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
[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
[Nareyek]	optimisation	search statistics	expected utility of algorithm	reinforcement learning	offline and online	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[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
[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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[Ali and Smith]	classification	instance features	algorithm	decision rules	offline	static
[Xu et al.]	SAT	instance features	satisfiability and runtime performance	sparse multinomial logistic regression, ridge regression	offline	static

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[Nikolić et al.]	SAT	instance features	search strategy	nearest neighbour	offline	static
[Stamatatos and Stergiou]	constraints	probing	propagation method	clustering	offline	static
[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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[Graff and Poli]	program induction	fitness function	runtime performance	regression	offline	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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[Smith-Miles and Hemert]	TSP	instance features	algorithm	self-organizing map, decision tree, neural network	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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[Hutter et al.]	SAT, MIP, TSP	instance features	algorithm performance	11 algorithms	offline	static
[Kanda et al.]	TSP	instance features	ranking	neural networks	offline	static
[Kadioglu et al.]	MIP	instance features	heuristic	clustering	online	static
[Seipp et al.]	planning	past performance	resource allocation	clustering and heuristic approaches	offline	static
[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
[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

continued on next page

citation	domain	features	predict what	predict how	predict when	portfolio
[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
[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	clustering	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
[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

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. 2014. Portfolio approaches for constraint optimization problems. In *LION 8*.

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.

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.

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*.

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.

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.

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.

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. H., Kaminski, R., Lindauer, M., and Schaub, T. 2014. 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*.

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.

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., 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.

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.

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., 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.

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.*

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.

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.

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.

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.*

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., 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.

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.

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.

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.

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.