Adaptive Tabu Tenure Computation in Local Search

Conference Paper · March 2008

DOI: 10.1007/978-3-540-78604-7_1 · Source: DBLP

3 authors, including:

Hakim Mabed
University of Franche-Comté

Alexandre Caminada
Université de Technologie de Belfort-Montbé…

Adaptive Tabu Tenure Computation in Local Search

I. Devarenne, H. Mabed, and A. Caminada

UTBM, SET Lab, 90010 Belfort Cedex, France
{isabelle.devarenne, hakim.mabed, alexandre.caminada}@utbm.fr

Abstract. Optimization methods based on complete neighborhood exploration such as Tabu Search are impractical against large neighborhood problems. Strategies of candidate list propose a solution to reduce the neighborhood exploration complexity. We propose in this paper a generic Tabu Search algorithm using adaptive candidate list strategy based on two alternate candidate lists. Each candidate list strategy corresponds to a given search phase: intensification or diversification. The optimization algorithm uses a Tabu list containing the variables causing loops. The paper proposes a classification of Tabu tenure managing in the literature and presents a new and original Tabu tenure adaptation mechanism. The generic method is tested on the k-coloring problem and compared with some best methods published in the literature. Obtained results show the competitiveness of the method.

1 Introduction

In this paper, we propose a generic Tabu Search based on Adaptive Candidate List strategy, noted ACL_TS. The method alternates the use of two candidate lists [22] corresponding to intensification and diversification phases of search. During the intensification phase, a variable v is chosen from the candidate list CL_I and moved. However, when a loop is detected, a variable belonging to an extended candidate list CL_D is selected during the next iteration. A loop appears when a given variable is chosen more then a given threshold α during the last M iterations. After loop detection, the variable causing the loop is made Tabu. The choice of the loop variable is then forbidden (neighborhood restriction) for a given number of iterations named Tabu tenure. Alternation of intensification and diversification operators allows controlling the concentration and the repartition of visited solutions in certain areas of search. The intensification operator is based on the selection of a variable from a limited set of critical variables. The diversification operator selects a different category of variables covering a broader set of variables of the problem. The working scheme of the generic method is given in figure 1.

ACL_TS is applied to the k-coloring problem [6, 7]. The problem consists, given an undirected graph, in coloring the nodes using only k colors in such manner to assign different colors to adjacent nodes. From optimization point of view, the objective is to minimize the number of conflicts due to reusing the

J. van Hemert and C. Cotta (Eds.): EvoCOP 2008, LNCS 4972, pp. 1–12, 2008.

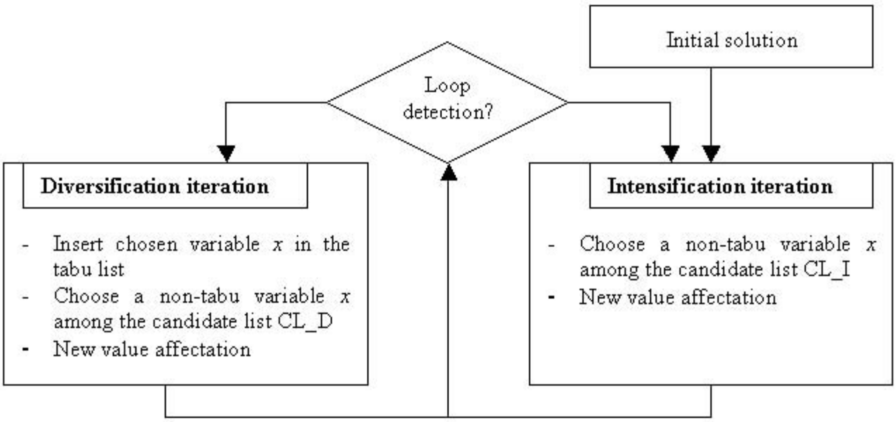


Fig. 1. Adaptive candidate list strategy

same color on linked nodes. During ordinary iteration, intensification operator chooses a node from the most conflicted nodes set. In the opposite, when a loop is detected, a conflicted node is randomly chosen. A node is said in conflict when it is colored with the same color than an adjacent node. In both cases the new assigned color is chosen among the best ones (less used color by the adjacent nodes).

Study performed in [6] shows that the method performances are very impacted by the Tabu tenure value. The Tabu notion was introduced first by Tabu Search method. Glover [13, 14] and Hansen and Jaumard [15], introduced a heuristic using a memory structure to exclude certain choices and restrict the neighborhood of the current solution to a subset of $V(s)$ also called accessible solutions. The Tabu list structure memorizes some information on past moves such as: solutions components, some moves or complete solutions. A Tabu tenure parameter has been introduced to prohibit some actions for a given number of iterations. In the literature there are different methods used to specify its value and its evolution during the search.

The rest of the paper is organized into 4 sections. In the first section, we present a classification of Tabu tenure specification approaches. Then in the second section, we present an adaptive mechanism for Tabu tenure calculation based on the number of visits of each variable. In the third section, we give analysis of the method results on the well-known DIMACS instances. Finally, a comparison with other famous works is made in section 4 and we conclude in the last section.

2 Tabu Tenure in the Literature

The Tabu tenure is a critical parameter that greatly influences the performance of the method. The duration of the prohibition period can be either static or dynamic. When the Tabu tenure is static, the value of Tabu duration is fixed throughout the search. In other words, the number of iterations for which the move is prohibited is fixed as in [16]. In the opposite, dynamic Tabu tenure varies during the execution. Several studies, such as those conducted by Hao et al. [17], have shown that a dynamic Tabu tenure could be more interesting than a static value.

We present different dynamic Tabu tenure computation used in the literature. All the presented methods use a Tabu list structure. Tabu tenure specification approaches are classified into four main method families: (a) approaches based on the running time, (b) approaches using a range of possible values, (c) approaches based on the current state of search, (d) approaches based on the historic of search.

2.1 Time Depending Tabu Tenure

In this approach, the Tabu tenure value is adjusted according to the spent time or to the current iteration number. The objective is then to progressively reduce the diversification level by decreasing the Tabu tenure value. This kind of approach is illustrated by Montemanni and Smith work [20] on the frequency assignment problem. The Tabu tenure is decreased during the run after each It_s iterations, independently of the search progression following the expression: $T = \beta \times T$, where β is a fixed value comprised in the interval $[0, 1[$. A value of β near to 1 allows to reduce slowly the Tabu tenure. The Tabu tenure T is equal to an initial value T_{init} at the beginning and to a minimum threshold T_{min} at the end. Parameters T_{min}, β et It_s are respectively fixed to 10, 0.96 and 5×10^4 for all tested FAP instances. Comparison made in [20] with three fixed Tabu tenure values shows that time depending Tabu tenure globally provides better results.

2.2 Random Bounded Tabu Tenure

In this second case, the Tabu tenure value is randomly chosen, at each iteration, inside a fixed interval. This principle is the most used in the literature and it is used as reference in this work. Intervals bounds are generally chosen according to some characteristics of the problem. For example, in Di Gaspero and Schaerf work [8], the Tabu tenure interval is calculated according to the number of variables N using the following expression: $[\frac{2}{3} \times N; N]$. Taillard [21] proposed a Tabu search method named *robust taboo search method* for quadratic assignment problem. In this method, Tabu tenure value is randomly chosen every $2 \times s_{max}$ iterations into the interval $[s_{min}; s_{max}]$, where $s_{min} = \lfloor 0.9 \times N \rfloor$ and $s_{max} = \lceil 1.1 \times N \rceil$[1]. Another example is given by Bachelet and Talbi [1] where the Tabu tenure is randomly chosen within the interval $[\frac{N}{2}; \frac{3 \times N}{2}]$.

2.3 Reactive Tabu Tenure

In the third approach, the Tabu tenure variation is a reaction to current solution state and no historic of search past is used. At each iteration, several pieces

[1] $\lfloor\rfloor$ and $\lceil\rceil$ represent respectively the integer part and integer part plus 1.

of information are extracted from the current solution and used to define the value of the prohibition period. Typically, these data designate the number of conflicted variables or the number of neighbors of the current solution.

Galinier and Hao [11] have proposed to increase the Tabu tenure according to the evaluation $F(s)$ of the current solution s. In order to do that, two parameters L (randomly chosen into the interval $[0;9]$) and λ (empirically fixed to 0.6) have been used as indicated in the following expression:

$$T = L + \lambda \times F(s) \tag{1}$$

In this work, Tabu tenure is maintained proportional to the cost function of the current solution. The given values for both L and λ parameters allow obtaining very good results. Although for other instances, parameter setting should be envisaged.

2.4 Adaptive Tabu Tenure

In the fourth approach, Tabu tenure is adjusted during the search according to the search history. We mention here three examples of adaptive Tabu tenure.

The first one is what Battiti et al. [2, 3] called *Reactive Tabu Search* method. The idea is to use a search memory composed by the previously visited solutions. After each move, the algorithm verifies if the current solution has already been found. If it is the case, the Tabu tenure T is increased. Otherwise the Tabu tenure value is decreased when no repetition is occurred during a sufficient long time. The Tabu tenure T is initialized to 1 at the beginning of search. When a solution is revisited, the Tabu tenure T is gradually increased using the following equation:

$$T = \min(\max(T \times 1.1, T + 1), L - 2) \tag{2}$$

where L represents the number of 0/1 variables. In the worst case, when the Tabu tenure is very high, two moves are possible. In [3], this method have been applied to the quadratic assignment problem. Comparison with Tabu search methods using static and interval based Tabu tenure shows very competitive results particularly for the highest size problems.

In the approach of Blöchliger [4] named *Approximated Cycle-Detection scheme (ACD)*, the method detects the cycles without storing the previously visited solutions. For that, a reference solution and a distance measurement are used. Every iteration, the distance between the reference solution and the current one is calculated. If this distance is equal to zero, a cycle is detected and the Tabu tenure is increased by an increment value η. Otherwise, it is slowly decreased. The parameter η varies during the search. Initialized to 5, it is incremented by 5 when a cycle is detected, and it is decremented by 1 every 15 000 iterations. The efficiency of the method depends on when the reference solution is updated and the relevance of used distance. The proposed method consists to update the reference solution when the current one is better in term of evaluation function and when the current solution is very far from the reference one.

In Consistent-Neighborhood Tabu (*CN-Tabu*) [10] method, the resolution is obtained by the exploration of partial solutions. Every iteration, a new assignment (x_i, v_i) is made, where x_i refers to an unassigned variable and v_i to the new assigned value. Each assignment (x_j, v_j) where x_j is an adjacent assigned variable is set Tabu and desinstantiated from the current solution. The Tabu tenure is dynamically calculated according to the number of times that the value v_j has been assigned to the variable x_j:

$$T(x_j, v_j) = nb_{mvt}(x_j, v_j) \tag{3}$$

where $T(x_j, v_j)$ is the prohibition period associated to the assignment (x_j, v_j), and $nb_{mvt}(x_j, v_j)$ is the number of times that the value v_j has been assigned to the variable x_j during the search.

Typically, the historic is not used to determine how much the Tabu tenure should be increased or decreased. Usually the historic provides only information on when the Tabu tenure needs to be adapted.

We propose here an adaptive Tabu tenure calculation where the value of Tabu tenure is adjusted separately for each decision variable. Memory structures and statistical data are used to determine when a decision variable is made Tabu and its prohibition period.

3 Adaptive Tabu Tenure in *ACL_TS* Method

In this section, we describe the process of Tabu tenure adaptation. In [6], we have compared dynamic and static Tabu tenure. In the dynamic case, the Tabu tenure value is randomly chosen into the interval $DT = [0.5 \times c(N); 1.5 \times c(N)]$ with N is the variables number and $c(N)$ the interval center. This variant is called $LD + DT$. Table 1 presents the results obtained by the $ACL\_TS : DT$ method combining loop detection and Tabu list for different Tabu tenure intervals on some Leighton instances (second DIMACS challenge instances[2]). These instances have been generated by Leighton [18]. All Leighton graphs used the same number of nodes (450) and the chromatic number is known. These instances are largely used in the literature. Three comparison criteria are retained: s the success rate over 10 runs, it the average number of iterations needed to resolve the instance and c the average conflicts number of the best solution found.

Tabu tenure interval is calculated according to the number of variables N. We observe that even if both instances presented in table 1 are composed of (450) nodes, the best results are not found by the same calculation function. All runs optimally solve the first problem with a Tabu tenure chosen in the first interval, whereas the second problem is never solved. Only the second settings allows to solve the second instance during 50% of runs. With the highest interval, both instances are never solved. Consequently, the instance size defined by the number of variables is not sufficient to determine the ideal value of Tabu tenure.

The dynamic adaptation of the Tabu tenure provides a serious alternative to statically determine the value or the set of values of the parameter. The used

[2] Available on `http://mat.gsia.cmu.edu/COLOR/instances.html`

Table 1. Basic ACL_TS (LD+DT): Influence of the interval of Tabu tenure

$f(N)$	$\left[0.5\times\frac{\sqrt{N}}{8};1.5\times\frac{\sqrt{N}}{8}\right]$			$\left[0.5\times\frac{\sqrt{N}}{7};1.5\times\frac{\sqrt{N}}{7}\right]$			$\left[0.5\times\frac{\sqrt{N}}{4};1.5\times\frac{\sqrt{N}}{4}\right]$		
DIMACS	s	it	c	s	it	c	s	it	c
le450_15a	<u>10</u>	<u>429 118</u>	0	5	147 775	2	0	-	4
le450_15b	0	-	1	<u>5</u>	<u>296 529</u>	2	0	-	4

method adjusts the Tabu tenure value according to the evolution of the search. The idea is to analyze the desired effect of the Tabu status. In fact, a good Tabu tenure value should prevent search cycles and should therefore be large enough to exclude variables provoking loops and orient the search toward new configurations.

We proposed to adapt Tabu tenure to each problem and to each variable of the problem by the use of variable search history. In previous work we have shown that the Tabu status is more efficient when it is applied only on variables provoking loops [6]. In addition, we aim to determine dynamically the Tabu tenure according to the number of loops provoked by each variable.

When a loop is detected the node *source* is made Tabu during a prohibition period calculated as follow:

$$AT(x_i) = \text{rand}(DT) + \frac{nbLoops(x_i)}{\sum\limits_{x_j \in V} nbLoops(x_j)} \times N \tag{4}$$

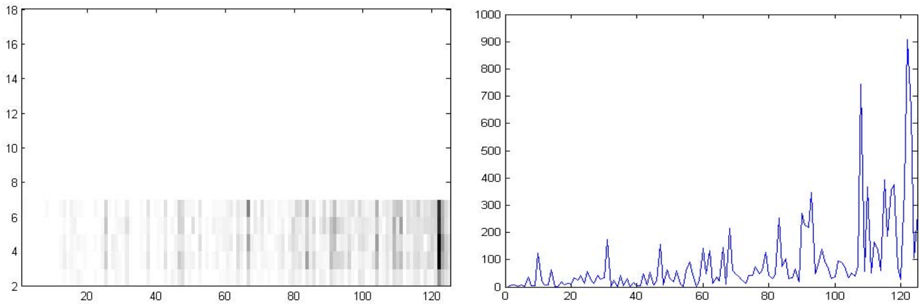
with $nbLoops(x_i)$ the number of loops provoked by the node x_i and DT the dynamic interval defined as follow:

$$DT = \left[0.5 \times \frac{\sqrt{N}}{2}; 1.5 \times \frac{\sqrt{N}}{2}\right] \tag{5}$$

This variant is noted *LD+AT* for Loop Detection and Adaptive Tabu tenure.

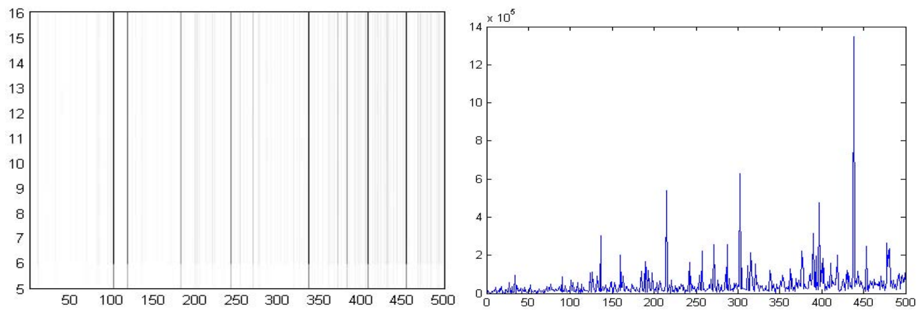
4 Study of the Tabu Tenure Repartition

This section presents the comparison between of the basic ACL_TS based on the LD+DT variant and the adaptive Tabu tenure (noted LD+AT) version. We observe here the Tabu tenure used for each node and the number of loops provoked per node. This study is made on some DIMACS instances: random graphs named DSJC$N.p$ when N is the number of nodes and p the probability that two nodes are joined. DSJC instances are also largely used in the literature. Figures 2 and 4 refers to a single successful run on the instance DSJC125.1. Figures 3 and 5 concerns the same not successful run on DSJC500.1. The two first figures correspond to the basic method LD+DT and the two others to the adaptive one LD+AT. Spectrograms (a) show the use frequency of each Tabu



(a) Tabu tenure used (b) Loops per node

Fig. 2. Method LD+TD instance DSJC125.1



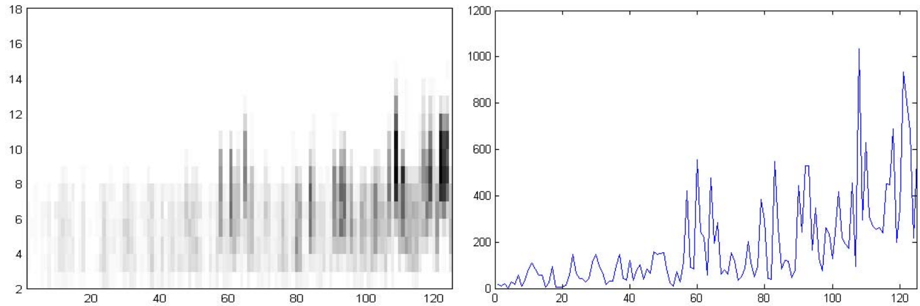
(a) Tabu tenure used (b) Loops per node

Fig. 3. Method LD+TD instance DSJC500.1

tenure value (y-axis) per node. Darker points correspond to most used values. Curves (b) represent the number of loops provoked by each node. In all curves or spectrograms, the x-axis represents the nodes classified by ascending order of degree.

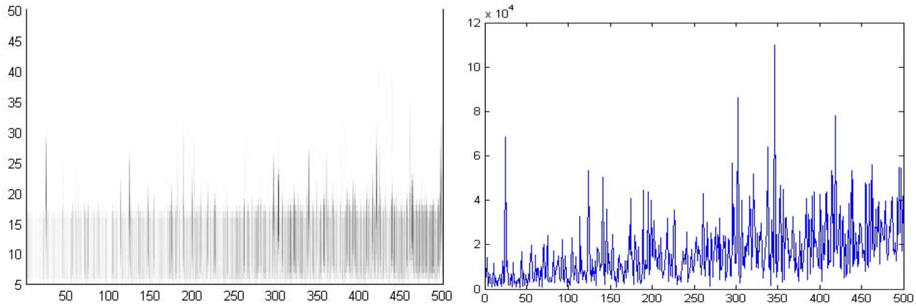
For the method LD+DT, Tabu tenure used per node are chosen randomly into an interval around $f(N) = \sqrt{N}/2$. Spectrograms 2(a) and 3(a) show uniform repartition over the values of the interval. The nodes provoking the highest number of loops (visible in 2(b) and 3(b) curves), appear with a dark color in spectrograms 2(a) and 3(a).

Unlike the first two figures, the spectrograms 4(a) and 5(a) are very different: Tabu tenure values are not uniformly used by the method LD+AT. We expect from this mechanism a better diversification of the search in favor of variables that are not involved in loops. Furthermore, according to the used equation 4, nodes causing most loops (curves 4(b) and 5(b)) use higher values of Tabu tenure. The penalty has increased significantly the values of used Tabu tenure, which corresponds to the expected behavior.



(a) Tabu tenure used (b) Loops per node

Fig. 4. Method LD+AT instance DSJC125.1



(a) Tabu tenure used (b) Loops per node

Fig. 5. Method LD+AT instance DSJC500.1

We have observed the difference of both methods in term of Tabu tenure used by each node during the search. Now, table 2 present comparatives results obtained with these both methods on Leighton instances. Two comparison criteria are used: the success rate s over 10 runs and the average iterations number it.

For all problems studied, we observed that adaptive Tabu tenure is generally better in term of success rate or in term of average iterations number for 7 out of 8 instances. Dynamic Tabu tenure is better only for the instance le450_5d. Furthermore, the method is robust; it finds a solution to all the problems, which demonstrates the effectiveness of the combination of adaptive Tabu tenure and loop detection mechanism.

These results are confirmed by DSJC instances in table 3. The success rate is better for the instance DSJC500.1 and the average number of unsatisfied constraints is lower for the adaptive method (the tests were carried out on five executions and graded from 0 to 10 according to their performance) for 5 out of 6 instances.

Table 2. Adaptation of the Tabu tenure on Leighton instances

		Dynamic Tabu tenure LD+DT		Adaptive Tabu tenure LD+AT	
problems	k	s	it	s	it
le450_5a	5	3	302 000	10	326 148
le450_5b	5	1	629 000	2	1 205 950
le450_5c	5	10	252 120	10	251 881
le450_5d	5	8	396 000	2	1 079 031
le450_15a	15	5	304 000	10	1 889 569
le450_15b	15	9	390 000	10	904 067
le450_15c	15	0	–	10	70.6×10^6
le450_15d	15	0	–	4	192.6×10^6

Table 3. Adaptation of the Tabu tenure on DSJC instances

	Dynamic Tabu tenure LD+DT		Adaptive Tabu tenure LD+AT	
problems	s	c	s	c
DSJC500.1	0	1	6	0.4
DSJC500.5	0	4.8	0	4.2
DSJC500.9	0	3.8	0	3.4
DSJC1000.1	0	15	0	13.4
DSJC1000.5	0	41.2	0	42.8
DSJC1000.9	0	4.4	0	4

5 Comparison with the Literature

In this section, we compare the results obtained by our method before and after the adaptation of the Tabu tenure with four others well-known works published in the literature. First column presents the studied instances and in the second column, we give the chromatic number χ when it is known and the best number of colors used to color each instance, noted k^*. Table 4 presents the minimum number of colors used to solve each instance found by each method.

The first method is named HCA for Hybrid Coloring Algorithm published by Galinier and Hao [11]. This hybrid method combines genetic algorithm and Tabu search. The Tabu tenure is calculated using the mechanism explained in the section 2.3. Notice that the algorithm has obtained the bests known results for several DIMACS instances.

The second method published by Galinier et al. [12], is a population-based method, named AMACOL (Adaptive Memory Algorithm for K-COLoring). This algorithm is also one of the most efficient algorithms in the literature.

The third method is the Iterated Local Search (ILS) algorithm presented by Chiarandini et al. in [5]. A Tabu search algorithm is run until the best solution found does not change during a fixed number of iterations. A perturbation is then applied on the best solution found so far and the Tabu search is run again.

The last algorithm is the Generic Tabu Search (GTS) published by Dorne and Hao [9]. This method uses reactive approach to compute the Tabu tenure (see section 2.3). The search is started from a greedy initial solution built by DSATUR procedure. The Tabu tenure depends on the number of conflicted nodes of the current solution.

Finally, we present our results in the last two columns: first, the basic version (LD+DT) and the adaptive Tabu tenure method (LD+AT) with DT interval equal to $\left[0.5 \times \frac{\sqrt{N}}{2}; 1.5 \times \frac{\sqrt{N}}{2}\right]$ for both methods.

Table 4. Comparison with other methods on DSJC and Leighton instances

problems	(χ, k^*)	HCA	AMACOL	ILS	GTS	LD+DT	LD+AT
DSJC500.1	(-,12)	-	12	13	13	13	**12**
DSJC500.5	(-,48)	48	48	50	50	50	**49**
DSJC500.9	(-,126)	-	126	127	127	128	128
DSJC1000.1	(-,20)	20	20	21	21	21	21
DSJC1000.5	(-,83)	83	84	90	90	89	89
DSJC1000.9	(-,224)	224	224	227	226	230	**227**
le450_15c	(15,15)	15	15	15	-	16	**15**
le450_15d	(15,15)	-	15	15	-	16	**15**
le450_25c	(25,25)	26	26	26	-	26	26
le450_25d	(25,25)	-	26	26	-	26	26

Among the different methods presented here, the method HCA gets the best performance. Our method using adaptive Tabu tenure outperforms our dynamic Tabu tenure method. In particular, adaptive method allows to solve 5 instances (in bold) with less colors than dynamic method on the 10 instances presented in this table. Globally, our adaptive method obtained results of the same quality than HCA on 2 instances and than AMACOL on 5 instances. On random graphs, results obtained by LD+AT are very competitive compared to the others methods. Compared to ILS method, LD+AT obtained better results for 3 instances but it is worst for the instance DSJC500.9. For all others instances, performance are identical in term of minimum number of colors needed.

6 Conclusion and Perspectives

In conclusion, we have seen that the Tabu tenure calculation impacts on the global performance of Tabu list based methods. In the literature, several studies have been undertaken to determine the value of the parameter. We have proposed in this paper a new computation method of the Tabu tenure depending on the specific search history of each variable. The idea is to compute the Tabu tenure according to the number of loops provoked by each variable. We have used the k-coloring problem as a framework for test our method. The presented results show the effectiveness of our generic method comparing to competitive and specialized methods. The adaptive candidate list based Tabu Search was also applied to structured frequencies plan affectation problem [19]. The method has been ranked first among three proposed works (by other teams) in the frame of the ALGOPDF project. In ACL_TS method, only the variables provoking loops become Tabu. The loop detection is determined by a threshold value specifying the number of recent visits (during the M last iterations) after which a variable is considered in loop. The Tabu mechanism being strongly impacted by the loop detection, the study of this second parameter is critical.

References

[1] Bachelet, V., Talbi, E.-G.: COSEARCH: a co-evolutionary metaheuristic. In: Proceedings of the Congress on Evolutionary Computation CEC 2000, pp. 1550–1557 (2000)

[2] Battiti, R., Tecchiolli, G.: The reactive Tabu search. ORSA Journal on Computing 6(2), 126–140 (1994)

[3] Battiti, R.V.J., Rayward-Smith, I.O., Smith, G. (eds.): Reactive Search: Toward Self-Tuning Heuristics, pp. 61–83. John Wiley and Sons Ltd, Chichester (1996)

[4] Blöchliger, I.: Suboptimal colorings and solution of large chromatic scheduling problems, Ph.D. thesis, Mathematics Department, Ecole polytechnique federale de Lausanne, Lausanne, Suisse, 20056

[5] Chiarandini, M., Dumitrescu, I., Stutzle, T.: Stochastic Local Search for the Graph Colouring Problem, Technical Report AIDA-05-03 (2005)

[6] Devarenne, I., Mabed, H., Caminada, A.: Intelligent neighborhood exploration in local search heuristics. In: 18th IEEE International Conference on Tools with Artificial Intelligence, Washington D.C. USA (2006)

[7] Devarenne, I., Mabed, H., Caminada, A.: Self-adaptive Neighborhood Exploration Parameters in Local Search. In: 7th EU/MEeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics, University of Málaga, Spain (2006)

[8] Di Gaspero, L., Schaerf, A.: A Tabu Search Approach to the Traveling Tournament Problem. In: MIC 2005: The 6th Metaheuristics International Conference, Vienna, Austria (2005)

[9] Dorne, R., Hao, J.K.: Tabu Search for graph coloring, T-coloring and Set T-colorings. In: Osman, I.H., et al. (eds.) Metaheuristics 1998: Theory and Applications, ch. 3, Kluver Academic Publishers (1998)

[10] Dupont, A.: Étude d'une métaheuristique hybride pour l'affectation de fréquences dans les réseaux tactiques évolutifs, PhD thesis, Univ. Montpellier II, France (2005)

[11] Galinier, P., Hao, J.-K.: Hybrid Evolutionary Algorithms for Graph Coloring. Journal of Combinatorial Optimization 3, 379–397 (1999)

[12] Galinier, P., Hertz, A., Zufferey, N.: An Adaptive Memory Algorithm for the k-colouring problem. Les Cahiers du GERAD G-2003-35 (2004)

[13] Glover, F.: Tabu Search - Part I. ORSA Journal on Computing 1(3), 190–206 (1989)

[14] Glover, F.: Tabu Search - Part II. ORSA Journal on Computing 2(1), 4–32 (1990)

[15] Hansen, P., Jaumard, B.: Algorithms for the maximum satisfiability problem. Computing 4(44), 279–303 (1990)

[16] Hao, J.-K., Galinier, P.: Tabu search for maximal constraint satisfaction problems. In: Smolka, G. (ed.) CP 1997. LNCS, vol. 1330, pp. 196–208. Springer, Heidelberg (1997)

[17] Hao, J.K., Dorne, R., Galinier, P.: Tabu search for frequency assignment in mobile radio networks. Journal of Heuristics 4(1), 47–62 (1998)

[18] Leighton, F.: A graph coloring algorithm for large scheduling problems. Journal of research of the national bureau of standards, 489–505 (1979)

[19] Mabed, H., Devarenne, I., Caminada, A., Defaix, T.: Frequency Planning for Military Slow Frequency Hopping System. In: International Network Optimization Conference 2007, Spa, Belgium (2007)

[20] Montemanni, R., Smith, D.H.: A Tabu search Algorithm with a dynamic Tabu list for the Frequency Assignment Problem, Technical Report, University of Glamorgan (2001)

[21] Taillard, E.: Robust taboo search for the quadratic assignment problem. Parallel Computing 17, 443–455 (1991)

[22] Neveu, B., Trombettoni, G., Glover, F.: A Candidate List Strategy with a Simple Diversification Device. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 423–437. Springer, Heidelberg (2004)