Computing 39, 345–351 (1987)

Computing

Using Tabu Search Techniques for Graph Coloring

A. Hertz and **D. de Werra,** Lausanne

Received May 5, 1987

Abstract — Zusammenfassung

Using Tabu Search Techniques for Graph Coloring. Tabu search techniques are used for moving step by step towards the minimum value of a function. A tabu list of forbidden movements is updated during the iterations to avoid cycling and being trapped in local minima. Such techniques are adapted to graph coloring problems. We show that they provide almost optimal colorings of graphs having up to 1000 nodes and their efficiency is shown to be significantly superior to the famous simulated annealing.

Key words: Graph coloring, tabu search, simulated annealing.

Die Tabu-Methoden zur Graphenfärbung. Tabu-Methoden werden benützt, um schrittweise den minimalen Wert einer Funktion zu erreichen. Eine sogenannte Tabuliste von verbotenen Schritten wird während des Prozesses nachgeführt, so daß man im Algorithmus keine Zyklen hat und nicht in lokalen Minima gefangen wird. Solche Methoden werden auf Graphenfärbung angepaßt. Wir zeigen, daß man mit dieser Technik fast optimale Färbungen für Graphen mit bis zu 1000 Knoten erhält. Die Effizienz dieser Methoden ist viel besser als diejenige der berühmten „Simulated Annealing" Algorithmen.

1. Introduction

Various techniques have been described for obtaining colorings of the nodes of large graphs. The efforts made by many researchers have been motivated by a collection of applications going from cluster analysis to group technology in Computer Integrated Manufacturing.

Recently the famous simulated annealing technique [6] which had some success in dealing with large combinatorial optimization problems has been applied to the graph coloring problem [2].

Such a method may be helpful in approximating the minimum value of a so-called energy function by slowly reducing a parameter which is analogous to the thermodynamical temperature in a physical system of particles [6].

Many papers have been devoted to the application of this approach to famous combinatorial optimization problems. Nevertheless the analogy between a system of particles with a temperature and a combinatorial optimization problem is not always obvious and perhaps not always justified. In particular the definition of an objective function which may be assimilated to an energy function may cause some difficulties.

A few years ago a simple idea was suggested by Glover [4] for moving stepwise to the optimum value of an objective function with a special feature designed to avoid being trapped by local minima. This is the tabu search technique which has not been exploited yet on very many types of combinatorial optimization problems.

In the next section we shall sketch the basic ideas of tabu search and Section 3 will contain the adaptation of this technique to graph coloring. Section 4 will be devoted to a combined tabu method. A discussion of the results will be contained in Section 5.

2. Tabu Search Technique

In rough terms the tabu search method can be sketched as follows: we want to move step by step from an initial feasible solution of a combinatorial optimization problem towards a solution giving the minimum value of some objective function.

For this we may represent each solution by a point in some space and we have to define a neighbourhood $N(s)$ of each point s.

The basic step of the procedure consists in starting from a feasible point s and generating a sample (with fixed size rep) of solutions in $N(s)$; then we choose the best neighbour s^* generated so far and we move to s^* whether $f(s^*)$ is better than $f(s)$ or not.

Up to this point this is close to a local improvement technique except the fact that we may move to a worse solution s^* from s (this is a situation which occurs in simulated annealing where a move to a worse solution may be accepted with a probability which decreases when the number of completed iterations increases).

The interesting feature of tabu search is precisely the construction of a list T of tabu moves: these are moves which are not allowed at the present iteration. The reason for this list is to exclude moves which would bring us back where we were at some previous iteration. Now a move remains a tabu move only during a certain number of iterations, so that we have in fact a cyclical list T where at each move $s \rightarrow s^*$ the opposite move $s^* \rightarrow s$ is added at the end of T while the oldest move in T is removed from T.

In conclusion the basic step consists in generating randomly a fixed number rep of possible moves from s (whenever a move in T is generated, it is destroyed and a new move is generated). Then the best one of the generated moves is realized and the tabu list T is updated accordingly.

Now a stopping rule should be also defined: in general we may give a maximum number nbmax of iterations. In our case we will use an estimation f^* of the minimum value of the objective function $f(s)$. As soon as we are close enough to f^* (or when we have reached f^*) we may stop the whole procedure.

More refined versions of the tabu search are described in [3].

3. Application to Node Colorings

Let us now describe how the tabu search technique may be used to find colorings of large graphs. We will essentially try to find a coloring of a given graph G which uses a fixed number k of colors. Then we may vary k as we wish.

Given a graph $G=(V,E)$ a feasible solution will be a partition $s=(V_1, V_2, \ldots, V_k)$ of the node set V into a fixed number k of subsets. If $E(V_i)$ is the collection of edges of G with both endpoints in V_i, we may define the objective function f as the number of edges for which both endpoints are in the same V_i (i.e. have the same color):

$$f(s)=\Sigma\left(|E(V_i)|: i=1, \ldots, k\right).$$

Clearly s will be a coloring of the nodes of G with k colors if and only if $f(s)=0$. In fact we can estimate the best possible value of $f(s)$ with $f^*=0$; this will give us a stopping condition in the algorithm.

From s we generate a neighbour s' (i.e. another partition into k subsets of nodes) as follows: we choose a random node x among all those which are adjacent to an edge in $E(V_1) \cup \ldots \cup E(V_k)$. Then assuming $x \in V_i$, we choose a random color $j \neq i$ and we obtain s' from $s=(V_1, \ldots, V_k)$ by setting:

$$V_j'=V_j \cup \{x\}; \quad V_i'=V_i \backslash \{x\}; \quad V_r'=V_r \text{ for } r=1, \ldots, k; r \neq i, j.$$

Having generated rep neighbours of s (which do not lead to tabu moves), we pick up the best one and we move to it.

The tabu list is obtained as follows: whenever a node x is moved from V_i to V_j to get the new solution, the pair (x, i) becomes tabu: node x cannot be returned to V_i for some iterations. As described before the list T of tabu moves is cyclic.

Now we shall continue the iterations until either we get a solution s such that $f(s)=f^*$ or until we reach the maximum number nbmax of iterations. In this case, we will not have obtained a coloring if for the last solution s we have $f(s)>0$.

Following Glover [3] we are using a function $A(z)$ which is the aspiration level of the objective function value next to be reached when the current value is $z=f(s)$. It is used like this: if a move to a neighbour s' is tabu but gives $f(s') \leq A(z)$, then we drop the tabu status of this move and we consider it as a normal member of the sample which is generated.

Initially we set $A(z)=z-1$ for all values of z. Then whenever we generate an s' with $f(s') \leq A(f(s))$ then we set $A(f(s))=f(s')-1$.

The complete formulation of the tabu search technique for our coloring problem is given in Table 1; we will call it TABUCOL.

Several local improvements have been introduced for reducing the computation time.

First during the process of generating neighbours s' of s we may get at some stage an s' (not in the tabu list) with $f(s')<f(s)$. Instead of continuing until we have generated rep neighbours, we move directly from s to s'.

Table 1. *The TABUCOL algorithm*

```
Input     $G=(V,E)$
          $k$ = number of colors
          $|T|$ = size of tabu list
          rep = number of neighbours in sample
          nbmax = maximum number of iterations.
Initialization
          Generate a random solution $s=(V_1,\dots,V_k)$
          nbiter := 0; choose an arbitrary tabu list $T$.
While $f(s)>0$ and nbiter < nbmax
          generate rep neighbours $s_i$ of $s$ with move $s \to s_i \notin T$ or $f(s_i) \le A(f(s))$
          (as soon as we get an $s_i$ with $f(s_i)<f(s)$ we stop the generation).
          Let $s'$ be the best neighbour generated
          update tabu list $T$
          (introduce move $s \to s'$ and remove oldest tabu move)
          $s := s'$
          nbiter := nbiter + 1
endwhile
Output    If $f(s)=0$, we get a coloring of $G$ with $k$ colors: $V_1,\dots,V_k$ are the color sets. Otherwise no
          coloring has been found with $k$ colors.
```

Also when the edges in $E(V_1) \cup \dots \cup E(V_k)$ form a star (i.e. they have a common node), then we examine if a solution s^* with $f(s^*)=0$ can be reached by moving at most three nodes (among which this common node will be counted). This improvement is motivated by the fact that one may be at some step very close to an optimum solution and miss it due to the random choice of the next move.

This search for improvements in at most three moves is very fast to perform but could become time-consuming if we allowed more moves.

An important parameter is the size $|T|$ of the tabu list. As suggested by Glover [3] we have taken $|T|=7$. Smaller values may create cycling and larger values do not improve the procedure while increasing the computation time.

4. A Combined Method

As we observed for annealing, it turns out that the efficiency of tabu search for large graphs (more than 500 nodes) can be increased by combining it with other techniques [2].

The idea is the following:
Given a parameter q, we construct consecutively color sets (i.e. independent sets) $V_1, V_2, \dots$ which are as large as possible, until we are left with at most q nodes. Then these are colored by the TABUCOL procedure.

Finding a large independent set can also be done efficiently with the tabu approach.

We may proceed as follows for finding an independent set with p nodes: a feasible solution s is a partition $(S, \bar{S})$ of the node set with $|S|=p$. The function $f(s)$ to be minimized is $f(s)=|E(S)|$ i.e. the number of edges inside S.

Let us denote by $N_S(x)$ the set of all neighbours of x in S. The set S (resp. $\bar{S}$) is ordered in such a way that if x is before y in S (resp. $\bar{S}$) then $|N_S(x)| \geq |N_S(y)|$ (resp. $|N_S(x)| \leq |N_S(y)|$).

We construct two tabu lists $T(S)$ and $T(\bar{S})$ of same cardinality $|T|$ which contain all the nodes which were moved in the last $|T|$ steps.

Here we do not proceed as in the general case by generating rep neighbours of the current solution s. We can work in a much simpler way: we generate the best move which does not involve tabu nodes.

This is done as follows:
At each step we choose the first node x in $S \backslash T(S)$ and the first node y in $\bar{S} \backslash T(\bar{S})$ and we obtain s' by setting $S=(S\backslash\{x\}) \cup \{y\}$ and $\bar{S}=(\bar{S}\backslash\{y\}) \cup \{x\}$; x (resp. y) enters the list $T(\bar{S})$ (resp. $T(S)$) while the oldest node in this list is removed.

For random graphs with n nodes and edge probability 0.5, Matula and Johri [5] have given estimations $\tilde{\alpha}(n)$ of the size of a largest independent set.

We use these in our construction of color sets $V_1, V_2, \ldots$ as follows: when we are left with a graph having n' nodes, we set $p \cong \tilde{\alpha}(n')$ and we run the tabu method until $f(s)=0$ or nbiter = nbmax. In the first case, we have found a new color class and we remove these nodes from the graph. In the second case, we decrease p by 1 and we reapply this tabu search. This is iterated until we get an independent set.

Finally, when the color sets $V_1, \ldots, V_k$ constructed leave an uncolored graph with at most q nodes we apply the TABUCOL procedure for the remaining nodes.

5. Numerical Results and Discussion

Computational experiments have been run on a CDC Cyber 170/855 with random graphs. The density (probability of presence of each edge) was 0.5.

We summarize in Table 2 the results obtained with TABUCOL. We give there the smallest value k for which all examples of the same size have been successful (i.e. a coloring has been obtained).

Table 2. *Computational results of TABUCOL*

n nb. of nodes	nb. of graphs in sample	$\tilde{X}(G)$	nb. of colors used	average CPU time (in sec.)	average nb. of iterations	rep
100	20	16	16	0.5	355	50
300	10	35	35	59.7	9826	170
500	5	50	51	3344.7	330818	250
1000	2	85	93	8719.2	279466	600

Comparisons with the simulated annealing technique (see Table 3) show that the tabu method provides colorings for smaller values of k than annealing and it also takes less CPU-time (see those in [2]).

Table 3. *Best k without failure in sample*

n nb. of nodes	100	300	500	1000
$\tilde{\chi}(G)$	16	35	50	85
k (combined tabu)	16	35	50	87
k (TABUCOL)	16	35	51	93
k (annealing)	16	36	54	98

Our best results are obtained with combined techniques. For graphs having up to 500 nodes we have obtained colorings with no more colors than the probabilistic estimation given in [5]. In fact, such a combined procedure based on the annealing method has already given for a random graph G with $n = 1000$ nodes a coloring with 91 colors [2]; this was an improvement in comparison to the pure annealing method (see Table 3).

Table 4 summarizes the results for the combined tabu method applied to the same graph G. The total number of colors is 87 which is close to the probabilistic estimation $\tilde{\varkappa}(G) = 85$ of the chromatic number $\varkappa(G)$ (see [5]).

Table 4. *Combined tabu for $n = 1000$ and $q = 500$*

	size $\lvert S \rvert$	nb. of sets of size $\lvert S \rvert$	nb. of colored nodes	nb. of colors used	time in sec.
Construction of large independent sets	15	3	511	37	successful searches 6417 unsuccessful searches 3000
	14	24			
	13	10			
TABUCOL	13	1	489	50	3837
	12	0			
	11	15			
	10	15			
	9	10			
	8	8			
	7	1			
		Totals	1000	87	13254

With an entirely different approach Bollobàs et al. [1] have obtained colorings with an average of 86.9 colors. We do not know any other procedure which constructs colorings with at most 87 colors in such graphs.

It is difficult to compare the times of our method and of the one of Bollobàs et al. [1]. Our approach has however a practical advantage; if we want a coloring with p colors where p is not too close to $\varkappa(G)$, we get it easily in a much shorter CPU-time (the same occurred for the annealing technique [2]).

Furthermore if for a given p, we do not get a coloring with p colors, then we know where the "bad" edges are and we have an idea of the edges which should be deleted in order to get the desired coloring.

In our codes we have not used any special data structure for representing the graph and the lists needed in the algorithm. It is likely that computing times can still be reduced; our purpose was simply to exploit the tabu method and to show that it can provide good heuristics for graph coloring.

Acknowledgments

This research was supported by the Swiss National Research Council Grant number 2.706-1.85.

The authors would like to thank E. Amaldi and S. Cohen for having suggested improvements in the code for tabu and for their active participation in the numerical experimentation.

References

[1] Bollobàs, B., Thomason, A.: Random graphs of small order. Random graphs '83. Annals of Discrete Mathematics *28*, 47–97 (1985).

[2] Chams, M., Hertz, A., de Werra, D.: Some experiments with simulated annealing for coloring graphs. EJOR *32*, 260–266 (1987).

[3] Glover, F.: Future paths for integer programming and links to artificial intelligence. CAAI Report 85-8, University of Colorado, Boulder CO (1985).

[4] Glover, F., McMillan, C., Novick, B.: Interactive decision software and computer graphics for architectural and space planning. Annals of Operations Research *5*, 557–573 (1985).

[5] Johri, A., Matula, D. W.: Probabilistic bounds and heuristic algorithm, for coloring large random graphs. Southern Methodist University, Dallas, Texas (1982).

[6] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A.: Equation of state calculations by fast computing machines. J. Chem. Phys. *21*, 1087–1092 (1953).

A. Hertz and D. de Werra
DMA-EPFL
CH-1015 Lausanne-Ecublens
Switzerland