Constraint Optimization for Timetabling Problems Using a Constraint Driven Solution Model

Anurag Sharma and Dharmendra Sharma

Information Technology and Engineering
University of Canberra, Australia
{Anurag.Sharma,Dharmendra.Sharma}@canberra.edu.au

Abstract. Many science and engineering applications require finding solutions to planning and optimization problems by satisfying a set of constraints. These constraint problems (CPs) are typically NP-complete and can be formalized as constraint satisfaction problems (CSPs) or constraint optimization problems (COPs). Evolutionary algorithms (EAs) are good solvers for optimization problems ubiquitous in various problem domains. A variation of EA - Intelligent constraint handling evolutionary algorithm (ICHEA) has been demonstrated to be a versatile constraints-guided EA for all forms of continuous constrained problems in our earlier works. In this paper we investigate an incremental approach through ICHEA in solving benchmark exam timetabling problems which is a classic discrete COP and compare its performance with other well-known EAs. Incremental and exploratory search in constraint solving has shown improvement in the quality of solutions.

Keywords: constraint satisfaction problems, constraint optimization problems, evolutionary algorithms, exam timetabling problems.

1 Introduction

Many engineering problems ranging from resource allocation and scheduling to fault diagnosis and design involve constraint satisfaction as an essential component that require finding solutions to satisfy a set of constraints over real numbers or discrete representation of constraints [12, 13, 19]. There are many classical algorithms that solve CSPs like branch and bound, backtrack algorithm, iterative forward search algorithm, local search but heuristic methods such as evolutionary algorithms (EAs) have mixed success and for many difficult problems these are the only available choice [2, 13, 17]. EAs however suffer from some of its inherent problems to solve CSPs as it does not make use of knowledge from constraints and blindly search in the vast solution space using its heuristic search mechanism. Constraints can reduce the search space and direct the evolutionary search towards feasible regions. Additionally *large* static CPs can be solved like a dynamic CP formulation where a subset of constraints is added incrementally. The incremental approach shows more effective results in terms of evaluation parameters of success rate (SR) and efficiency. SR is the rate of successful trials for each problem i.e. $SR = successful\ trials / \ total\ trials$.

S. Cranefield and A. Nayak (Eds.): AI 2013, LNCS 8272, pp. 196–201, 2013.

The main contribution of this paper is to show that incremental approach in solving constraints leads to better quality solutions. Incremental approach also helps in getting feasible solutions without any need to have a separate bespoke algorithm. We have enhanced the existing ICHEA [20, 21] to solve discrete COPs. Now ICHEA can be used to solve any form of a CP. The paper is organized as follows: Section 2 describes the cost function of the benchmark timetabling problems and Section 3 describes how the algorithm of ICHEA incrementally solves them. Section 4 demonstrates experiments on benchmark exam timetabling problems. Section 5 discusses the experimental results and Section 6 concludes the paper by summarizing the results confirming the claim against the established hypothesis and proposing some further possible extensions to the research.

2 Solving Exam Timetabling Problems

There are two types of constraints: if constraints are required to be satisfied under any circumstances to have an acceptable solution are known as *hard constraints*. Another type of constraints are called *soft constraints* that are considered to be desirable but not essential [3]. Solutions, which satisfy all the hard constraints, are often called *feasible* solutions. Soft constraints can have some degree of satisfaction or order of preferences for a particular problem. Soft constraints can be represented by penalty functions for COPs where higher weights demonstrate lower preferences and vice versa for higher preferences. We used University of Toronto benchmark exam timetabling problems (version I) given in [18, 23] where the given weights based on the spread of exams for each student is:

$$W_d = S_d 2^{4-d} \tag{1}$$

where d is the distance between two timeslots in the range [0 4], S_d is total corresponding students and W_d is the total corresponding weight. The cost function is the average weight corresponds to each student given as:

$$f = \frac{1}{S}\sum_{d=0}^{4} S_p 2^{4-d} \tag{2}$$

We mainly used our modified *Kempe* chain [8, 14] techniques for mutating timetables in ICHEA that also follows a reversible hill climbing technique that works like a backtracking algorithm. The details will be provided in the extended journal paper.

3 ICHEA Algorithm for Exam Timetabling Problems

Some *large* CPs like exam timetabling problems can be divided into several components (subsets of constraints) then each component can be solved incrementally. This divide and conquer approach solves a CP by taking a component to get feasible solutions before taking next component. In the literature, exam timetabling problems sort the constraints according to the largest degree (LD), saturation degree (SD), largest weighted degree (LWD), largest penalty (LP) or random Order (RO) [4, 7, 9]. LD and

SD are commonly used sorting order. In LD exams are ordered decreasingly according to the number of conflicts each exam has with others, and in SD the exams are ordered increasingly according to the number of remaining timeslots available to assign them without causing conflicts. The definition of other sorting orders can be found in [7]. ICHEA uses LD to sort all the exams based on clashes with other exams. It takes only 5% of the sorted exams in every increment and once a feasible solution is obtained the optimization operators are applied for G generations before taking next increment of exams. The value of G is 500 in our experiments.

Solving CPs incrementally has many advantages. Incrementality in solving CPs also comes handy when a new constraint is added or an existing constraint is changed. A by-product of incrementality in search is a set of generated partial solutions for each increment that can be stored separately and later reused, where a new constraint can be added or an existing constraint can be changed without making too much distortion to the current solution. More importantly incremental ICHEA does not have to define any problem specific algorithm to get feasible solutions as many other approaches like [1, 6, 14] use bespoke algorithms or SD graph-coloring heuristics to get the feasible solutions.

4 Experiments for Discrete COPs

Hyper-heuristics have been frequently used to solve benchmark exam timetabling problems which show promising results [8, 14]. ICHEA is a meta-heuristic algorithm that uses multiple mutation strategies to optimize a CP as described in Section 3. All the benchmark problems have been experimented on a Windows 7 machine with Pentium (R) i5 CPU 2.52 GHz and 3.24 GB RAM except the problem Pur93 which was run on a server machine (Intel Xeon CPU 2.90GHz and 128 GB RAM) because of its size and memory requirements. We ran all the problems overnight because of their size and complexity. Additionally, real world timetabling problem does not required to be solved within minutes or hours [1, 14]. Even though smaller sized problems like Hec92 and Sta83 can be solved within an hour or two; however problem Pur93 had to be run for almost 24 hours because of its huge size.

Table 1. Statistical summary of results from IICHEA and ICHEA

Instance	Best	Median	Worst	SD
Car91	4.91 (5.1)	5.04 (5.3)	5.16 (5.46)	0.01 (0.15)
Car92	4.08 (4.3)	4.1 (4.45)	4.2 (4.54)	0.05 (0.10)
Ear83	33.24 (33.6)	34.02 (34.69)	34.7 (37.39)	0.57 (1.24)
Hec92	10.13 (10.17)	10.33 (10.45)	10.61 (11.15)	0.15 (0.37)
Kfu93	13.58 (13.8)	13.8 (14.1)	14.21 (15.09)	0.20 (0.37)
Lse91	10.37 (10.95)	10.51 (11.34)	10.67 (11.8)	0.11 (0.27)
Pur93	4.67 (5.2)	4.78 (5.43)	4.99 (5.81)	0.12 (0.21)
Rye92	8.63 (9.07)	8.76 (9.4)	8.85 (9.7)	0.08 (0.19)
Sta83	157.03 (157.03)	157.03 (157.03)	157.03 (157.03)	0.0 (0.0)
Tre92	8.33 (8.8)	8.5 (9.3)	8.8 (9.6)	0.16 (0.28)
Uta92	3.28 (3.48)	3.41 (3.60)	3.57 (3.64)	0.07 (0.07)
Ute92	24.85(24.9)	24.9 (25.7)	25.1 (27.0)	0.10 (0.87)

All the experimental results have been verified through the standard evaluator program available in the dedicated website for research on benchmark exam timetabling problems [23]. We executed each problem for 10 trials to get SRs and establish statistical evaluation in Table 1.

Many times incremental approach to solve a complex static CP gives better results than solving entire constraints altogether. We used both approaches in the experiments to demonstrate supremacy of one approach over another. We observed that this incremental approach also helps in quickly providing feasible partial solutions and eventually feasible solutions at the SR of 100% for all the benchmark problems; whereas SRs of non-incremental ICHEA are very low for bigger problems like Car91 and Uta92 have only 0%-10% of SR, and 30%-70% for other problems of medium size. Non-incremental ICHEA also takes much longer duration to get the first feasible solution. The unpromising outcome from non-incremental ICHEA has led us to do the experiments with incremental ICHEA only. We first sort the constraints (exams clashes) according to LD then remove first 5% of the total exams as input for each increment in ICHEA. *Intermarriage* crossover constructs new partial feasible solutions which are then optimized using mutation strategies for feasible partial solutions. We used two instances of ICHEA for the experiments to demonstrate the validation of incrementality. The first and second instances of ICHEA optimize the partial solutions for 0 and 500 generations respectively. The only difference between these two instances is the first one does not apply optimization strategies to partial solutions while the other optimizes the partial solution for 500 generations. However, both instances get the feasible solutions incrementally. To distinguish the two instances the first one is called ICHEA and second one is called incremental ICHEA (IICHEA) as it fully exploits the notion of incrementality.

The statistical results of IICHEA and ICHEA on all the problems from University of Toronto benchmark exam timetabling problems (version I) from [18, 23] are shown in Table 1. We only used version I because it has been mostly reported in the literature. ICHEA results are in the brackets. We also compared our best solutions with other published results from [1, 5, 8, 10, 11, 14–16, 22] cited frequently in the literature in Table 2.

Table 2. Best results from the literature compared with IICHEA

Algorithms	Car91	Car92	Ear83	Hec92	Kfu93	Lse91	Pu93	Rye92	Sta83	Tre92	Uta92	Ute92	Yor83
IICHEA	4.9	4.1	33.2	10.1	13.6	10.4	4.7	8.6	157.0	8.3	3.3	24.8	36.2
[15]	7.1	6.2	36.4	10.8	14.0	10.5	3.9	7.3	161.5	9.6	3.5	25.8	41.7
[31]	5.1	4.3	35.1	10.6	13.5	10.5	-	8.4	157.3	8.4	3.5	25.1	37.4
[16]	5.4	4.4	34.8	10.8	14.1	14.7	-	-	*134.9*	8.7	-	25.4	37.5
[52]	4.5	3.9	33.7	10.8	13.8	10.4	-	8.5	158.4	7.9	3.1	25.4	36.4
[1]	5.2	4.4	34.9	10.3	13.5	10.2	-	8.7	159.2	8.4	3.6	26.0	36.2
[26]	5.2	4.3	36.8	11.1	14.5	11.3	-	9.8	157.3	8.6	3.5	26.4	39.4
[9]	4.6	3.8	32.7	10.1	12.8	9.9	4.3	7.9	157.0	7.7	3.2	27.8	34.8
[13]	4.9	4.1	33.2	10.3	13.2	10.4	-	-	156.9	8.3	3.3	24.9	36.3
[23]	4.5	3.8	32.5	10.0	12.9	10.0	5.7	8.1	157.0	7.7	3.1	24.8	34.6

5 Discussion

To solve a CSP ICHEA works with allele coupling only. So only the definition of constraints and the rules for coupling of two constraints need to be provided for *intermarriage* crossover. Experimental results for benchmark exam timetabling problems for COPs are very promising. Results for problems *Ear83*, *Hec92*, *Sta83*, *Tre92*, *Ute92* are in top three and other results are also in the upper half of the best results. It is noted that IICHEA has been giving consistent results for all the problems. It is noted that exam timetabling problems show good results with hyper-heuristics. Using the incrementality technique of IICHEA on these hyper-heuristics can produce even better results as shown in the comparative results between ICHEA and IICHEA. Incrementality in ICHEA produces better results than without incrementality. Consequently, IICHEA can also be used for real time discrete COPs. IICHEA also does not require having a separate problem specific algorithm to get feasible solutions as a preprocessor for constraint optimization. It has found feasible solutions for all the problems at the SR of 100%.

6 Conclusion

This paper focuses on incorporating ICHEA for solving discrete COPs. ICHEA has been designed as a generic framework for evolutionary search that extracts and exploits information from constraints. ICHEA has shown promising results experimented on benchmark exam timetabling problems. We proposed another version of *intermarriage* crossover operator for discrete CSPs to get the feasible solutions. Constraint optimization requires additional optimization techniques that are not all generic in its current form. ICHEA uses many problem specific mutation strategies to optimize exam timetabling problems. A major experimental observation was realizing the efficacy of incrementality in evolutionary search. Incrementality helps in getting feasible solutions with SR of 100% that also produces solutions of better quality. Incremental ICHEA can also be used for real time dynamic COPs in discrete domain. The competitive results from ICHEA shows its potential in making a generic evolutionary computational model that discovers information from constraints.

References

1. Abdullah, S., et al.: Investigating Ahuja–Orlin's large neighbourhood search approach for examination timetabling. Spectr. 29(2), 351–372 (2006)
2. Brailsford, S.: Constraint satisfaction problems: Algorithms and applications. Eur. J. Oper. Res. 119(3), 557–581 (1999)
3. Burke, E., et al.: A Time-Predefined Local Search Approach to Exam Timetabling Problems, vol. 1153, pp. 76–90 (2003)
4. Burke, E., et al.: Hybrid Graph Heuristics within a Hyper-Heuristic Approach to Exam Timetabling Problems (2005)

5. Burke, E., Bykov, Y.: A Late Acceptance Strategy in Hill-Climbing for Exam Timetabling Problems. Presented at the PATAT 2008 Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (2008)
6. Burke, E.K., Newall, J.P., Weare, R.F.: A memetic algorithm for university exam timetabling. In: Burke, E.K., Ross, P. (eds.) PATAT 1995. LNCS, vol. 1153, pp. 241–250. Springer, Heidelberg (1996)
7. Burke, E.K., et al.: Adaptive selection of heuristics for improving exam timetables. Ann. Oper. Res., 1–17 (2012)
8. Burke, E.K., et al.: Hybrid variable neighbourhood approaches to university exam timetabling. Eur. J. Oper. Res. 206(1), 46–53 (2010)
9. Caramia, M., Dell'Olmo, P., Italiano, G.F.: New Algorithms for Examination Timetabling. In: Näher, S., Wagner, D. (eds.) WAE 2000. LNCS, vol. 1982, pp. 230–241. Springer, Heidelberg (2001)
10. Carter, M.W., et al.: Examination Timetabling: Algorithmic Strategies and Applications. J. Oper. Res. Soc. 47(3), 373 (1996)
11. Casey, S., Thompson, J.: GRASPing the Examination Scheduling Problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 232–244. Springer, Heidelberg (2003)
12. Craenen, B.G.W., et al.: Comparing evolutionary algorithms on binary constraint satisfaction problems. IEEE Trans. Evol. Comput. 7(5), 424–444 (2003)
13. Craenen, B.G.W.: Solving constraint satisfaction problems with evolutionary algorithms. Phd Dissertation, Vrije Universiteit (2005)
14. Demeester, P., et al.: A hyperheuristic approach to examination timetabling problems: benchmarks and a new problem from practice. J. Sched. 15(1), 83–103 (2012)
15. Eley, M.: Ant algorithms for the exam timetabling problem. In: Burke, E.K., Rudová, H. (eds.) PATAT 2007. LNCS, vol. 3867, pp. 364–382. Springer, Heidelberg (2007)
16. Merlot, L.T.G., Boland, N., Hughes, B.D., Stuckey, P.J.: A Hybrid Algorithm for the Examination Timetabling Problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 207–231. Springer, Heidelberg (2003)
17. Müller, T.: Constraint-based Timetabling. PhD Dissertation, Charles University (2005)
18. Qu, R., et al.: A survey of search methodologies and automated system development for examination timetabling. J. Sched. 12(1), 55–89 (2008)
19. Shang, Y., Fromherz, M.P.J.: Experimental complexity analysis of continuous constraint satisfaction problems. Inf. Sci. 153, 1–36 (2003)
20. Sharma, A., Sharma, D.: ICHEA for Discrete Constraint Satisfaction Problems. In: Thielscher, M., Zhang, D. (eds.) AI 2012. LNCS, vol. 7691, pp. 242–253. Springer, Heidelberg (2012)
21. Sharma, A., Sharma, D.: Solving Dynamic Constraint Optimization Problems Using ICHEA. In: Huang, T., Zeng, Z., Li, C., Leung, C.S., et al. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 434–444. Springer, Heidelberg (2012)
22. Yang, Y., Petrovic, S.: A Novel Similarity Measure for Heuristic Selection in Examination Timetabling. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 247–269. Springer, Heidelberg (2005)
23. Benchmark Exam Timetabling Datasets, `http://www.cs.nott.ac.uk/~rxq/data.htm`