, constraints	past performance	resource allocation	reinforcement learning	online	static
[Gebser et al.]	Answer Set Programming	instance features, probing	runtime performance	SVM	offline	static
[Xu et al.]	MIP	instance features, probing	algorithm	random forests	offline	dynamic
[Maturana et al.]	evolutionary algorithms	past performance	algorithm	statistical models	online	static
[Helmert et al.]	planning	past performance	resource allocation	statistical model	offline	static
[Kiziltan et al.]	constraints	instance features	resource allocation	8 classification algorithms, ridge regression	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Smith-Miles and Hemert]	TSP	instance features	algorithm	self-organizing map, decision tree, neural network	offline	static
[Prudêncio et al.]	machine learning	instance features	ranking	nearest neighbour	offline	static
[Kotthoff]	SAT, QBF, constraints	instance features, probing	algorithm	5 regression algorithms, 2 classification algorithms	offline	static
[Yun and Epstein]	constraints	instance features	portfolio	case-based reasoning, hand-crafted rules	offline	dynamic
[Hurley and OSullivan]	SAT	instance features	ranking	case-based reasoning with voting	offline	static
[Shukla et al.]	inventory routing problem	past performance	portfolio	statistical model	offline	static
[Malitsky et al.]	SAT	past performance	resource allocation	nearest neighbour	offline and online	static
[Bischl et al.]	optimisation	instance features	algorithm	SVM	offline	static
[Veerapen et al.]	Quadratic Assignment Problem and TSP	past performance	algorithm	statistical model	online	static
[Valenzano et al.]	planning	past performance	resource allocation	statistical model	offline and online	static
[Hutter et al.; Hutter et al.]	SAT, MIP, TSP	instance features	algorithm performance	11 regression algorithms	offline	static
[Kanda et al.]	TSP	instance features	ranking	neural networks	offline	static
[Kadioglu et al.]	MIP	instance features	algorithm	clustering	online	static
[Seipp et al.]	planning	past performance	resource allocation	clustering and heuristic approaches	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Maratea et al.; Maratea et al.]	ASP	instance features	algorithm	classification	offline	static
[Muñoz et al.]	optimisation	instance features, algorithm parameters	runtime performance	neural network regression	offline	static
[Park et al.]	software design	instance features	runtime performance	SVM	offline	static
[Sabharwal et al.]	SAT	instance features	resource allocation and switch algorithm?	nearest neighbour and decision tree classification	offline and online	static
[Abell et al.]	black-box optimisation	instance features	algorithm	clustering	offline	static
[Hutter et al.]	SAT, MIP, TSP	instance features and algorithm parameters	algorithm performance	random forests, linear regression, neural networks, Gaussian processes, regression trees	offline	static
[Musliu and Schwengerer]	graph coloring	instance features	algorithm	six classifiers	offline	static
[Amadini et al.]	constraints	instance features	algorithm	range of different approaches	offline	static
[Alhossaini and Beck]	planning	instance features	model	SVM	offline	static
[Seijen et al.]	reinforcement learning	past performance	abstraction	MDP	online	static
[Malitsky et al.]	SAT	instance features	algorithm	clustering	online	static
[Mehta et al.]	constraints	instance features	algorithm	classification, regression and clustering	offline	static
[Malitsky et al.]	SAT	instance features	algorithm	clustering	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Rayner et al.]	combinatorial search	probing	subset of algorithms	optimisation	offline	static
[Sun and Pfahringer]	machine learning	past performance	ranking	pairwise rules and trees	offline	static
[Collautti et al.]	SAT	instance features, past performance	algorithm	nearest neighbour, random forests	offline	static
[Maratea et al.]	ASP	instance features	algorithm	PART decision rules	offline	static
[Wang et al.]	feature selection	instance features	algorithm	nearest neighbour and optimisation	offline	static
[King et al.; King et al.]	power systems	instance features	algorithm	neural net, decision tree, random forest	offline	static
[Yuen et al.]	evolutionary algorithms	past performance	algorithm	linear regression	online	static
[Loth et al.]	constraints	past performance	algorithm	reinforcement learning	online	static
[Simon et al.]	software design	instance features	algorithm	neural networks, decision trees	offline	dynamic
[Amadini et al.]	constraints	instance features	algorithm, resource allocation	5 different classifiers	offline and online	static
[Cauwet et al.]	optimisation	past performance	resource allocation	statistical model	online	static
[Hoos et al.]	ASP, SAT, QBF, CSP	past performance	resource allocation	answer set programming	offline	static
[Hurley et al.]	CSP	instance features	instance encoding, algorithm	classification, regression, clustering	offline	static
[Kotthoff]	CSP, SAT, QBF	instance features	ranking	classification, regression, meta-learning	offline	static
[Tang et al.]	numerical optimisation	past performance	algorithm portfolio	optimisation	offline	dynamic

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Fawcett et al.]	planning	instance features	runtime	regression	offline	static
[Amadini and Stuckey]	COP	instance features	resource allocation	nearest neighbour	offline	static
[Blet et al.]	CSP	instance features	algorithm	M5P regression	offline	static
[Malitsky et al.]	Minimal Correction Subset	instance features, past performance	algorithm	nearest neighbour, random forests	offline	static
[Malitsky et al.]	Minimal Correction Subset	instance features	resource allocation	nearest neighbour, regression	offline	static
[Ansótegui et al.]	MaxSAT	instance features	algorithm	clustering	offline	static
[Malitsky and O'Sullivan]	CSP, MaxSAT, SAT	instance features, past performance	algorithm	random forest and linear regression	offline	static
[Smith et al.]	classification	past performance	algorithm	collaborative filtering	offline	static
[Garbajosa et al.]	planning	instance features	algorithm	classifier ensemble	online	static
[Amadini et al.]	constraints	instance features	resource allocation	nearest neighbour	offline	static
[Pihera and Nysret]	TSP	instance features	algorithm	5 classifiers	offline	static
[St-Pierre and Teytaud]	Go	past performance	policy	static rule and reinforcement learning	offline and online	static
[van Rijn et al.]	machine learning	instance features	algorithm	decision stumps, random forests	offline	static
[Lieder et al.]	sorting	instance features	performance	Bayesian regression	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Lindauer]	ASP, CSP, SAT, QBF, OR	instance features	resource allocation	lots	offline	static
[Hoos et al.]	ASP	instance features	resource allocation	pairwise classification, regression, clustering	offline	static
[Tierney and Malitsky]	container premarshalling	instance features, past performance	algorithm	hierarchical cost-sensitive clustering	offline	static
[Lindauer et al.]	SAT, QBF, ASP, container premarshalling	instance features	resource allocation	random forest pairwise classification, ridge regression, k-means clustering	offline	static
[Lindauer et al.]	aslib 1.0	instance features	resource allocation	pairwise classification, regression, clustering	offline	static
[Kotthoff et al.]	TSP	instance features	algorithm	classification, regression, pairwise regression	offline	static
[Sabar and Kendall]	combinatorial search	past performance	algorithm	reinforcement learning	online	static
[Oentaryo et al.]	SAT	instance features and past performance	ranking	stochastic optimisation	offline	static
[Chu and Stuckey]	constraints	instance features	algorithm	partial least squares regression	offline	static
[Balafrej et al.]	constraints	past performance	propagation method	multi-armed bandits	online	static
[Luo et al.]	stencil computation	instance features	solution space	multiple linear regression	offline	static

Table I: Summary of the Algorithm Selection literature.

REFERENCES

Abell, T., Malitsky, Y., and Tierney, K. 2013. Features for exploiting black-box optimization problem structure. In *LION 7*.

Aha, D. W. 1992. Generalizing from case studies: A case study. In *Proceedings of the 9th International Workshop on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1–10.

Aiguzhinov, A., Soares, C., and Serra, A. P. 2010. A similarity-based adaptation of naive bayes for label ranking: Application to the metalearning problem of algorithm recommendation. In *13th International conference on Discovery Science*. Springer-Verlag, 16–26.

Alhossaini, M. and Beck, J. C. 2013. Instance-specific remodelling of planning domains by adding macros and removing operators. In *Symposium on Abstraction, Reformulation, and Approximation*.

Ali, S. and Smith, K. A. 2006. On learning algorithm selection for classification. *Applied Soft Computing 6,* 2, 119–138.

Allen, J. A. and Minton, S. 1996. Selecting the right heuristic algorithm: Runtime performance predictors. In *The 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*. Springer-Verlag, 41–53.

Amadini, R., Gabbrielli, M., and Mauro, J. 2013. An empirical evaluation of portfolios approaches for solving CSPs. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Lecture Notes in Computer Science Series, vol. 7874. Springer Berlin Heidelberg, 316–324.

Amadini, R., Gabbrielli, M., and Mauro, J. 2014a. Portfolio approaches for constraint optimization problems. In *LION 8*.

Amadini, R., Gabbrielli, M., and Mauro, J. 2014b. SUNNY: a lazy portfolio approach for constraint solving. *TPLP 14,* 4-5, 509–524.

Amadini, R. and Stuckey, P. J. 2014. Sequential time splitting and bounds communication for a portfolio of optimization solvers. In *Principles and Practice of Constraint Programming*. 108–124.

Ansótegui, C., Malitsky, Y., and Sellmann, M. 2014. MaxSAT by improved instance-specific algorithm configuration. In *AAAI*. 2594–2600.

Arbelaez, A., Hamadi, Y., and Sebag, M. 2009. Online heuristic selection in constraint programming. In *Symposium on Combinatorial Search*.

Arbelaez, A., Hamadi, Y., and Sebag, M. 2010. Continuous search in constraint programming. In *22nd IEEE International Conference on Tools with Artificial Intelligence*. 53–60.

Armstrong, W., Christen, P., McCreath, E., and Rendell, A. P. 2006. Dynamic algorithm selection using reinforcement learning. In *International Workshop on Integrating AI and Data Mining*. 18–25.

Balafrej, A., Bessire, C., and Paparrizou, A. 2015. Multi-Armed Bandits for Adaptive Constraint Propagation. In *IJCAI*.

Beck, J. C. and Fox, M. S. 2000. Dynamic problem structure analysis as a basis for constraint-directed scheduling heuristics. *Artificial Intelligence 117,* 1, 31–81.

Beck, J. C. and Freuder, E. C. 2004. Simple rules for low-knowledge algorithm selection. In *CPAIOR*. Springer, 50–64.

Bhowmick, S., Eijkhout, V., Freund, Y., Fuentes, E., and Keyes, D. 2006. Application of machine learning in selecting sparse linear solvers. Tech. rep., Columbia University.

Bhowmick, S., Toth, B., and Raghavan, P. 2009. Towards Low-Cost, High-Accuracy classifiers for linear solver selection. In *Proceedings of the 9th International Conference on Computational Science*. ICCS '09. Springer-Verlag, Berlin, Heidelberg, 463–472.

Bischl, B., Mersmann, O., Trautmann, H., and Preuss, M. 2012. Algorithm selection based on exploratory landscape analysis and Cost-Sensitive learning. In *14th International Conference on Genetic and Evolutionary Computation*. GECCO '12. ACM, New York, NY, USA, 313–320.

Blet, L., Ndiaye, S., and Solnon, C. 2014. Experimental comparison of BTD and intelligent backtracking: Towards an automatic per-instance algorithm selector. In *Principles and Practice of Constraint Programming*. 190–206.

Borrett, J. E. and Tsang, E. P. K. 2001. A context for constraint satisfaction problem formulation selection. *Constraints 6,* 4, 299–327.

Borrett, J. E., Tsang, E. P. K., and Walsh, N. R. 1996. Adaptive constraint satisfaction: The quickest first principle. In *ECAI*. 160–164.

Bougeret, M., Dutot, P., Goldman, A., Ngoko, Y., and Trystram, D. 2009. Combining multiple heuristics on discrete resources. In *IEEE International Symposium on Parallel & Distributed Processing*. IEEE Computer Society, Washington, DC, USA, 1–8.

Brazdil, P. and Soares, C. 2000. A comparison of ranking methods for classification algorithm selection. In *Proceedings of the 11th European Conference on Machine Learning*. ECML '00. Springer-Verlag, London, UK, 63–74.

Brazdil, P. B., Soares, C., and Da Costa, J. P. 2003. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Mach. Learn. 50,* 3, 251–277.

Brewer, E. A. 1995. High-level optimization via automated statistical modeling. In *Proceedings of the 5th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. PPOPP '95. ACM, New York, NY, USA, 80–91.

Bridge, D., O'Mahony, E., and O'Sullivan, B. 2011. Case-Based reasoning for autonomous constraint solving. In *Autonomous Search*, Y. Hamadi, E. Monfroy, and F. Saubion, Eds. Springer Berlin Heidelberg, 73–95.

Brodley, C. E. 1993. Addressing the selective superiority problem: Automatic Algorithm/Model class selection. In *ICML*. 17–24.

Cahill, E. 1994. Knowledge-based algorithm construction for real-world engineering PDEs. *Mathematics and Computers in Simulation 36,* 4-6, 389–400.

Carbonell, J., Etzioni, O., Gil, Y., Joseph, R., Knoblock, C., Minton, S., and Veloso, M. 1991. PRODIGY: an integrated architecture for planning and learning. *SIGART Bull. 2*, 51–55.

Carchrae, T. and Beck, J. C. 2004. Low-Knowledge algorithm control. In *AAAI*. 49–54.

Carchrae, T. and Beck, J. C. 2005. Applying machine learning to Low-Knowledge control of optimization algorithms. *Computational Intelligence 21,* 4, 372–387.

Caseau, Y., Laburthe, F., and Silverstein, G. 1999. A Meta-Heuristic factory for vehicle routing problems. In *Proceedings of the 5th International Conference on Principles and Practice of Constraint Programming*. Springer-Verlag, London, UK, 144–158.

Cauwet, M.-L., Liu, J., and Teytaud, O. 2014. Algorithm portfolios for noisy optimization: Compare solvers early. In *LION 8*.

Cavazos, J. and O'Boyle, M. F. P. 2006. Method-specific Dynamic Compilation Using Logistic Regression. *SIGPLAN Not. 41,* 10, 229–240.

Chu, G. and Stuckey, P. J. 2015. Learning Value Heuristics for Constraint Programming. In *Integration of AI and OR Techniques in Constraint Programming*. 108–123.

Cicirello, V. A. and Smith, S. F. 2005. The max k-armed bandit: A new model of exploration applied to search heuristic selection. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press, 1355–1361.

Collautti, M., Malitsky, Y., Mehta, D., and O'Sullivan, B. 2013. SNNAP: solver-based nearest neighbor for algorithm portfolios. In *ECML/PKDD*. 435–450.

Cook, D. J. and Varnell, R. C. 1997. Maximizing the benefits of parallel search using machine learning. In *Proceedings of the 14th National Conference on Artificial Intelligence*. AAAI Press, 559–564.

Cowling, P., Kendall, G., and Soubeiga, E. 2001. A Parameter-Free hyperheuristic for scheduling a sales summit. In *Proceedings of the 4th Metaheuristic International Conference*. 127–131.

de la Rosa, T., Celorrio, S. J., and Borrajo, D. 2008. Learning relational decision trees for guiding heuristic planning. In *ICAPS*. 60–67.

de la Rosa, T., García Olaya, A., and Borrajo, D. 2007a. Using cases utility for heuristic planning improvement. In *Case-Based Reasoning Research and Development*. 137–148.

de la Rosa, T., García Olaya, A., and Borrajo, D. 2013. A case-based approach to heuristic planning. *Applied Intelligence 39,* 1, 184–201.

de la Rosa, T., Jiménez, S., Fuentetaja, R., and Borrajo, D. 2011. Scaling up heuristic planning with relational decision trees. *J. Artif. Int. Res. 40,* 1, 767–813.

de la Rosa, T., Olaya, A. G., and Borrajo, D. 2007b. Case-based recommendation of node ordering in planning. In *International Florida Artificial Intelligence Research Society Conference*. 393–398.

Demmel, J., Dongarra, J., Eijkhout, V., Fuentes, E., Petitet, A., Vuduc, R., Whaley, R. C., and Yelick, K. 2005. Self-Adapting linear algebra algorithms and software. *Proceedings of the IEEE 93,* 2, 293–312.

Domshlak, C., Karpas, E., and Markovitch, S. 2010. To max or not to max: Online learning for speeding up optimal planning. In *AAAI*.

Elsayed, S. A. M. and Michel, L. 2010. Synthesis of search algorithms from high-level CP models. In *Proceedings of the 9th International Workshop on Constraint Modelling and Reformulation*.

ELSAYED, S. A. M. AND MICHEL, L. 2011. Synthesis of search algorithms from high-level CP models. In *17th International Conference on Principles and Practice of Constraint Programming*. Springer-Verlag, Berlin, Heidelberg, 256–270.

EPSTEIN, S. AND PETROVIC, S. 2011. Learning a mixture of search heuristics. In *Autonomous Search*, Y. Hamadi, E. Monfroy, and F. Saubion, Eds. Springer Berlin Heidelberg, 97–127.

EPSTEIN, S. L. AND FREUDER, E. C. 2001. Collaborative learning for constraint solving. In *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming*. Springer-Verlag, London, UK, 46–60.

EPSTEIN, S. L., FREUDER, E. C., WALLACE, R., MOROZOV, A., AND SAMUELS, B. 2002. The adaptive constraint engine. In *Principles and Practice of Constraint Programming*. Springer, 525–540.

EPSTEIN, S. L., WALLACE, R. J., FREUDER, E. C., AND XINGJIAN, L. 2005. Learning propagation policies. In *Second International Workshop on Constraint Propagation and Implementation*.

EWALD, R., SCHULZ, R., AND UHRMACHER, A. M. 2010. Selecting simulation algorithm portfolios by genetic algorithms. In *IEEE Workshop on Principles of Advanced and Distributed Simulation*. PADS '10. IEEE Computer Society, Washington, DC, USA, 1–9.

FAWCETT, C., VALLATI, M., HUTTER, F., HOFFMANN, J., HOOS, H., AND LEYTON-BROWN, K. 2014. Improved features for runtime prediction of domain-independent planners. In *ICAPS*.

FIALHO, A., SCHOENAUER, M., AND SEBAG, M. 2010. Toward comparison-based adaptive operator selection. In *12th Annual Conference on Genetic and Evolutionary Computation*. 767–774.

FINK, E. 1997. Statistical selection among Problem-Solving methods. Tech. Rep. CMU-CS-97-101, Carnegie Mellon University.

FINK, E. 1998. How to solve it automatically: Selection among Problem-Solving methods. In *Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems*. AAAI Press, 128–136.

FUKUNAGA, A. S. 2000. Genetic algorithm portfolios. In *IEEE Congress on Evolutionary Computation*. Vol. 2. 1304–1311.

FUKUNAGA, A. S. 2002. Automated discovery of composite SAT variable-selection heuristics. In *18th National Conference on Artificial Intelligence*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 641–648.

FUKUNAGA, A. S. 2008. Automated discovery of local search heuristics for satisfiability testing. *Evol. Comput. 16*, 31–61.

GAGLIOLO, M. AND SCHMIDHUBER, J. 2005. A neural network model for Inter-Problem adaptive online time allocation. In *15th International Conference on Artificial Neural Networks: Formal Models and Their Applications*. Springer, 7–12.

GAGLIOLO, M. AND SCHMIDHUBER, J. 2006. Learning dynamic algorithm portfolios. *Ann. Math. Artif. Intell. 47,* 3-4, 295–328.

GAGLIOLO, M. AND SCHMIDHUBER, J. 2008. Towards distributed algorithm portfolios. In *International Symposium on Distributed Computing and Artificial Intelligence, Advances in Soft Computing*. Springer.

GAGLIOLO, M. AND SCHMIDHUBER, J. 2010. Algorithm selection as a bandit problem with unbounded losses. In *Learning and Intelligent Optimization*. Lecture Notes in Computer Science Series, vol. 6073. Springer Berlin Heidelberg, 82–96.

GAGLIOLO, M. AND SCHMIDHUBER, J. 2011. Algorithm portfolio selection as a bandit problem with unbounded losses. *Annals of Mathematics and Artificial Intelligence 61,* 2, 49–86.

GAGLIOLO, M., ZHUMATIY, V., AND SCHMIDHUBER, J. 2004. Adaptive online time allocation to search algorithms. In *ECML*. Springer, 134–143.

GARBAJOSA, A., DE LA ROSA, T., AND FUENTETAJA, R. 2014. Planning with ensembles of classifiers. In *ECAI*. 1007–1008.

GARRIDO, P. AND RIFF, M. 2010. DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. *Journal of Heuristics 16*, 795–834.

GEBRUERS, C., GUERRI, A., HNICH, B., AND MILANO, M. 2004. Making choices using structure at the instance level within a case based reasoning framework. In *CPAIOR*. 380–386.

GEBRUERS, C., HNICH, B., BRIDGE, D., AND FREUDER, E. 2005. Using CBR to select solution strategies in constraint programming. In *Proc. of ICCBR-05*. 222–236.

GEBSER, M., KAMINSKI, R., KAUFMANN, B., SCHAUB, T., SCHNEIDER, M. T., AND ZILLER, S. 2011. A portfolio solver for answer set programming: preliminary report. In *11th International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer-Verlag, Berlin, Heidelberg, 352–357.

GENT, I., JEFFERSON, C., KOTTHOFF, L., MIGUEL, I., MOORE, N., NIGHTINGALE, P., AND PETRIE, K. 2010a. Learning when to use lazy learning in constraint solving. In *19th European Conference on Artificial Intelligence*. 873–878.

GENT, I., KOTTHOFF, L., MIGUEL, I., AND NIGHTINGALE, P. 2010b. Machine learning for constraint solver design - a case study for the alldifferent constraint. In *3rd Workshop on Techniques for implementing Constraint Programming Systems (TRICS)*. 13–25.

GEREVINI, A. E., SAETTI, A., AND VALLATI, M. 2009. An automatically configurable portfolio-based planner with macro-actions: PbP. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling*. 350–353.

GOMES, C. P. AND SELMAN, B. 1997a. Algorithm portfolio design: Theory vs. practice. In *UAI*. 190–197.

GOMES, C. P. AND SELMAN, B. 1997b. Practical aspects of algorithm portfolio design. In *Proc. of 3rd ILOG International Users Meeting*.

GOMES, C. P. AND SELMAN, B. 2001. Algorithm portfolios. *Artificial Intelligence 126,* 1-2, 43–62.

GRAFF, M. AND POLI, R. 2010. Practical performance models of algorithms in evolutionary program induction and other domains. *Artificial Intelligence 174,* 15, 1254–1276.

GRATCH, J. AND DEJONG, G. 1992. COMPOSER: a probabilistic solution to the utility problem in Speed-Up learning. In *AAAI*. 235–240.

GUERRI, A. AND MILANO, M. 2004. Learning techniques for automatic algorithm portfolio selection. In *ECAI*. 475–479.

GUO, H. 2003. Algorithm selection for sorting and probabilistic inference: A machine Learning-Based approach. Ph.D. thesis, Kansas State University.

GUO, H. AND HSU, W. H. 2004. A Learning-Based algorithm selection meta-reasoner for the Real-Time MPE problem. In *Australian Conference on Artificial Intelligence*. 307–318.

HAIM, S. AND WALSH, T. 2009. Restart strategy selection using machine learning techniques. In *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing*. Springer-Verlag, Berlin, Heidelberg, 312–325.

HELMERT, M., RÖGER, G., AND KARPAS, E. 2011. Fast downward stone soup: A baseline for building planner portfolios. In *ICAPS-2011 Workshop on Planning and Learning (PAL)*. 28–35.

HOOS, H., LINDAUER, M., AND SCHAUB, T. 2014a. claspfolio 2: Advances in algorithm selection for answer set programming. *TPLP 14,* 4-5, 569–585.

HOOS, H. H., KAMINSKI, R., LINDAUER, M., AND SCHAUB, T. 2014b. aspeed: Solver scheduling via answer set programming. *Theory and Practice of Logic Programming FirstView*, 1–26.

HORVITZ, E., RUAN, Y., GOMES, C. P., KAUTZ, H. A., SELMAN, B., AND CHICKERING, D. M. 2001. A bayesian approach to tackling hard computational problems. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 235–244.

HOUGH, P. D. AND WILLIAMS, P. J. 2006. Modern machine learning for automatic optimization algorithm selection. In *Proceedings of the INFORMS Artificial Intelligence and Data Mining Workshop*.

HOWE, A. E., DAHLMAN, E., HANSEN, C., SCHEETZ, M., AND VON MAYRHAUSER, A. 1999. Exploiting competitive planner performance. In *Proceedings of the 5th European Conference on Planning*. Springer, 62–72.

HUBERMAN, B. A., LUKOSE, R. M., AND HOGG, T. 1997. An economics approach to hard computational problems. *Science 275,* 5296, 51–54.

HURLEY, B., KOTTHOFF, L., MALITSKY, Y., AND O'SULLIVAN, B. 2014. Proteus: A hierarchical portfolio of solvers and transformations. In *CPAIOR*.

HURLEY, B. AND OSULLIVAN, B. 2012. Adaptation in a CBR-Based solver portfolio for the satisfiability problem. In *Case-Based Reasoning Research and Development*. Lecture Notes in Computer Science Series, vol. 7466. 152–166.

HUTTER, F., HAMADI, Y., HOOS, H. H., AND LEYTON-BROWN, K. 2006. Performance prediction and automated tuning of randomized and parametric algorithms. In *CP*. 213–228.

HUTTER, F., HOOS, H. H., AND LEYTON-BROWN, K. 2013. Identifying key algorithm parameters and instance features using forward selection. In *LION 7*.

HUTTER, F., XU, L., HOOS, H. H., AND LEYTON-BROWN, K. 2012. Algorithm runtime prediction: The state of the art. *CoRR abs/1211.0906*.

HUTTER, F., XU, L., HOOS, H. H., AND LEYTON-BROWN, K. 2014. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence 206,* 0, 79–111.

JOSHI, A., WEERAWARANA, S., RAMAKRISHNAN, N., HOUSTIS, E. N., AND RICE, J. R. 1996. Neuro-Fuzzy support for Problem-Solving environments: A step toward automated solution of PDEs. *IEEE Comput. Sci. Eng. 3,* 1, 44–56.

KADIOGLU, S., MALITSKY, Y., SABHARWAL, A., SAMULOWITZ, H., AND SELLMANN, M. 2011. Algorithm selection and scheduling. In *17th International Conference on Principles and Practice of Constraint Programming*. 454–469.

KADIOGLU, S., MALITSKY, Y., AND SELLMANN, M. 2012. Non-model-based search guidance for set partitioning problems. In *AAAI*.

KADIOGLU, S., MALITSKY, Y., SELLMANN, M., AND TIERNEY, K. 2010. ISAC Instance-Specific algorithm configuration. In *19th European Conference on Artificial Intelligence*. IOS Press, 751–756.

KAMEL, M. S., ENRIGHT, W. H., AND MA, K. S. 1993. ODEXPERT: an expert system to select numerical solvers for initial value ODE systems. *ACM Trans. Math. Softw. 19,* 1, 44–62.

KANDA, J., DE CARVALHO, A., HRUSCHKA, E., AND SOARES, C. 2010. Using meta-learning to classify traveling salesman problems. In *Eleventh Brazilian Symposium on Neural Networks*. 73–78.

KANDA, J., DE CARVALHO, A., HRUSCHKA, E., AND SOARES, C. 2011. Selection of algorithms to solve traveling salesman problems using meta-learning. *Int. J. Hybrid Intell. Syst. 8,* 3, 117–128.

KANDA, J., SOARES, C., HRUSCHKA, E., AND DE CARVALHO, A. 2012. A meta-learning approach to select meta-heuristics for the traveling salesman problem using MLP-Based label ranking. In *19th International Conference on Neural Information Processing*. Springer-Verlag, Berlin, Heidelberg, 488–495.

KING, J. E., JUPE, S. C. E., AND TAYLOR, P. C. 2014. Network state-based algorithm selection for power flow management using machine learning. *IEEE Transactions on Power Systems PP,* 99, 1–8.

KING, J. E., TAYLOR, P. C., AND JUPE, S. C. E. 2013. Autonomic control algorithm selection in decentralised power systems: A voltage control case study. In *International Conference and Exhibition on Electricity Distribution (CIRED 2013)*. 1–4.

KIZILTAN, Z., MANDRIOLI, L., MAURO, J., AND O'SULLIVAN, B. 2011. A classification-based approach to managing a solver portfolio for CSPs. In *22nd Irish Conference on Artificial Intelligence and Cognitive Science*.

KOTTHOFF, L. 2012. Hybrid regression-classification models for algorithm selection. In *20th European Conference on Artificial Intelligence*. 480–485.

KOTTHOFF, L. 2014. Ranking algorithms by performance. In *LION 8*.

KOTTHOFF, L., GENT, I. P., AND MIGUEL, I. 2011. A preliminary evaluation of machine learning in algorithm selection for search problems. In *4th Annual Symposium on Combinatorial Search*. 84–91.

KOTTHOFF, L., GENT, I. P., AND MIGUEL, I. 2012. An evaluation of machine learning in algorithm selection for search problems. *AI Communications 25,* 3, 257–270.

KOTTHOFF, L., KERSCHKE, P., HOOS, H., AND TRAUTMANN, H. 2015. Improving the state of the art in inexact TSP solving using per-instance algorithm selection. In *LION 9*.

KOTTHOFF, L., MIGUEL, I., AND NIGHTINGALE, P. 2010. Ensemble classification for constraint solver configuration. In *16th International Conference on Principles and Practices of Constraint Programming*. 321–329.

KROER, C. AND MALITSKY, Y. 2011. Feature filtering for Instance-Specific algorithm configuration. In *Proceedings of the 23rd International Conference on Tools with Artificial Intelligence*.

KUEFLER, E. AND CHEN, T. 2008. On using reinforcement learning to solve sparse linear systems. In *Proceedings of the 8th International Conference on Computational Science*. ICCS '08. Springer-Verlag, Berlin, Heidelberg, 955–964.

LAGOUDAKIS, M. G. AND LITTMAN, M. L. 2000. Algorithm selection using reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 511–518.

LAGOUDAKIS, M. G. AND LITTMAN, M. L. 2001. Learning to select branching rules in the DPLL procedure for satisfiability. In *LICS/SAT*. 344–359.

LANGLEY, P. 1983a. Learning effective search heuristics. In *IJCAI*. 419–421.

LANGLEY, P. 1983b. Learning search strategies through discrimination. *International Journal of Man-Machine Studies*, 513–541.

LEITE, R. AND BRAZDIL, P. 2010. Active testing strategy to predict the best classification algorithm via sampling and metalearning. In *ECAI*. 309–314.

LEITE, R., BRAZDIL, P., VANSCHOREN, J., AND QUEIROS, F. 2010. Using active testing and Meta-Level information for selection of classification algorithms. In *3rd PlanLearn Workshop*.

LEYTON-BROWN, K., NUDELMAN, E., AND SHOHAM, Y. 2002. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*. Springer-Verlag, London, UK, 556–572.

LIEDER, F., PLUNKETT, D., HAMRICK, J. B., RUSSELL, S. J., HAY, N. J., AND GRIFFITHS, T. L. 2014. Algorithm selection by rational metareasoning as a model of human strategy selection. In *Advances in Neural Information Processing Systems*. Vol. 27.

LINDAUER, M. 2014. Algorithm selection, scheduling and configuration of boolean constraint solvers. Ph.D. thesis, University of Potsdam.

LINDAUER, M., HOOS, H. H., AND HUTTER, F. 2015a. From sequential algorithm selection to parallel portfolio selection. In *Proceedings of the International Conference on Learning and Intelligent Optimization (LION'15)*.

LINDAUER, M., HOOS, H. H., HUTTER, F., AND SCHAUB, T. 2015b. AutoFolio: Algorithm configuration for algorithm selection. In *Proceedings of the Twenty-Ninth AAAI Workshops on Artificial Intelligence*.

LITTLE, J., GEBRUERS, C., BRIDGE, D., AND FREUDER, E. 2002. Capturing constraint programming experience: A Case-Based approach. In *Modref*.

LOBJOIS, L. AND LEMAÎTRE, M. 1998. Branch and bound algorithm selection by performance prediction. In *Proceedings of the 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 353–358.

LOTH, M., SEBAG, M., HAMADI, Y., AND SCHOENAUER, M. 2013. Bandit-based search for constraint programming. In *Principles and Practice of Constraint Programming*. 464–480.

LUO, Y., TAN, G., MO, Z., AND SUN, N. 2015. FAST: A Fast Stencil Autotuning Framework Based On An Optimal-solution Space Model. In *29th ACM on International Conference on Supercomputing*. ACM, 187–196.

MALITSKY, Y., ASHISH, S., SAMULOWITZ, H., AND SELLMANN, M. 2012. Parallel SAT solver selection and scheduling. In *Principles and Practice of Constraint Programming*.

MALITSKY, Y., MEHTA, D., AND O'SULLIVAN, B. 2013a. Evolving instance specific algorithm configuration. In *Symposium on Combinatorial Search*.

MALITSKY, Y. AND O'SULLIVAN, B. 2014. Latent features for algorithm selection. In *SoCS*.

MALITSKY, Y., O'SULLIVAN, B., PREVITI, A., AND MARQUES-SILVA, J. A. 2014a. Timeout-sensitive portfolio approach to enumerating minimal correction subsets for satisfiability problems. In *ECAI*. 1065–1066.

MALITSKY, Y., OSULLIVAN, B., PREVITI, A., AND MARQUES-SILVA, J. A. 2014b. A portfolio approach to enumerating minimal correction subsets for satisfiability problems. In *CPAIOR*.

MALITSKY, Y., SABHARWAL, A., SAMULOWITZ, H., AND SELLMANN, M. 2011. Non-model-based algorithm portfolios for SAT. In *Theory and Applications of Satisfiability Testing (SAT)*. 369–370.

MALITSKY, Y., SABHARWAL, A., SAMULOWITZ, H., AND SELLMANN, M. 2013b. Algorithm portfolios based on cost-sensitive hierarchical clustering. In *IJCAI*.

MARATEA, M., PULINA, L., AND RICCA, F. 2012. Applying machine learning techniques to ASP solving. In *ICLP*. 37–48.

MARATEA, M., PULINA, L., AND RICCA, F. 2013a. Automated selection of grounding algorithm in answer set programming. In *AI*IA*. 73–84.

MARATEA, M., PULINA, L., AND RICCA, F. 2013b. A multi-engine approach to answer-set programming. *Theory and Practice of Logic Programming*, 1–28.

MATURANA, J., FIALHO, A., SAUBION, F., SCHOENAUER, M., LARDEUX, F., AND SEBAG, M. 2011. Adaptive operator selection and management in evolutionary algorithms. In *Autonomous Search*, Y. Hamadi, E. Monfroy, and F. Saubion, Eds. Springer Berlin Heidelberg, 161–189.

MEHTA, D., O'SULLIVAN, B., KOTTHOFF, L., AND MALITSKY, Y. 2013. Lazy branching for constraint satisfaction. In *ICTAI*.

MINTON, S. 1993a. An analytic learning system for specializing heuristics. In *IJCAI'93: Proceedings of the 13th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 922–928.

MINTON, S. 1993b. Integrating heuristics for constraint satisfaction problems: A case study. In *AAAI: Proceedings of the 11th National Conference on Artificial Intelligence*. 120–126.

MINTON, S. 1996. Automatically configuring constraint satisfaction programs: A case study. *Constraints 1*, 7–43.

MUÑOZ, M. A., KIRLEY, M., AND HALGAMUGE, S. K. 2012. A meta-learning prediction model of algorithm performance for continuous optimization problems. In *Parallel Problem Solving from Nature - PPSN XII*. Lecture Notes in Computer Science Series, vol. 7491. Springer Berlin Heidelberg, 226–235.

MUSLIU, N. AND SCHWENGERER, M. 2013. Algorithm selection for the graph coloring problem. In *LION 7*.

NAREYEK, A. 2001. Choosing search heuristics by Non-Stationary reinforcement learning. In *Metaheuristics: Computer Decision-Making*. Kluwer Academic Publishers, 523–544.

NIKOLIĆ, M., MARIĆ, F., AND JANIČIĆ, P. 2009. Instance-Based selection of policies for SAT solvers. In *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing*. SAT '09. Springer-Verlag, Berlin, Heidelberg, 326–340.

NUDELMAN, E., LEYTON-BROWN, K., HOOS, H. H., DEVKAR, A., AND SHOHAM, Y. 2004. Understanding random SAT: beyond the Clauses-to-Variables ratio. In *Principles and Practice of Constraint Programming CP 2004*, M. Wallace, Ed. Lecture Notes in Computer Science Series, vol. 3258. Springer Berlin / Heidelberg, 438–452.

OENTARYO, R. J., HANDOKO, D., AND LAU, H. C. 2015. Algorithm selection via ranking. In *AAAI*.

O'MAHONY, E., HEBRARD, E., HOLLAND, A., NUGENT, C., AND O'SULLIVAN, B. 2008. Using case-based reasoning in an algorithm portfolio for constraint solving. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*.

PAPARRIZOU, A. AND STERGIOU, K. 2012. Evaluating simple fully automated heuristics for adaptive constraint propagation. In *ICTAI*.

PARK, E., CAVAZOS, J., AND ALVAREZ, M. A. 2012. Using Graph-based Program Characterization for Predictive Modeling. In *10th International Symposium on Code Generation and Optimization*. 196–206.

PENG, F., TANG, K., CHEN, G., AND YAO, X. 2010. Population-based algorithm portfolios for numerical optimization. *Evolutionary Computation, IEEE Transactions on 14,* 5, 782–800.

PETRIK, M. 2005. Statistically optimal combination of algorithms. In *Local Proceedings of SOFSEM 2005*.

PETROVIC, S. AND QU, R. 2002. Case-Based reasoning as a heuristic selector in Hyper-Heuristic for course timetabling problems. In *KES*. 336–340.

PFAHRINGER, B., BENSUSAN, H., AND GIRAUD-CARRIER, C. G. 2000. Meta-Learning by landmarking various learning algorithms. In *17th International Conference on Machine Learning*. ICML '00. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 743–750.

PIHERA, J. AND NYSRET, M. 2014. Application of machine learning to algorithm selection for TSP. In *ICTAI*.

PRUDÊNCIO, R. B., DE SOUTO, M. C., AND LUDERMIR, T. B. 2011. Selecting machine learning algorithms using the ranking meta-learning approach. In *Meta-Learning in Computational Intelligence*. Vol. 358. 225–243.

PRUDÊNCIO, R. B. C. AND LUDERMIR, T. B. 2004. Meta-learning approaches to selecting time series models. *Neurocomputing 61,* 0, 121–137.

PULINA, L. AND TACCHELLA, A. 2007. A multi-engine solver for quantified boolean formulas. In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming*. CP'07. Springer-Verlag, Berlin, Heidelberg, 574–589.

PULINA, L. AND TACCHELLA, A. 2009. A self-adaptive multi-engine solver for quantified boolean formulas. *Constraints 14,* 1, 80–116.

PULINA, L. AND TACCHELLA, A. 2010. AQME'10. *JSAT 7,* 2-3, 65–70.

RAYNER, C., STURTEVANT, N., AND BOWLING, M. 2013. Subset selection of search heuristics. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*. 637–643.

ROBERTS, M. AND HOWE, A. E. 2006. Directing a portfolio with learning. In *AAAI 2006 Workshop on Learning for Search*.

ROBERTS, M. AND HOWE, A. E. 2007. Learned models of performance for many planners. In *ICAPS 2007 Workshop AI Planning and Learning*.

ROBERTS, M., HOWE, A. E., WILSON, B., AND DESJARDINS, M. 2008. What makes planners predictable? In *ICAPS*. 288–295.

RUAN, Y., HORVITZ, E., AND KAUTZ, H. A. 2002. Restart policies with dependence among runs: A dynamic programming approach. In *CP*, P. V. Hentenryck, Ed. Lecture Notes in Computer Science Series, vol. 2470. Springer, 573–586.

SABAR, N. R. AND KENDALL, G. 2015. Population based monte carlo tree search hyper-heuristic for combinatorial optimization problems. *Information Sciences 314,* 0, 225–239.

SABHARWAL, A., SAMULOWITZ, H., SELLMANN, M., AND MALITSKY, Y. 2013. Boosting sequential solver portfolios: Knowledge sharing and accuracy prediction. In *LION 7*.

SAKKOUT, H. E., WALLACE, M. G., AND RICHARDS, E. B. 1996. An instance of adaptive constraint propagation. In *Proc. of CP96*. Springer Verlag, 164–178.

SAMULOWITZ, H. AND MEMISEVIC, R. 2007. Learning to solve QBF. In *Proceedings of the 22nd National Conference on Artificial Intelligence*. AAAI Press, 255–260.

SAYAG, T., FINE, S., AND MANSOUR, Y. 2006. Combining multiple heuristics. In *STACS*. Vol. 3884. Springer, Berlin, Heidelberg, 242–253.

SEIJEN, H. V., WHITESON, S., AND KESTER, L. 2013. Efficient abstraction selection in reinforcement learning. *Computational Intelligence*.

SEIPP, J., BRAUN, M., GARIMORT, J., AND HELMERT, M. 2012. Learning portfolios of automatically tuned planners. In *ICAPS*. AAAI.

SHUKLA, N., TIWARI, M., AND CEGLAREK, D. 2012. Genetic-algorithms-based algorithm portfolio for inventory routing problem with stochastic demand. *International Journal of Production Research*, 1–20.

SILLITO, J. 2000. Improvements to and estimating the cost of solving constraint satisfaction problems. M.S. thesis, University of Alberta.

SILVERTHORN, B. AND MIIKKULAINEN, R. 2010. Latent class models for algorithm portfolio methods. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*.

SIMON, D., CAVAZOS, J., WIMMER, C., AND KULKARNI, S. 2013. Automatic Construction of Inlining Heuristics Using Machine Learning. In *IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*. 1–12.

SMITH, M. R., MITCHELL, L., GIRAUD-CARRIER, C. G., AND MARTINEZ, T. R. 2014. Recommending learning algorithms and their associated hyperparameters. In *MetaSel*.

SMITH, T. E. AND SETLIFF, D. E. 1992. Knowledge-based constraint-driven software synthesis. In *Knowledge-Based Software Engineering Conference*. 18–27.

SMITH-MILES, K. AND HEMERT, J. 2011. Discovering the suitability of optimisation algorithms by learning from evolved instances. *Annals of Mathematics and Artificial Intelligence 61,* 2, 87–104.

SMITH-MILES, K. A. 2008. Towards insightful algorithm selection for optimisation using Meta-Learning concepts. In *IEEE International Joint Conference on Neural Networks*. 4118–4124.

SMITH-MILES, K. A., JAMES, R. J., GIFFIN, J. W., AND TU, Y. 2009. A knowledge discovery approach to understanding relationships between scheduling problem structure and heuristic performance. In *Learning and Intelligent Optimization*. Vol. 5851. Springer Berlin Heidelberg, 89–103.

SOARES, C. AND BRAZDIL, P. 2000. Zoomed ranking: Selection of classification algorithms based on relevant performance information. In *Principles of Data Mining and Knowledge Discovery*. Lecture Notes in Computer Science. 126–135.

SOARES, C., BRAZDIL, P. B., AND KUBA, P. 2004. A Meta-Learning method to select the kernel width in support vector regression. *Mach. Learn. 54,* 3, 195–209.

ST-PIERRE, D. L. AND TEYTAUD, O. 2014. The nash and the bandit approaches for adversarial portfolios. In *CIG*.

STAMATATOS, E. AND STERGIOU, K. 2009. Learning how to propagate using random probing. In *Proceedings of the 6th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer-Verlag, Berlin, Heidelberg, 263–278.

STEER, K. C., WIRTH, A., AND HALGAMUGE, S. K. 2008. Information theoretic classification of problems for metaheuristics. In *Simulated Evolution and Learning*. Lecture Notes in Computer Science Series, vol. 5361. Springer Berlin Heidelberg, 319–328.

STERGIOU, K. 2008. Heuristics for dynamically adapting propagation. In *ECAI*. 485–489.

STERGIOU, K. 2009. Heuristics for dynamically adapting propagation in constraint satisfaction problems. *AI Commun. 22,* 3, 125–141.

STERN, D. H., SAMULOWITZ, H., HERBRICH, R., GRAEPEL, T., PULINA, L., AND TACCHELLA, A. 2010. Collaborative expert portfolio management. In *AAAI*. 179–184.

STREETER, M. J., GOLOVIN, D., AND SMITH, S. F. 2007. Combining multiple heuristics online. In *Proceedings of the 22nd National Conference on Artificial Intelligence*. AAAI Press, 1197–1203.

STREETER, M. J. AND SMITH, S. F. 2008. New techniques for algorithm portfolio design. In *UAI*. 519–527.

SUN, Q. AND PFAHRINGER, B. 2013. Pairwise meta-rules for better meta-learning-based algorithm ranking. *Machine Learning 93,* 1, 141–161.

TANG, K., PENG, F., CHEN, G., AND YAO, X. 2014. Population-based algorithm portfolios with automated constituent algorithms selection. *Information Sciences 279,* 0, 94–104.

TERASHIMA-MARÍN, H., ROSS, P., AND VALENZUELA-RENDÓN, M. 1999. Evolution of constraint satisfaction strategies in examination timetabling. In *Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 635–642.

TIERNEY, K. AND MALITSKY, Y. 2015. An algorithm selection benchmark of the container pre-marshalling problem. In *Learning and Intelligent OptimizatioN (LION) 2015*.

TOLPIN, D. AND SHIMONY, S. E. 2011. Rational deployment of CSP heuristics. In *IJCAI*. 680–686.

TSANG, E. P. K., BORRETT, J. E., AND KWAN, A. C. M. 1995. An attempt to map the performance of a range of algorithm and heuristic combinations. In *Proc. of AISB'95*. IOS Press, 203–216.

Valenzano, R., Sturtevant, N., Schaeffer, J., and Buro, K. 2010. Simultaneously searching with multiple settings: An alternative to parameter tuning for suboptimal single-agent search algorithms. In *ICAPS*. 177–184.

Valenzano, R. A., Nakhost, H., Müller, M., Schaeffer, J., and Sturtevant, N. R. 2012. ArvandHerd: parallel planning with a portfolio. *European Conference on Artificial Intelligence (ECAI)*, 786–791.

van Rijn, J. N., Holmes, G., Pfahringer, B., and Vanschoren, J. 2014. Algorithm selection on data streams. In *Discovery Science*. 325–336.

Veerapen, N., Maturana, J., and Saubion, F. 2012. An Exploration-Exploitation Compromise-Based adaptive operator selection for local search. In *14th International Conference on Genetic and Evolutionary Computation*. GECCO '12. ACM, New York, NY, USA, 1277–1284.

Vrakas, D., Tsoumakas, G., Bassiliades, N., and Vlahavas, I. 2003. Learning rules for adaptive planning. In *Proceedings of the 13th International Conference on Automated Planning and Scheduling*. 82–91.

Wang, G., Song, Q., Sun, H., Zhang, X., Xu, B., and Zhou, Y. 2013. A feature subset selection algorithm automatic recommendation method. *J. Artif. Int. Res. 47,* 1, 1–34.

Wang, J. and Tropper, C. 2007. Optimizing time warp simulation with reinforcement learning techniques. In *Proceedings of the 39th conference on Winter simulation*. WSC '07. IEEE Press, Piscataway, NJ, USA, 577–584.

Watson, J. 2003. Empirical modeling and analysis of local search algorithms for the job-shop scheduling problem. Ph.D. thesis, Colorado State University, Fort Collins, CO, USA.

Weerawarana, S., Houstis, E. N., Rice, J. R., Joshi, A., and Houstis, C. E. 1996. PYTHIA: a knowledge-based system to select scientific algorithms. *ACM Trans. Math. Softw. 22,* 4, 447–468.

Wei, W., Li, C. M., and Zhang, H. 2008. Switching among Non-Weighting, clause weighting, and variable weighting in local search for SAT. In *Proceedings of the 14th International Conference on Principles and Practice of Constraint Programming*. Springer-Verlag, Berlin, Heidelberg, 313–326.

Wilson, D., Leake, D., and Bramley, R. 2000. Case-Based recommender components for scientific Problem-Solving environments. In *Proc. of the 16th International Association for Mathematics and Computers in Simulation World Congress*.

Wu, H. and van Beek, P. 2007. On portfolios for backtracking search in the presence of deadlines. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*. IEEE Computer Society, Washington, DC, USA, 231–238.

Xu, L., Hoos, H. H., and Leyton-Brown, K. 2007a. Hierarchical hardness models for SAT. In *CP*. 696–711.

Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K. 2007b. SATzilla-07: the design and analysis of an algorithm portfolio for SAT. In *CP*. 712–727.

Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K. 2008. SATzilla: portfolio-based algorithm selection for SAT. *J. Artif. Intell. Res. (JAIR) 32*, 565–606.

Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K. 2011. Hydra-MIP: automated algorithm configuration and selection for mixed integer programming. In *RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion at the International Joint Conference on Artificial Intelligence (IJCAI)*.

Xu, Y., Stern, D., and Samulowitz, H. 2009. Learning adaptation to solve constraint satisfaction problems. In *Learning and Intelligent Optimization*.

Yu, H., Dang, F., and Rauchwerger, L. 2002. Parallel reductions: An application of adaptive algorithm selection. In *Proceedings of the 15th International Conference on Languages and Compilers for Parallel Computing*. Springer-Verlag, Berlin, Heidelberg, 188–202.

Yu, H. and Rauchwerger, L. 2006. An adaptive algorithm selection framework for reduction parallelization. *IEEE Transactions on Parallel and Distributed Systems 17,* 10, 1084–1096.

Yu, H., Zhang, D., and Rauchwerger, L. 2004. An adaptive algorithm selection framework. In *Proceedings of the 13th International Conference on Parallel Architectures and Compilation Techniques*. IEEE Computer Society, Washington, DC, USA, 278–289.

Yuen, S. Y., Chow, C. K., and Zhang, X. 2013. Which algorithm should i choose at any point of the search: An evolutionary portfolio approach. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. 567–574.

Yun, X. and Epstein, S. L. 2012. Learning algorithm portfolios for parallel execution. In *Proceedings of the 6th International Conference Learning and Intelligent Optimisation LION*. Springer, 323–338.