Tabu Search for Constraint Solving and Its Applications

Jin-Kao Hao
LERIA
University of Angers
2 Boulevard Lavoisier
49045 Angers Cedex 01 - France

1. Introduction

The Constraint Satisfaction Problem (CSP) is a very general problem able to conveniently formulate a wide range of important applications in combinatorial optimization. These include well-known classical problems such as graph k-coloring and satisfiability, as well as many practical applications related to resource assignments, planning and timetabling. The feasibility problem of Integer Programming can also be easily represented by the CSP formalism.

As elaborated below, we provide case studies for specific CSP applications, including the important realm of frequency assignment, and establish that tabu search is the best performing approach for these applications.

The Constraint Satisfaction Problem is defined by a triplet (X, D, C) where:

- $X = \{x_1, \cdots, x_n\}$ is a finite set of n variables.
- $D = \{Dx_1, \cdots, Dx_n\}$ is a set of associated domains. Each domain Dx_i specifies the finite set of possible values of the variable x_i.
- $C = \{C_1, \cdots, C_p\}$ is a finite set of p constraints. Each constraint is defined on a set of variables and specifies which combinations of values are compatible for these variables.

Given such a triplet, the basic problem consists in finding a complete assignment of the values to the variables such that that all the constraints are satisfied. Such an assignment is called a solution to the given CSP instance. Since the set of all assignments is given by the Cartesian product $Dx_1 \times \cdots \times Dx_n$, solving a CSP means to determine a particular assignment among a potentially huge search space. The CSP is known to be a NP-hard problem in the general case.

A popular conventional approach for solving a CSP is based on systematic tree search strategies combined with various domain reduction techniques. This approach has been intensively investigated by the Constraint Programming (CP) community since 1986 and implemented in many contemporary CP systems. Nevertheless, exact methods based on systematic tree search may fail to solve large CSP instances in practice, because the computing time required may become prohibitive.

We have investigated Tabu Search for solving various hard CPS problems and have found it to be highly effective [1-6], thereby providing a valuable alternative to the classical reliance on systematic tree search. Our general TS procedure, TabuCSP, is composed of a number of standard key elements [7]. A *configuration* of the search space is identified by a complete assignment of values to each problem variable. A *penalty-based evaluation function* is defined to quantify the degree of the constraint violation of a given configuration. A *conflict-guided one-change or swap move* operator is employed to generate neighboring configurations. An attribute-based *tabu list* is implemented to prevent the search from revisiting previously encountered solutions. Optionally, strategies for reinforced diversification and intensification are introduced for improved search capacity.

To illustrate the advantages of such an approach, we consider below two case studies on two well-known problems: the Progressive Party Problem (PPP) [3] from the CSPLib (http://www.csplib.org/) and the Frequency Assignment Problem [4]. Additional examples of applying TabuCSP and its variants are provided in references [1,2,5,6] including Sports League Scheduling and Graph Coloring.

2. Case Study 1: Progressive Party Problem

The Progressive Party Problem (PPP) appeared initially during a yacht rally (Bembridge Yacht Rally, Island of Wight) and aimed at the organization of a party. In this problem, there are a given number H of host boats that invite the G guest crews of other boats on their board. The size c(g) of each guest crew g in [1..G] and the capacity C(h) of each boat h in [1...H] are given. The party, that lasts for T successive time periods (each time period lasts half hour), is organized according to a set of hard constraints:

- The capacities of the host boats must be respected.
- Each guest crew must move to a different boat for each different time period.
- Two guest crews can meet at most once.

The party organizers need to elaborate a valid plan that assigns to each guest crew a host boat for each time period such that each guest crew is assigned a host boat for each time period while respecting the given constraints.

In the initial problem instance, there are G = 29 crews and H = 13 boats, the party is organized for T = 6 time periods. However the problem can be defined for any value $6 \leq T \leq 10$ (given the constraints, 10 is an upper bound for the time periods). A larger T implies a problem more difficult to solve. Clearly, PPP is a constraint satisfaction problem.

PPP is known to be very challenging for Integer Linear Programming (ILP). Constraint programming (CP) proves to be a more appropriate approach leading to much better results. Yet as shown in [3], Tabu Search remains the best performing approach for solving this problem. The following Table extracted from [3] shows the best known results reported in the literature till 2000 with three different methods: ILP (1996, 2003), CP (1996, 1997) and TabuCSP (1998) where P_T denotes PPP with a time period T = 6 to 10.

problem	ILP	*CP*		*TabuCSP*	
		CP1	CP2	T[sec]	Iter
P_6	fail	27 min.	a few sec.	0.3 sec.	210
P_7	fail	28 min.	a few sec.	0.5 sec.	330
P_8	fail	fail	a few sec.	1.7 sec.	1,366
P_9	fail	fail	several hours	67.5 sec.	51,507
P_{10}	fail	fail	fail	fail	

From the table, we see that the first ILP[1] fails to solve any of P_6 to P_{10}. An improved ILP[2] (not shown in the Table) manages to solve the basic problem P_6 with considerable computing time

[1] S. C. Brailsford, P. M. Hubbard, B. M. Smith, The progressive party problem: a difficult problem of combinatorial optimization. *Computers & Operations Research*, 23:845-8 56, 1996.

[2] E. Kalvelagen, On solving the progressive party problem as a MIP. *Computers & Operations Research* 30(11):1713-1726, 2003.

(1.5 hours on a PC with a 1.2 GHz AMD Athlon processor, 512 MB Ram running Linux). The results of CP indicated by CP1[3] and CP2[4] are much more better. CP1 solves P_6 and P_7 in less than half an hour, but fails to solve larger problems. CP2 performs better: P_6 to P_8 are solved in a few seconds. The problem P_9 is also solved, but using several hours (and millions of backtracks).

Using TabuCSP, we solve the problem up to 9 time periods. P_6 to P_8 are solved very easily in less than two seconds (on a Sun ULTRA 1, 128 RAM, 134 MHz) with a few hundreds of iterations while P_9 is solved in about one minute with a few thousands of iterations. It is still an open question whether a solution exists for 10 time periods. However, TabuCSP frequently finds configurations involving only one violated constraint.

3. Case study 2: Frequency Assignment Problem

The Frequency Assignment Problem (FAP) is one of the key applications in GSM mobile networks engineering. Although different versions of the FAP can be defined, the main purpose is to assign a limited number of available frequencies to each cell in a mobile radio network while minimizing electro-magnetic interference due to the re-use of frequencies. The difficulty of this application comes from the fact that an acceptable solution of the FAP must satisfy a set of multiple constraints, which impose conflicting objectives. The most severe constraint concerns a very limited radio spectrum consisting of a small number of frequencies (or channels). Telecommunications operators must operate their networks typically with only several tens of frequencies, whatever the traffic volume to be covered. This constraint imposes a high degree of frequency re-use, which in turn increases the probability of frequency interference. Interference occurs when two frequencies assigned to a same cell or two adjacent cells are not sufficiently separated.

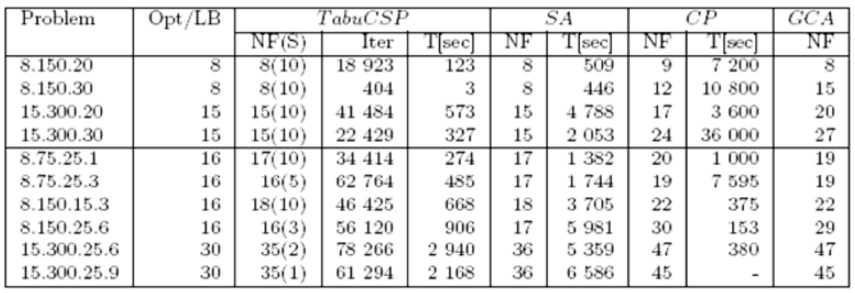
Consequently, frequency planning must take into consideration these interference constraints which state that frequencies assigned to some cells must satisfy a given separation distance in the frequency domain.

- *co-cell constraint*: any pair of frequencies assigned to a radio cell must have a certain distance between them in the frequency domain.
- *adjacent-cell constraint*: any pair of frequencies assigned to two adjacent cells must be sufficiently separated in the frequency domain.

The basic FAP can be shown to be NP-hard in its simplest form because it is reduced to the graph coloring problem. More generally, the problem is equivalent to the so-called "set T-coloring problem". Many heuristic methods have been devised to tackle the FAP, including greedy graph coloring algorithms (GCA), constraint programming (CP), simulated annealing (SA), artificial neural networks, evolutionary algorithms and Tabu Search.

The following table extracted from [4] shows a comparison between TabuCSP with three other methods applied to a set of FAP instances from France Telecom. The column NF(S) is the main quality measure and shows the number of frequencies used to obtain an interference-free frequency assignment with the number of hits indicated in parenthesis.

[3] B. M. Smith, S. C. Brailsford, P. M. Hubbard, H. P. Williams, The Progressive party problem: integer linear programming and constraint programming compared. *Constraints*, 1(1/2):119-138, 1996.
[4] N. Beldiceanu, E. Bourreau, P. Chan, D. Rivreau, Partial search strategy in CHIP. Presented at *Metaheuristics International Conference*, Sophia-Antipolis, France, 1997.



Problem	Opt/LB	*TabuCSP*			*SA*		*CP*		*GCA*
		NF(S)	Iter	T[sec]	NF	T[sec]	NF	T[sec]	NF
8.150.20	8	8(10)	18 923	123	8	509	9	7 200	8
8.150.30	8	8(10)	404	3	8	446	12	10 800	15
15.300.20	15	15(10)	41 484	573	15	4 788	17	3 600	20
15.300.30	15	15(10)	22 429	327	15	2 053	24	36 000	27
8.75.25.1	16	17(10)	34 414	274	17	1 382	20	1 000	19
8.75.25.3	16	16(5)	62 764	485	17	1 744	19	7 595	19
8.150.15.3	16	18(10)	46 425	668	18	3 705	22	375	22
8.150.25.6	16	16(3)	56 120	906	17	5 981	30	153	29
15.300.25.6	30	35(2)	78 266	2 940	36	5 359	47	380	47
15.300.25.9	30	35(1)	61 294	2 168	36	6 586	45	-	45

We notice that TabuCSP outperforms CP and GCA on these instances in terms of the quality of solutions. Moreover, TabuCSP outperforms the SA algorithm on 4 instances and does as well as SA on the other instances (In terms of computing time, TabuCSP is always faster than SA even for solutions of better quality).

4. Concluding remarks

Our experiences with Tabu Search as well as findings reported by several other researchers show that Tabu Search is a very effective and flexible method for solving hard constraint satisfaction problems and more generally constrained combinatorial search problems. The adaptation of this methodology to a particular CSP instance can be realized in a relatively straightforward way, by focusing on the design of the evaluation function and the neighborhood relations. Tabu Search is also a valuable component when designing hybrid search algorithms using the memetic framework where Tabu Search is well qualified to ensure effective intensification of the search algorithm.

Beyond solving particular CSP problems, Tabu Search can also be used as a general problem solver for constraint satisfaction and constrained optimization problems. This is clearly demonstrated by Nonobe & Ibaraki[5] and in in [7]. This perspective seems somewhat overlooked and certainly deserves more research effort.

References

1. Philippe Galinier, Jin-Kao Hao: Tabu Search for Maximal Constraint Satisfaction Problems. Lecture Notes in Computer Science 1330: 196-208, 1997.

2. Raphaël Dorne and Jin-Kao Hao, Tabu search for graph coloring, T-colorings and set T-colorings. "Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization", Chapter 6, pp77-92, S. Voss, S. Martello, I.H. Osman and C. Roucairol (Eds.), Kluwer, 1998.

3. Philippe Galinier and Jin-Kao Hao, Solving the progressive party problem by local search. "Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization", Chapter 29, pp418-432, S. Voss, S. Martello, I.H. Osman and C. Roucairol (Eds.), Kluwer, 1998.

[5] K. Nonobe, T. Ibaraki, A tabu search approach to the constraint satisfaction problem as a general problem solver. European Journal of Operational Research 106:599-623, 1998.

4. Jin-Kao Hao, Raphaël Dorne, Philippe Galinier: Tabu Search for Frequency Assignment in Mobile Radio Networks. J. Heuristics 4(1): 47-62 (1998)

5. Philippe Galinier and Jin-Kao Hao, Hybrid evolutionary algorithms for graph coloring. Journal of Combinatorial Optimization. 3(4): 379-397, 1999.

6. Jean-Philippe Hamiez, Jin-Kao Hao: Solving the Sports League Scheduling Problem with Tabu Search. A. Nareyek (Ed.) Local Search for Planning and Scheduling 2000, Lecture Notes in Computer Science 2148: 24-36, 2000.

7. Philippe Galinier and Jin-Kao Hao, A general approach for constraint solving by local search. Journal of Mathematical Modelling and Algorithms 3(1): 73-88, 2004.