An Efficient Local Search algorithm for Over-Constrained CSPs

Jufang Li, Yan Chi
School of Management, Hunan International Economics University, Changsha, 410205, China
lijf@hieu.edu.cn

Abstract. In industrial experiences of solving constraint satisfaction problems (CSPs), we may often meet an over-constraint situation, which means that we cannot find a solution that satisfies all the constraints. In such situations, the classical CSP is usually extended to take into account costs, preferences, etc. The new objective is usually to minimize the total constraint violation cost or other criterions. The new problems can be described as valued constraint satisfaction problems (VCSP). VCSP can hardly be resolved by the classical constraint satisfying techniques, and this paper provides an application of the Guided Local Search in solving VCSPs.

1. Introduction

Theoretical CSP models, such as job-shop problems and traveling salesman problems, often have well defined problem structures. It can be asserted beforehand that the problems must have some feasible solutions. But when putting a real problem in the CSP framework, it may be very difficult to estimate whether there exists some feasible solutions or not. Sometimes there is not any solution that satisfies all the problem constraints. Such an over-constrained situation can be solved as a Valued Constraint Satisfaction Problem (VCSP) [1].

In the VCSP framework, each constraint of a classical CSP is annotated with a valuation (usually a number) denoting the impact or cost of its violation. Hard constraints representing those physical properties of a problem that must be satisfied can be associated with a big value. While soft constraints representing those properties that should be satisfied when possible can be associated with other proper values. Then the optimization objective of a VCSP would be trying to find a solution that minimizes the violated constraints’ total value [2].

Although VCSP is a very important framework for handling many real problems, it still not arouses much attention. Some versions of branch and bound or other complete methods have been suggested for tackling it. But they are especially limited by the combinatorial explosion problem since the constraint violation combinations for selective are made even more [3]. So we try to use incomplete methods to solve it, and we found the Guided Local Search (GLS) more fitted among the heuristic search methods we have known.

2. Valued Constraint Satisfaction Problem

Formally speaking, whereas a CSP is defined as a triple $P = (V,D,C)$, where $V$ is a set of variables, $D$ a set of associated domains, and $C$ a set of constraints between variables, a VCSP can be defined as a CSP, which is extended with [4]:

[1] [2] [3] [4]
● a valuation structure \( S \), which is itself a quintuple \( (E, \phi, T, \bot, \otimes) \), where \( E \) is a valuation set, \( \phi \) a total order on \( E \), \( T \) and \( \bot \) the maximum and the minimum elements in \( E \), and \( \otimes \) a binary closed aggregation operator on \( E \), which satisfies the following properties: commutativity, associativity, monotonicity according to \( \phi \), \( T \) as absorbing element and \( \bot \) as identity;

● a valuation function \( \varphi \) from \( C \) to \( E \).

The valuation set \( E \) is used to define a gradual notion of constraint violation and inconsistency. The elements of \( E \) can be compared using the total order \( \phi \) and aggregated using the operator \( \otimes \). The maximum element \( T \) is used to express imperative constraint violation and complete inconsistency, the minimum element \( \bot \) to express constraint satisfaction and complete consistency. The valuation function \( \varphi \) associates with each constraint a valuation, which denotes its importance (the valuation of any imperative constraint equals \( T \)).

Let \( A \) be an assignment of the problem variables and \( C_{\text{unsat}}(A) \) be the set of the problem constraints unsatisfied by \( A \). The valuation \( \varphi(A) \) of \( A \) is the aggregation of the valuations of all the constraints in \( C_{\text{unsat}}(A) \):

\[
\varphi(A) = \bigotimes_{c \in C_{\text{unsat}}(A)} \varphi(c)
\]

Then the goal is to find an assignment, whose valuation is lower than \( T \) and minimum. The valuation of the problem is the valuation of such an assignment. The global valuation of a partial assignment is the minimum valuation of all its complete extensions (on all the problem variables). Note that finding an optimal assignment or an optimal consistent relaxation are two equivalent problems.

3. A Local Search Algorithm for Solving VCSP

Many heuristic search methods are based on simple local search procedure. Voudouris and Tsang\(^5\) have given a local search algorithm for solving partial constraint satisfaction problems (PCSP). Since PCSP can be seen as another form of VCSP, we directly use the scheme to solve our VCSPs. In their algorithm, a move will only change one variable’s value at a time. Variables are examined in an arbitrary static order which starts form a random and complete assignment. Each time a variable is examined, the current value of the variable will change to the value which yields the minimum value for the cost function. Ties are randomly resolved allowing moves which transit to solutions with equal cost. These moves will enable local search to examine plateau of states occurring in the landscapes of many CSPs. In such a situation, a local minimum is concluded if all variables have been examined and no moves would result in a better solution.

This simple local search procedure will be iteratively called as the basis for the following described GLS.

4. Main Ideas of GLS

GLS is a stochastic method which is first presented by Chris Voudouris and Edward Tsang. It is a general and compact optimization technique suitable for a wide range of combinatorial optimization problems, such as Traveling Salesman Problem and Quadratic Assignment Problem\(^6\).

The method takes advantage of problem and search related information to guide local search in a search space. This is made possible by augmenting the cost function of the problem to include a set of penalty terms. Local search is confined by the penalty terms and focuses attention on promising regions of the search space. Iterative calls are made to local search. Each time local search gets caught in a local minimum, the penalties are modified and local search is called again to minimize the modified cost function. Penalty modifications regularize the solutions generated by local search to be in accordance to prior or gathered during search information. In fact, GLS objectives are similar to those of a well-informed tabu search algorithm except that, constraints in tabu search refer to moves and are usually implemented as lists, while constraints in GLS refer to some solution features and take
the form of penalties that augment the cost function \(^\text{[7,8]}\). The details of GLS can be found in \([6]\), we will not mention them again here.

5. Application of GLS in VCSP

When applying GLS to solve a problem, not only a basic local search procedure must be provided, but also a set of solution features must be defined to guide the local search procedure. Both are relied on the particular problem characters, thus ensure the GLS algorithm to be adapted to different problems. In section 3, the local search procedure used to solve a VCSP has been introduced. In the follows, we will focus on the selection of solution features and how they will be used to guide the search procedure. To facilitate understanding, we also give an over-constrained Map Coloring problem as an example.

5.1 An Over-Constrained Map Coloring Problem

A map-coloring problem involves choosing colors for the countries on a map in such a way that no two neighboring countries are the same color. For easy expressing, we consider six countries: Belgium, Denmark, France, Germany, Netherlands, and Luxembourg. They are individually denoted as variable \(B, D, F, G, N, \) and \(L\). The domains of each variable are the values that can be used to color the map. According to the countries’ neighboring relations, this reduced problem has the following eight constraints:

\[
\begin{align*}
B \neq F; & \quad D \neq G; \quad G \neq F; \quad B \neq N; \quad G \neq N; \quad L \neq F; \quad L \neq G; \quad B \neq L.
\end{align*}
\]

If the number of colors that can be used is not limited, the problem may have numerous solutions. But here we prescribe that at most two colors are used, which are denoted by 0 and 1. By solving it with an exact enumeration method, we find that the over-constrained situation appears, namely there has no solution that satisfies all the eight constraints. We will then solve it as a VCSP. Since Luxembourg is relatively small and not conspicuous, we assume that the three constraints related to \(L\) can be relaxed, and their valuations are low. By contraries, the other constraints are given high valuations meaning that they’d better not be violated. Then the optimization objective is to minimize the total valuation of the violated constraints.

The above over-constrained map coloring problem can be formally defined as such a VCSP:

- a set of variables: \(V = \{D, N, G, B, L, F\}\);
- associated domains: all variables have the same domain \(\{0,1\}\);
- a set of constraints: \(\mathcal{C} = \{c_1, c_2, \ldots, c_8\}\); among which,
  \[
  \begin{align*}
  c_1 : & \quad B \neq F; \\
  c_2 : & \quad D \neq G; \\
  c_3 : & \quad G \neq F; \\
  c_4 : & \quad B \neq N; \\
  c_5 : & \quad G \neq N; \\
  c_6 : & \quad L \neq G; \\
  c_7 : & \quad B \neq L; \\
  c_8 : & \quad L \neq F;
  \end{align*}
  \]
- a valuation function \(\varphi\); which give each constraint a valuation:
  \[
  \begin{align*}
  \varphi(c_1) = 100000; \quad \varphi(c_2) = 100000; \quad \varphi(c_3) = 100000; \quad \varphi(c_4) = 100000; \\
  \varphi(c_5) = 9043; \quad \varphi(c_6) = 568; \quad \varphi(c_7) = 257; \\
  \end{align*}
  \]
- an optimization objective which is to minimize the solution cost expressed as:
  \[
  \varphi(s) = \sum_k \varphi(c_k),
  \]
  in which \(k\) denotes the index of those constraints violated in solution \(s\).

5.2 Selection of Solution Features

Solution features are used to characterize solutions. A solution feature can be any solution property that satisfies a non-trivial constraint. This means that not all solutions have a particular property. The cost of features represents direct or indirect impact of the corresponding solution properties on the solution cost.
For VCSP, it is intuitive to define a solution feature as whether a particular constraint of the problem is violated or not in a solution \( s \). The cost of a solution feature is just the valuation of the associated constraint. We represent such a definition of solution feature as the following indicator function:

\[
I_{c_i}(s) = \begin{cases} 
1, & \text{if } c_i \text{ is violated in solution } s \\
0, & \text{if } c_i \text{ is satisfied in solution } s 
\end{cases}
\]

With this indicator function, the original solution cost can be further expressed as the following formula, in which \( m \) denotes the number of constraints in a VCSP. For the map coloring problem described above, \( m = 8 \).

\[
\varphi(s) = \sum_{i=1}^{m} I_{c_i}(s) \cdot \varphi(c_i)
\]

### 5.3 Augmented Cost Function

Restrictions on solution features are made possible by augmenting the cost function \( \varphi \) of a VCSP to include a set of penalty terms. The new cost function formed is called the augmented cost function and it is defined as follows:

\[
h(s) = \sum_{i=1}^{m} I_{c_i}(s) \cdot \varphi(c_i) + \lambda \sum_{i=1}^{m} I_{c_i}(s) \cdot p_{c_i} 
\]

\[
h(s) = \sum_{i=1}^{m} I_{c_i}(s) \left[ \varphi(c_i) + \lambda \cdot p_{c_i} \right]
\]

In the above formula, a penalty parameter \( p_{c_i} \) is introduced for each solution feature (each constraint). It gives the degree up to which the solution feature is restricted to appear in a solution. A regularization parameter \( \lambda \) is also introduced to represent the relative importance of penalties with respect to the solution cost. It is used to control the influence of the gradually got search related information on the search process.

When GLS iteratively calls the local search procedure, penalty parameters of some solution features are modified each time a local minimum is found, thus change the cost function. The modification may be simply incrementing a penalty parameter’s value by 1. Information of search process is gradually introduced into the augmented cost function since the modified penalty parameter is deliberately selected each time. Sources of such information are the cost of solution features and the local minimum itself. The following part describes how to modify the penalty parameters.

### 5.4 penalty modifications

A particular local minimum solution may exhibit a number of solution features. For VCSP, if the feature with regard to constraint \( c_i \) is exhibited by a local minimum solution \( s^* \), the corresponding indicator function \( I_{c_i}(s) \) will be set a value 1, else 0. Then the penalty parameters for all solution features that maximize the following utility expression will be incremented by one.

\[
util(s, c_i) = I_{c_i}(s) \cdot \frac{\varphi(c_i)}{1 + p_{c_i}}
\]

In other words, incrementing the penalty parameter of a feature is considered an action with utility given by formula (4). In a local minimum, the actions with maximum utility are selected and then performed. The utility function (4) makes full use of the information about what solution features appear in a local minimum and how important each solution feature is. It also incorporates the penalty parameter to prevent the scheme from being totally biased towards penalizing features of high cost, since the increment of a penalty parameter can be also seen as a counter of how many times the
corresponding feature has been penalized. Certainly, the penalty parameter can be incremented by larger step considering detail problems.

According to the above scheme, each time the local search procedure of the VCSP gets into a local minimum, the violation of some constraints may be penalized. Furthermore, violation of high valuation (more important) constraint may be penalized more frequently. While violation of low valuation constraints still have chance to be penalized if those important constraints have already been considered many times. In the short term, such a scheme may enable the local search to escape form a local minimum. While in the long term, the scheme may guide the local search to be more concentrated on satisfying high valuation (more important) constraints.

5.5 Solution of the Map Coloring Problem

Given the essentials of GLS algorithm for solving VCSP described above, we can easily solve the over-constrained map coloring problem now. Initially, penalty parameters of all the constraints are set to 0. An arbitrarily selected assignment that sets all countries’ color to 0 is used as the initial solution. Another important parameter that has been left out so far is the regularization parameter $\lambda$ in the augmented cost function (3). According to Christos Voudouris and Edward Tsang\[4\], $\lambda$ is problem dependent and should be comparable to the difference of the original objective function value between two successive local minimum. If it is too small, the local search will not be able to escape from local minima. Else if it is too large, the selected moves will solely remove the penalized features from the solution, thus introduces risks if information is wrong. So we initially set $\lambda$ to 80000 for our map coloring problem. We also find that GLS can be quite tolerant to the choice of the $\lambda$ value varying from 50000 to 100000.

Using a P4 2.4G PC and 100 iterations as the stopping criteria, the final result is returned within a second as follows:

\[ D = 0, N = 0, G = 1, B = 1, L = 0, F = 0. \]

This is in accordance with the original objective value 257. We find that this is just the optimal solution of the simple problem, only the most unimportant constraint $c_8$ is not satisfied.

In fact, GLS may be more powerful when used to solve larger scale VCSPs, since it is a kind of heuristic search method. But we haven’t done more experiment to compare GLS with other heuristics such as tabu search or others because of the lack of benchmark problems.

6. Conclusion

VCSP is a kind of combinatorial problems that may be frequently met in real problems while not causing much attention of researchers. It mainly deals with the situation when feasible solutions of the original problem do not exist or cannot be easily found. For such special situations, we have to relax some constraints to solve the problem, and the classical tree search method of CSP may be not applicable here since the number of “feasible” combinations grows, thus enlarges the search space. This is even distinct for those problems having a large size. At the same time, GLS is a relatively novel search technology. By introducing the concepts of solution features and penalties, it can use the information got during the former search to intelligently guide the following local search to travel to more promising areas in the search space.

In this paper, we only discuss the VCSP that must assign a value to all the variables even violating some constraints. In another situation, it may be happen that no constraints could be relaxed while some variables may be left unassigned a value. In such a situation, we may artificially add some soft constraints stating that each variable should be assigned a value. We will go on investigating such problems.

References

[1] T Schiex, H Fargier, G Verfaillie. 1995 Valued Constraint Satisfaction Problems: Hard and Easy Problems. 14th International Joint Conference on Artificial Intelligence, Montral
[2] Luc Jaulin. Solving set-valued constraint satisfaction problems. 2012 *Computing*, **94** 297
[3] Cohen D Cooper, Jeavons P. Binarisation for valued constraint satisfaction problems. 2015 *SIAM journal on discrete mathematics*, **31** 2279
[4] Grard Vé rfaillie, Simon de Givry. Algorithmic Problems and Solutions in the Valued Constraint Satisfaction Problem Framework. 1997 *Proceedings of the EUFIT-97 session on “Valued Constraint Satisfaction”*, Aachen, Germany
[5] C Voudouris, E Tsang. The Tunnelling Algorithm for Partial CSPs and Combinatorial Optimization Problems. 1994 *Technical Report CSM-213*, Department of Computer Science, University of Essex
[6] Christos Voudouris. Guided Local Search for Combinatorial Optimization Problems. 1997 *PhD Thesis*. Department of Computer Science, University of Essex
[7] Grabowski Jó zef, Wodecki Mieczysław. 2005 *A very fast tabu search algorithm for job shop problem. Metaheuristic optimization via memory and evolution*. Kluwer Academic Publishers
[8] R Christopher L Riley, Cesar Rego. 2018 Intensification, diversification, and learning via relaxation adaptive memory programming: a case study on resource constrained project scheduling. *Journal of Heuristics*, **24** 1.