The Adaptive Tabu Search and Its Application to the Quadratic Assignment Problem

Elena Ikonomovska, Ivan Chorbev, Dejan Gjorgjevik, Dragan Mihajlov

Faculty of Electrical Engineering Skopje,
Karpos II bb, 1000 Skopje, Republic of Macedonia
{ Elena.Ikonomova, ivan, dejan, dragan,}@etf.ukim.edu.mk

ABSTRACT

This article presents a new algorithm for combinatorial optimization based on the basic Tabu Search scheme named Adaptive Tabu Search (A-TS). The A-TS introduces a new, complex function for evaluation of moves. The new evaluation function incorporates both the aspiration criteria and the long-term memory. A-TS also introduces a new decision making mechanism, providing means for avoiding possible infinite loops. The performance of A-TS was measured by applying it to the Quadratic Assignment Problem. The experimental results are compared to published results from other authors. The data shows that A-TS performs favorably against other established techniques.

1 INTRODUCTION

Heuristic search algorithms have proven to be very useful in solving difficult combinatorial optimization problems. Due to their ability to escape local optima, most successful heuristic local search techniques are Simulated Annealing, Genetic Algorithms, and Tabu Search with its variations. Tabu Search has been very successful in achieving near-optimal (and sometimes optimal) solutions to a variety of hard problems.

This paper introduces the Adaptive Tabu Search (A-TS), an improved tabu search algorithm for combinatorial optimization. Adaptive Tabu Search introduces a new evaluation function to the basic scheme of Tabu Search. Our Tabu scheme also proposes a new mechanism for selecting the best move. The selection process uses the evaluation function which incorporates both long-term memory and aspiration criteria.

The performance of our A-TS is evaluated by using instances of the Quadratic Assignment Problem (QAP), chosen from the QAP Library (QAPLIB) [3]. By solving the same problem instances of QAP used by other cited researchers [2][4][5][14], we aimed to derive objective conclusions of the advantages of our Adaptive Tabu Search.

Section 2 presents a formal definition of the QAP. Section 3 provides a brief overview of the basic Tabu Search algorithm and its popular variations. In section 4 we describe the main improvements that we propose to the basic TS algorithm, resulting in our Adaptive Tabu Search (A-TS) algorithm. The environment used to test A-TS is described in section 5. Section 6 presents the experimental results. Our conclusions and areas of further research are given in section 7.

2 THE QUADRATIC ASSIGNMENT PROBLEM

The Quadratic Assignment Problem (QAP) is NP-hard combinatorial optimization problem [13]. Its many practical instances come from areas such as design and resource allocation, microprocessor design and scheduling. Due to the complexity of QAP, in some ways, it has become a benchmark by which new techniques are validated.

For the first time, QAP is stated by Koopmans and Beckman in 1957 [11]. It can be described as follows: Given two $n \times n$ matrices $\mathbf{A}=(a_{ij})$ and $\mathbf{B}=(b_{ij})$, find a permutation $\boldsymbol{\pi^*}$ minimizing

$$\min_{\pi \in \Pi(n)} f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot b_{\pi(i)\pi(j)}$$

where $\Pi(n)$ is the set of permutations of n elements. In other words, it deals with identifying optimal assignments of facilities to locations such that the cost of the resulting system is minimized. Shani and Gonzalez [13] have shown that the problem is *NP*-hard and that there is no *e*-approximation algorithm for the QAP unless $P = NP$.

In practice, a large number of real world problems lead to QAP instances of considerable size, that cannot be solved exactly. For example, an application in image processing requires solving more than 100 QAP problems of size $n = 256$ [15]. Even with today's fastest computers, relatively small problems require prohibitive amounts of time to solve to provable optimality [1]. The use of heuristic methods for solving large QAP instances is currently the only practicable solution.

3 TABU SEARCH OVERVIEW

Glover introduced Tabu Search (TS) in the late 80's [6]. The basic idea behind TS is that, adding short-term memory to local search, improves its ability to locate optimal solutions. Revisiting previously or recently visited solutions is discouraged, and operations that would do so are labeled as being “tabu” or “taboo”. Glover proposed the use of both statically and dynamically sized memory structures for tracking tabu operations. In 1991 Taillard created the Robust Tabu Search (RO-TS) [14], which introduced a dynamic randomly-sized short-term memory design. Battiti and Tecchiolli developed the RE-TS [2] in 1994. They introduced a dynamically sized short-term memory, dependent on the runtime characteristics of the algorithm. Also, they utilized a form of long-term memory that helped prevent searches from stagnating.

Many other TS variations have been developed that incorporate various forms of dynamically-sized short-term memory and long-term memory [9][10]. Still, the RO-TS and RE-TS remain among the most successful and popular. The following concepts are common to most (if not all) Tabu Search techniques, but their specific implementations are somewhat flexible.

A move *m* is an operation by which one solution is transformed into a new, neighboring solution. The neighborhood of the solution, *N(i,k)*, is the set of all solutions that can be derived from the given solution *i*, at iteration *k*, by applying a valid move. For the QAP, a common move strategy consists of swapping facilities assigned to two locations.

The Tabu List implements the short-term memory. It is the most influential piece of any TS design. The basic purpose of the list is to maintain a record of moves that are tabu (discouraged) during a number of following iterations. Usually, a move added to the Tabu List is the reciprocal of the move last accepted and applied to the current solution. The reciprocal is recorded in order to prevent the search from “undoing” recent moves.

During a TS run, it is possible that a single solution will be visited multiple times. To some degree, this behavior is desirable - it supports the concepts of exploitation and exploration. On repeated visits of a solution, the Tabu List will most likely contain a different set of tabu moves, and the search may travel a new path. However, the problem arises when the algorithm continuously revisits the same set of solutions repeatedly (infinite loop), leaving large areas of the search space unexplored. Increasing the length of the list, decreases the probability of entering an infinite loop. On the other hand, longer lists limit the exploration of the search space. The so called long-term memory has a great deal in solving this problem.

When selecting the next move to perform, TS evaluates the neighborhood of the current solution and attempts to find the best non-tabu move; “best” being determined as the objective value of the resulting solution, should the move be applied. Sometimes, however, it may be desirable to allow a tabu move to be chosen. The conditions under which a tabu move would be allowed are known as the aspiration criteria. The most common aspiration criteria is to test whether the implementation of the tabu move would result in the best-fit solution yet found, for the current run. The above criteria is used by Battiti and Tecchiolli in the RE-TS. Figure 1 shows the basic elements of TS.

Step 1. Create an initial solution i at random. Set $i^*=i$ and $k=0$.
Step 2. Set $k=k+1$ and generate a subset V^* of solutions in $N(i,k)$ such that either one of the tabu conditions $tr(i,m)\in \mathrm{Tr}$ is violated ($r=1,...,t$) or at least one of the aspiration conditions $ar(i,m)\in Ar(i,m)$ holds ($r=1,...,a$).
Step 3. Choose a best $j=i\oplus m$ in V^* (with respect to objective function f) and set $i=j$.
Step 4. If $f(i) < f(i^*)$ then set $i^*=i$.
Step 5. Update tabu and aspiration conditions.
Step 6. If a stopping condition is met then stop. Else go to Step 2.

Figure 1: Tabu Search pseudo code.

4 THE ADAPTIVE TABU SEARCH

The Adaptive Tabu Search, that we propose, explores the meaning of finding the “best” move. The search for the best move is a very computation demanding operation. Therefore, it plays a major part in the speed and accuracy of the solving process. The local search in TS consists of evaluating all moves applicable to the current solution, and choosing the best one. In the A-TS approach, the non-tabu move that generates the greatest improvement of the objective function is chosen and applied. In this case, no aspiration criteria are being utilized. However, in some instances, none of the evaluated non-tabu moves provides any improvement. The proposed evaluation function is triggered only when all evaluated moves are tabu or non-improving, non-tabu. The move for which the evaluation function returns the lowest value is accepted and performed.

Any implementation of TS must provide a balance between exploring and exploiting the search space. The risk of visiting certain solutions infinite number of times must be avoided. On the other hand, the potential benefit from revisiting a single solution has to be encouraged. The aim of A-TS is to achieve this balance and maintain it throughout the whole search.

The evaluation function makes its decisions considering the long-term memory and the remaining time (iterations) for the move as tabu (*tabu_time_left*). The long-term memory is implemented as a list of counters, remembering the application of each possible move during the search. In the evaluation function, the number of occurrences of the move (*frequency*) is multiplied with an adaptive coefficient (k_1). The value of k_1 is proportional to the value of the move itself, the frequency of the application of the move and the current iteration. The main objective of the adaptive coefficient is to prevent the search from getting caught in an infinite loop.

On the other hand, the function includes an aspiration criterion. It allows a tabu move to be performed, if it seems promising and not risky in terms of loops or local stagnation. The criterion is implemented using another adaptive coefficient (k_2), whose value also changes and is proportional to the value of the move. The adaptive nature of our Tabu Search scheme is based on these two adaptive coefficients. The final form of the evaluation function is:

evaluation_ func (*move_value, frequency, tabu_time_left*)=
move_value + k_1 * *frequency* + k_2 * *tabu_time_left*

where:

$$k_1 = \begin{cases} c_1 \cdot iter \cdot abs(\max(move\_value, 1)), \text{ if } freq > avgfreq \\ c_2 \cdot abs(move\_value), \text{ if } freq \le avgfreq \end{cases}$$

$$k_2 = \begin{cases} 0, \text{ if } move\ is\ tabu \\ c_3 \cdot abs(move\_value), \text{ if } move\ is\ not\ tabu \end{cases}$$

The coefficients k_1 and k_2 control the influence of the move frequency and the remaining time of the move in the tabu list. The coefficients c_1, c_2 and c_3 are tuned up experimentally, according to the specific problem being solved. Their influence upon the accuracy of the obtained solutions is considerable. However, they have no significant influence on the number of iteration required to reach the optimal solution. The values for c_1, c_2 and c_3 used here are 10, 0.01 and 0.01 respectively.

5 BENCHMARK INSTANCES

The problem instances used in the development and testing of A-TS were obtained from the QAPLIB, a public library of QAP problems and their best-known solutions [3]. The number in the problem's name corresponds to the size of the problem. QAPLIB currently contains over 100 instances that have been used in earlier researches. Some of them originate from real life applications, like hospital layout (kra30*, els19), typewriter design (bur26*), etc.

As shown by Taillard [15], the quality of solutions produced by heuristic methods strongly depends on the problem type. For problems taken from the real world, many heuristic methods perform rather poorly. They are not able to find solutions within 10% of the value of the best solutions known, even if excessive computing time is allowed. Conversely, the same methods may perform very well on randomly generated problems. For such problems, almost all heuristic methods are able to find high quality solutions (i.e., solutions approximately one percent worse than the best solution known). Therefore, it is reasonable to analyze the performance of A-TS by splitting the problem instances into two categories: (i) real world, irregular and structured problems, and (ii) randomly generated, regular and unstructured problems.

6 EXPERIMENTAL RESULTS

A-TS is compared with a set of the best heuristic methods available for the QAP, such as the genetic hybrid method of Fleurent and Ferland [5] (GH), the reactive tabu search of Battiti and Tecchiolli [2] (RE-TS), the tabu search of Taillard [14] (RO-TS) and a simulated annealing from Connolly [4] (SA). In the comparison, a large subset of well known problem instances is considered, with sizes between n = 12 and n = 35, contained in the QAPLIB.

The complexity of one iteration, for each algorithm considered, varies: SA has the lower complexity with $O(n)$ per iteration. RO-TS and RE-TS have a complexity of $O(n^2)$ per iteration, GH has a complexity of $O(n^3)$, while A-TS has a complexity of $O(n(n-1)/2) \approx O(n^2)$.

In order to make fair comparisons between these algorithms, the same computational time was given to each test problem trial, by performing a number of iterations equal to $nI_{max}(62.5n - 5)$ for A-TS, to $10nI_{max}$ [12] for RE-TS and RO-TS, $125n^2I_{max}$ [12] for SA and $2.5I_{max}$ [12] for GH.

Tests are performed with I_{max}=10. The experiments evaluate their ability in producing relatively good solutions under strong time constraints.

Problem name	Best known value	RO-TS	RE-TS	SA	GH	A-TS
Els19	17212548	21.261	6.714	16.028	**0.515**	10.0914
Tai20b	122455319	**0**	—	6.7298	**0**	1.4522
Tai25b	344355646	0.0072	—	1.1215	**0**	0.0559
Tai30b	637117113	0.0547	—	4.4075	**0.0003**	1.7026
Tai35b	283315445	0.1777	—	3.1746	**0.1067**	1.1849
Kra30a	88900	0.4702	2.0079	1.4657	0.1338	**0.0267**
Kra30b	91420	0.0591	0.7121	0.1947	0.0536	**0**
Chr25a	3796	6.9652	9.8894	12.4973	2.6923	**0**

Table 1: *Quality of various heuristic methods for irregular problems, measured in percent above the best solution value known. Best results are in boldface.*

Table 1 compares all mentioned methods on real life, irregular and structured problems opposed to A-TS. In particular, the average quality of the solutions produced by these methods is shown, measured in percent above the best solution value known. The RE-TS, S-TS, and RO-TS data contained in table 1 and 2, was gathered from L. M. Gambardella, É. D. Taillard and M. Dorigo [12]. The results of the mentioned authors are averaged over 10 runs, while the results of A-TS are averaged over 100 runs.

Table 1 shows that methods like RE-TS or SA are not well adapted for irregular problems. Sometimes, they produce solutions over 10% worse than the best solutions known. Other heuristic methods are able to exhibit solutions at less than 1% of the optimum value, with the same computing effort. For problem types tai*b, GH seems to be the best method overall. Our approach produces solutions with average deviation smaller than 1% in most of the cases.

Table 2 provides the same type of comparisons as those of table 1, only for unstructured problems. Table 2 shows that our technique outperforms all of the other techniques, for all of the listed problems. In half of the cases, our results achieve the exact best solutions in all 100 trials, whereas in the rest, the average gap (deviation from the optimal) is below 1%.

Problem name	Best known value	RO-TS	RE-TS	SA	GH	A-TS
Nug20	2570	0.101	0.911	0.327	0.047	**0**
Nug30	6124	0.271	0.872	0.500	0.249	**0**
Tai20a	703482	0.769	0.705	1.209	0.732	**0.046**
Tai25a	1167256	1.128	0.892	1.766	1.371	**0.736**
Tai30a	1818146	0.871	1.044	1.434	1.160	**0**
Tai35a	2422002	1.356	1.192	1.886	1.455	**0.014**

Table 2: *Quality of various heuristic methods for regular problems measured in percent above the best solution value known. Best results are in boldface.*

Additional comparison of the algorithms, based on the number of iterations needed to achieve the optimal solution, was performed. A series of runs performed with the A-TS were compared with published results of the metaheuristic search techniques RE-TS and RO-TS. Table 3 shows comparisons over some of the problems from the Taillard set, ranging in sizes from 12 to 35. 100 runs were performed on each problem by A-TS, opposed to 30 runs performed by the authors of the other approaches. The RE-TS, and RO-TS data contained in this table was gathered from Battiti and Tecchiolli [2]. The best result in each row is bolded.

Problem	Max.Iter A-TS/Others	RE-TS	RO-TS	A-TS Avg. Iter.
Tai12a	10K/100K	282.3	210.7	**165.7**
Tai15a	10K/100K	**1780.3**	2168.0	2145.5
Tai17a	100K/100K	**4133.9**	5020.4	4363.9
Tai20a	100K/500K	37593.2	34279	**31650.8**
Tai25a	400K/1M	38989.7	80280.4	**19945**
Tai30a	560K/2M	**68178.2**	146315.7	104084.2
Tai35a	760K/4M	**281334.0**	448514.5	290458.1

Table 3: *Comparison of average iterations before convergence to best solution for RE-TS, RO-TS, and A-TS.*

7 CONCLUSION

This paper describes a novel approach to the Tabu search scheme. We propose a new decision making mechanism with a new evaluation function to integrate within the standard TS. The resulting Adaptive Tabu Search (A-TS) augments the exploration and exploitation of the search space, through the incorporation of long-term memory, aspiration criteria and the value of the move in a single evaluation function. By using search history, the adaptive coeficients within the combined evaluation function provide useful feedback to the process.

Instances of the Quadratic Assignment problem were used for quantitative evaluation of the algorithm. Experimental results show that A-TS performs favorably. In some cases, the optimal result was found in less iteration than other techniques. For most of the problems, especially regular problem instances, A-TS seems to be the best choice.

Based on the encouraging results, further research of A-TS will be performed. Its implementation to more complex, real life problems, will provide more details of the algorithm quality and advantages.

References

[1] Anstreicher, K.M. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming, Series B* 97:27-42, 2003.

[2] Battiti, R. and Tecchiolli, G. The reactive tabu search. *ORSA Journal on Computing, 6, 2,* 126-140, 1994.

[3] R. E.Burkard, S. E. Karisch and F. Rendl. QAPLIB—a quadratic assignment problem library. *Journal of Global Optimization*, 10, 391–403, 1997.

[4] Connolly D T (1990). An improved annealing scheme for the QAP. *Eur J Op Res* 46: 93–100.

[5] Fleurent C and Ferland J (1994). Genetic hybrids for the quadratic assignment problem. *DIMACS Series in Mathematics and Theoretical Computer Science* 16: 190–206.

[6] Glover, F. Future Paths for Integer Programming and Link to Artificial Intelligence. *Computers and Operations Research*, 12:533-549, 1986.

[7] Glover, F., Laguna, M. *Tabu Search*. Kluwer Academic Publishers, 1997.

[8] Glover, F. Laguna, M. Tabu Search. *Modern Heuristic Techniques for Combinatorial Problems*. Colin Reeves, ed. Blackwell Scientific Publishing, 71-140, 1993.

[9] Koopmans T C and Beckmann M J (1957). Assignment problems and the location of economics activities. *Econometrica* **25**: 53–76.

[10] L. M. Gambardella, É. D. Taillard and M. Dorigo. Ant colonies fot the quadratic assignment problem.IDSIA-1997.

[11] Sahni, S. and Gonzalez, T. P-complete approximation problems. *J. ACM 23*, 555-565, 1976.

[12] Taillard, E. D. Robust tabu search for the quadratic assingnment problem. *Parallel Computing*, 17:443-455, 1991.

[13] Taillard, E. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3:87-103, 1995.