Осма национална конференција со меѓународно учество - ЕТАИ 2007
Eighth National Conference with International Participation - ETAI 2007
Охрид, Република Македонија - Ohrid, Republic of Macedonia, 19 - 21. IX 2007

A DATA MINING APPROACH TO PERFORMANCE REFINEMENT OF A HEURISTIC ALGORITHM

Elena Ikonomovska, Dejan Gjorgjevik

University St. Cyril & Methodius, Faculty of Electrical Engineering and Information Technologies, Skopje, Republic of Macedonia, elenai@feit.ukim.edu.mk, dejan@feit.ukim.edu.mk

***Abstract*-- This paper introduces a novel approach to performance refinement of a heuristic algorithm for combinatorial optimization. The proposed Adaptive Tabu Search (A–TS) algorithm introduces adaptive behavior in the traditional Tabu Search algorithm. The adaptive nature of this algorithm is based on two adaptive coefficients that drive the heuristic. Choosing appropriate values for these parameters has great impact on A–TS performance and accuracy. This article presents a novel approach towards improving the performance of A-TS utilizing data mining techniques for tuning the adaptive coefficients. The performance of A–TS was measured by applying it to the Quadratic Assignment Problem. Published results from other authors were used for comparison.**

***Index terms*—data mining, heuristic, quadratic assignment problem**

1. INTRODUCTION

For many practical, real-life problems, heuristic algorithms seem to be the only way to get good solutions in a reasonable amount of time. Among the most successful heuristic techniques are Simulated Annealing, Genetic Algorithms, and Tabu Search with its variations. Tabu Search algorithms have proven to be very successful in achieving near-optimal and sometimes optimal solutions to a variety of difficult combinatorial optimization problems. Like other heuristic local search techniques, the performance of any Tabu Search algorithm strongly depends on the values of several parameters.

Data mining techniques are used increasingly often for analyzing, and understanding large amounts of data. In this paper we employ a data mining technique for performance refinement of a heuristic algorithm for combinatorial optimization, which presents an improved variation of the Tabu Search algorithm, named Adaptive Tabu Search (A-TS) [4]. Adaptive Tabu Search introduces a new evaluation function to the basic scheme of Tabu Search. The evaluation function is used in the selection process and involves several parameters and coefficients. Choosing appropriate values for these parameters has great impact on A-TS performance. To achieve better performance, coefficient tuning was performed, by applying data mining techniques.

The improved performance of our A-TS was evaluated by using instances of the Quadratic Assignment Problem (QAP), chosen from the QAP Library (QAPLIB) [1]. By solving the same problem instances of QAP used by other cited researchers [2][3][5][6], we aimed to derive objective conclusions about the advantages of our Adaptive Tabu Search and the use of data mining techniques for optimizing the performance of meta-heuristic algorithms.

Section II presents a formal definition of the QAP. Section III provides a brief overview of the basic Tabu Search algorithm and its popular variations. In section IV, we describe the main improvements that we proposed to the basic TS algorithm, resulting in our Adaptive Tabu Search (A-TS) algorithm. The implementation issues of a data mining technique used for coefficients tuning are addressed in section V. The environment used to test A-TS and the experimental setup is described in section VI. Section VII presents the experimental results. Our conclusions and areas of further research are given in section VIII.

2. THE QUADRATIC ASSIGNMENT PROBLEM

The Quadratic Assignment Problem (QAP) is an NP-hard combinatorial optimization problem [7]. Its many practical instances come from areas such as design and resource allocation, microprocessor design and scheduling. Koopmans and Beckman formulated QAP for the first time in 1957 [8]. It can be described as follows: Given two n × n matrices A=(a_{ij}) and B=(b_{ij}), find a permutation π* 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 Π(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. In practice, a large number of real world problems lead to QAP instances of considerable size that cannot be solved exactly. The use of heuristic methods for solving large QAP instances is currently the only practical solution [10].

3. TABU SEARCH OVERVIEW

Tabu Search (TS) was introduced by Glover in the late 80's [11][12]. The basic idea behind TS is that adding short-term memory to local search improves its ability to locate optimal solutions. By maintaining a list of "tabu" moves, revisiting previously or recently visited solutions is discouraged. Glover proposed the use of both statically and dynamically sized memory structures for tracking tabu operations. Some of the most famous variations of Tabu Search are the Robust Tabu Search (RO-TS) by Taillard [6] and the Reactive Tabu Search (RE-TS) by Battiti and Tecchiolli [2].

Many other TS variations have been developed that incorporate various forms of dynamically sized short-term memory and long-term memory [13][14]. 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 main weakness of TS is its tendency to explore a too limited region of the search space, i.e., the search lacks breadth, unless systematic and effective diversification schemes are used [15]. 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. By increasing the length of the list, the probability of entering an infinite loop decreases. On the other hand, longer lists limit the exploration of the search space. The so-called long-term memory has a great deal of influence in solving this problem.

4. THE ADAPTIVE TABU SEARCH

The search for the best move is a very computationally 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 in the neighborhood applicable to the current solution, and choosing the best one. In our approach, we aimed to find the best way for evaluating the moves and determining which is the best move in the current iteration. 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. Our 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 (for this particular problem) is accepted and performed.

Rather than taking into account only the value of a move, the evaluation function makes its decisions considering both the long-term memory and the remaining time for the move as tabu (*tabu*). 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 (*freq*) is multiplied with an adaptive coefficient (k_1). The value of k_1 is proportional to the value of the move itself (*value*), the frequency of the application of the move and the current iteration. The main objective of this adaptive coefficient is to prevent the search from being caught in an infinite loop. This is done by discouraging moves that have been frequently applied. This implements a successful diversification strategy, as will be shown in section 7.

The evaluation function also 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. This criterion is implemented using another adaptive coefficient (k_2), whose value also changes and is proportional to the value of the move. This is because a tabu move with a value much greater than the rest of the non improving, non tabu or tabu moves in a current iteration will be the best move according to the evaluation function even if it has been applied very recently. When applied the adaptive coefficient k_2, a move will be discouraged according to the value of the move.

The adaptive nature of our Tabu Search scheme is based on these two adaptive coefficients. Their values change with every iteration. The final form of the evaluation function is:

$$eval(value, freq, tabu) = value + k_1 \cdot freq + k_2 \cdot tabu \qquad (1)$$

where:

$$k_1 = \begin{cases} c_1 \cdot i \cdot \max(value, 1), & \text{if } freq > avgfreq \\ c_2 \cdot abs(value), & \text{if } freq \le avgfreq \end{cases} \qquad (2)$$

$$k_2 = \begin{cases} 0, & \text{if } move\ is\ not\ tabu \\ c_2 \cdot abs(value), & \text{if } move\ is\ tabu \end{cases} \qquad (3)$$

and i is the current iteration.

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. These coefficients drive the heuristic. Therefore, their influence upon the accuracy of the obtained solutions is significant. Since their values also depend on the values of coefficients c_1 and c_2, scientific method should be applied for fine-tuning the coefficients c_1 and c_2.

5. THE APRIORI ALGORITHM

The performance of tabu search algorithms depends of many search parameters such as tabu tenures, move selection probabilities, coefficients etc. Significant performance benefits can be achieved by determining appropriate values for these search parameters. In this paper, we present the use of a data mining method for optimizing the coefficients c_1 and c_2 incorporated in the evaluation function, in order to improve the performance of A-TS.

When improving the performance of a certain algorithm the goal is to increase the accuracy and minimize the time needed for producing satisfactory solutions. The coefficients c_1 and c_2 are part of the evaluation function that plays a major role in the speed and accuracy of the search. Determining the relationships between the coefficients c_1 and c_2 and the number of iterations needed for producing optimal or near optimal solutions, would provide valuable information for improving the performance of the algorithm. This kind of relationships or associations can be discovered by the use of data mining algorithms for mining association rules in large data sets. The process is known as association rule mining.

The performance of A-TS is analysed by the use of a data mining technique for mining association rules. Before describing the algorithm for association rule mining we will give some definitions first.

An *item* is a triple that represents either a categorical attribute with its value, or a quantitative attribute with its range. The value of a quantitative attribute can be represented as a range where the upper and lower limits are the same. We use the term *itemset* to represent a set of items. A *k-itemset* is an itemset that contains k items. An itemset is *frequent* if it satisfies a minimum support (*min_sup*) threshold, which can be set by the user or a domain expert. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D, the task relevant data, be a set of transactions where each transaction T is a set of items such that $T \subseteq I$. An itemset satisfies the minimum support constraint if the occurrence frequency of the itemset is greater than or equal to the product of min_sup and the total number of transactions in D.

An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \emptyset$. The rule $A \Rightarrow B$ holds in the transaction set D with support s, where s is the percentage of transactions in D that contain $A \cup B$. The rule $A \Rightarrow B$ has confidence c in the transaction set D if c is the percentage of transactions in D containing A which also contain B. That is,

$$Support\ (A \Rightarrow B) = prob\ \{A \cup B\} \qquad (4)$$

$$Confidence\ \ (A \Rightarrow B) = prob\ \{B \mid A\} \qquad (5)$$

Rules that satisfy both a minimum support threshold (*min_sup*) and a minimum confidence threshold (*min_conf*) are called strong.

The process of inducing association rules consists of two steps:

Step1: *Finding all frequent itemsets.*
Step2: *Generating strong association rules from the frequent itemsets.*

Among the best algorithms for mining boolean association rules in large sets of data is the Apriori algorithm [16]. The algorithm uses prior knowledge of frequent itemset properties to prune (reduce) the search space. The Apriori algorithm uses an important property called the *Apriori property* that states: *All subsets of a frequent itemset must also be frequent*. Once the frequent itemsets from the transactions in D have been found, it is straightforward to generate strong association rules from them.

6. EXPERIMENTAL SETUP

For this specific problem of optimizing the performance of the A-TS, rules will only be interesting if they represent non-trivial correlations between the coefficients c_1 and c_2 as antecedents and the number of iterations and the difference between the produced and the optimal solution as consequents.

The transaction set D was generated by performing 10000 runs of the algorithm A-TS. Ten different values for each of the coefficients c_1 and c_2 were used and one hundred different initial solutions as a starting point for the search. The values for coefficient c_1 vary in the interval [10, 100]. This is an arithmetic progression where the first term is 10 and the common difference is 10. The values for coefficient c_2 are in the interval [1000^{-1}, 100^{-1}] and change with arithmetic progression where the first term is 1000 and the common difference is -100. One hundred different seed values for generating the initial solution were used. The table obtained consists of five quantitative attributes (the number of iterations named *num_iterations*, the difference between the produced and the optimal solution named *gap*, *seed*, c_1 and c_2).

To avoid the potential information loss, the attributes c_1 and c_2 are not partitioned into intervals. Instead, they are directly mapped to consecutive intervals, such that the order of the intervals is preserved. Partitioning was done over the attributes *num_iterations* and *gap*, where the number of intervals was kept small. Best results were achieved when implementing the equi-width strategy due to the type and structure of the data.

As it was previously stated, the Apriori algorithm is well known algorithm for mining frequent itemsets for Boolean association rules. A boolean association rule concerns associations between the presence or absence of items and not associations between quantitative items or attributes. Since our data set consists of quantitative attributes, we refer to this mining problem as Quantitative association rules problem.

The Boolean association rules problem can be considered as a special case of Quantitative association rules problem. Therefore, mapping the Quantitative association rules problem into the Boolean association rules problem will allow us to use any algorithm for finding Boolean association rules including the Apriori algorithm in order to find Quantitative association rules [17].

After partitioning the attributes, the Apriori algorithm was employed to find the frequent itemsets and generate the association rules. The value of the parameter min_sup was set to 0.01.

For determining the interesting strong rules from the generated set of association rules, additional measure of interestingness should be applied [19]. The most popular objective measure of interestingness is *lift*. Lift is defined as the ratio of the frequency of the consequent (*B*) in the transactions that contain the antecedent (*A*) over the frequency of the consequent in the data as a whole.

$$Lift(A \Rightarrow B) = Confidence(A \Rightarrow B) / Support(B) \quad (6)$$

Lift values greater than 1 indicate that the consequent is more frequent in transactions containing the antecedent than in transactions that do not. It indicates the influence of the antecedent on the frequency of the consequent.

Using this measure, we define an interesting rule as a rule that satisfies the user-specified minimum confidence threshold and has *lift* greater then one. The minimum confidence threshold in this process was set to 0.5.

7. EXPERIMENTAL RESULTS

The problem instances used in the development and testing of A-TS are obtained from QAPLIB, a public library of QAP problems and their best-known solutions [1]. The number in the problem's name corresponds to the size of the problem. Most of these problems come from practical applications or they are randomly generated with non-uniform laws that imitate the distributions observed on real world problems.

The process of coefficients tuning was performed for a subset of QAP problem instances of sizes between n=20 and n=35. For each of them, association rule mining was performed. The interesting rules were extracted and the most promising values for the coefficients c_1 and c_2 were set manually. To compare the results obtained before and after the coefficients tuning one hundred trials were performed for each problem. Trials were performed only for short runs. The experiments evaluate the improvement of the quality of produced solutions under strong time constraints.

TABLE I.
QUALITY OF A-TS FOR REGULAR AND IRREGULAR QAP PROBLEMS MEASURED IN PERCENT ABOVE THE BEST SOLUTION VALUE KNOWN.

Problem name	Best known value	Before tuning	After tuning	Diff. in iter.	Improvement	In %
Tai20b	122455319	13.983	13.886	285	0.097	0.69
Tai25b	344355646	3.909	2.589	-1629	1.320	33.768
Tai30b	637117113	4.119	3.707	-468	0.412	10.002
Tai35b	283315445	2.635	2.635	0	0	0
Kra30a	88900	0.583	0.495	-665	0.088	15.094
Kra30b	91420	0.002	0.002	0	0	0
Chr25a	3796	1.452	1.452	0	0	0
Nug20	2570	0	0	0	0	0
Nug30	6124	0.020	0.020	0	0	0
Tai20a	703482	0.250	0.250	0	0	0
Tai25a	1167256	0.814	0.814	0	0	0
Tai30a	1818146	0.371	0.327	-523	0.044	11.860
Tai35a	2422002	0.618	0.579	-2288	0.039	6.311

Table I provides a comparison of the quality of solutions before and after tuning. The last column represents the improvement as a difference between the solution quality before and after tuning. Before coefficients tuning, trials were performed for c_1=10 and c_2=100^{-1}. The initial values for coefficients c_1 and c_2 were obtained from the previous experience with A-TS [4]. Therefore, for half of the problem instances the improvement is zero because the optimal coefficients values were the same before and after. For the other half of the cases, the average value of the improvement is 12.954 percent. This shows that our algorithm is robust and obtains high quality solutions for a considerable number of problem instances, without additional coefficients tuning for the different problem instances. However, the improvement achieved for some of the QAP problems is significant and justifies this research.

The column that represents the difference in iterations shows that in almost all of the cases the improvement is achieved when the number of iterations is increased. This means that, the quality of the algorithm is a compromise between the speed and the accuracy. When coefficient tuning was applied to improve the speed of the algorithm, the number of iterations was decreased, but this resulted in a lower accuracy of the algorithm.

Tuned 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 [6] (RO-TS) and a simulated annealing from Connolly [3] (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 varies for the compared algorithms: SA has the lowest 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($n^2/2$) per iteration. Tests are performed on short runs. The reason to compare algorithms on short runs is to evaluate their ability in producing relatively good solutions under strong time constraints.

As shown by Taillard [9], the quality of solutions produced by heuristic methods strongly depends on the problem type, that is, the structure of the data matrices ***A*** and ***B***. For problems taken from the real world, many heuristic methods perform rather poorly. 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.

TABLE II.
QUALITY OF VARIOUS HEURISTIC METHODS FOR IRREGULAR PROBLEMS AND SHORT RUNS MEASURED IN PERCENT ABOVE THE BEST SOLUTION VALUE KNOWN. BEST RESULTS ARE IN BOLDFACE

Problem name	Best known value	RO-TS	RE-TS	SA	GH	A-TS
Tai20b	122455319	6.700	—	14.392	**0.150**	13.886
Tai25b	344355646	11.486	—	8.831	**0.874**	2.589
Tai30b	637117113	13.284	—	13.515	**0.952**	3.707
Tai35b	283315445	10.165	—	6.935	**1.084**	2.635
Kra30a	88900	2.666	2.155	1.813	1.576	**0.495**
Kra30b	91420	0.478	1.061	1.065	0.451	**0.002**
Chr25a	3796	15.969	16.844	27.139	15.158	**1.452**

Table II compares A-TS with the above-mentioned methods on real life, irregular and structured problems. In particular, the average quality of the solutions produced by these methods is shown, measured in percent above the best solution value known. The RO-TS, RE-TS, SA, and GH data contained in table II, III, and IV was gathered from L. M. Gambardella, É. D. Taillard and M. Dorigo [18]. The results of the mentioned authors are averaged over 10 runs, while the results of A-TS are averaged over 100 runs. The experiments evaluate their ability to produce relatively good solutions under strong time constraints.

Table II 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. For problem types *Tai..b*, GH seems to be the best method overall. For problem instances that originate from real life applications (*Kra30a* and *Kra30b*) A-TS performs best. Our approach produces solutions with average deviation smaller than 3% in most of the cases.

TABLE III.
QUALITY OF VARIOUS HEURISTIC METHODS FOR REGULAR PROBLEMS AND SHORT RUNS MEASURED IN PERCENT ABOVE THE BEST SOLUTION VALUE KNOWN. BEST RESULTS ARE IN BOLDFACE.

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	**0.020**
Tai20a	703482	0.769	0.705	1.209	0.732	**0.250**
Tai25a	1167256	1.128	0.892	1.766	1.371	**0.814**
Tai30a	1818146	0.871	1.044	1.434	1.160	**0.327**
Tai35a	2422002	1.356	1.192	1.886	1.455	**0.579**

Table III provides the same type of comparisons as those of table II, for regular, unstructured problems. Table III shows that for all of the listed problems, our technique outperforms the other methods. For all of the problem instances the average gap (deviation from the optimal) of the produced solutions is below 1%.

An 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 the published results of the metaheuristic search techniques RE-TS and RO-TS.

Table IV shows a comparison over some of the problems from the Taillard set, ranging in sizes from 12 to 35. A hundred runs were performed on each problem by A-TS, as opposed to 30 runs performed by the authors of the other approaches. The RE-TS and RO-TS data contained in this table were gathered from Battiti and Tecchiolli [2]. The best result in each row is shown in bold.

TABLE IV.
COMPARISON OF AVERAGE ITERATIONS BEFORE CONVERGENCE TO BEST SOLUTION FOR RE-TS, RO-TS, AND A-TS.

Problem name	Best known value	Average iterations to best solution		
		RE-TS	RO-TS	A-TS
Tai12a	224416	282.3	210.7	**165.7**
Tai15a	388214	**1780.3**	2168.0	2145.5
Tai17a	491812	4133.9	5020.4	**4094.25**
Tai20a	703482	37593.2	34279	**31650.8**
Tai25a	1167256	38989.7	80280.4	**16619.08**
Tai30a	1818146	**68178.2**	146315.7	81038.33

Tai35a	2422002	281334.0	448514.5	**240595.1**

The experimental results presented in Table IV show that our algorithm performs competitively against other techniques. Although the accent was placed on the accuracy of the algorithm rather than the speed, the results are very satisfactory. In nearly all of the cases, A-TS converges in a smaller number of iterations than the other variations of Tabu Search.

8. CONCLUSION

This paper describes a novel approach to performance refinement of our Adaptive Tabu Search algorithm. Using search history, the adaptive coefficients within the combined evaluation function of A-TS play an important role in the search, providing useful feedback to the process. The adaptive coefficients were tuned for achieving best performance. We propose the use of a data mining technique for discovering interesting correlations between the coefficients being tuned and the quality of the produced solutions.

Instances of the Quadratic Assignment problem were used for coefficients tuning and quantitative evaluation of the algorithm under strong time constraints. The experimental results showed that tuned A-TS performs favorably. In some cases, the optimal result was found in fewer iterations as compared to other techniques. For most of the problems, especially for regular problem instances and real life problem instances, A-TS seems to be the best choice.

Although the aim of using a data mining technique was to improve this particular tabu search algorithm, the implications are quite general. The same ideas can easily be adapted and applied to other heuristic algorithms as well. The utilization of an established data mining technique for coefficients tuning significantly improves the hand-tuned algorithm. Based on the encouraging results, further research in the area of introducing data mining techniques for performance refinement of heuristic algorithms will be performed. One of the directions of our further research is predictive model for real-time coefficients tuning independent of the problem type.

9. REFERENCES

[1] R. E. Burkard, S. E. Karisch and F. Rendl. QAPLIB—A Quadratic Assignment Problem Library. Journal of Global Optimization, 10, 391–403, 1997.

[2] R. Battiti and G. Tecchiolli, The Reactive Tabu Search. ORSA Journal on Computing, 6, 2, 126-140, 1994.

[3] D. T. Connolly (1990). An Improved Annealing Scheme for the QAP. Eur. J. Op. Res. 46: 93–100.

[4] E. Ikonomovska, I. Chorbev, D. Gjorgjevik, D. Mihajlov. The Adaptive Tabu Search and Its Application to the Quadratic Assignment Problem. Information Society 2006, Ljubljana, Slovenia.

[5] C. Fleurent and J. Ferland (1994). Genetic Hybrids for the Quadratic Assignment Problem. DIMACS Series in Mathematics and Theoretical Computer Science 16: 190– 206.

[6] E. D. Taillard, Robust Tabu Search for the Quadratic Assignment Problem. Parallel Computing, 17:443-455, 1991.

[7] S. Sahni and T. Gonzalez, P-complete Approximation Problems. J. ACM 23, 555-565, 1976.

[8] T. C. Koopmans and M. J. Beckmann (1957). Assignment Problems And The Location Of Economics Activities. Econometrica 25: 53–76.

[9] E. Taillard, Comparison Of Iterative Searches for the Quadratic Assignment Problem. Location Science, 3:87-103, 1995.

[10] K. M. Anstreicher, Recent Advances in the Solution of Quadratic Assignment Problems. Mathematical Programming, Series B 97:27-42, 2003.

[11] F. Glover, Tabu Search – Part I. *ORSA Journal on Computing*, 1(3): 109-206, 1989.

[12] F. Glover, Tabu Search – Part II. *ORSA Journal on Computing*, 2:4-32, 1990.

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

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

[15] T. G. Crainic, M. Gendreau, J. Potvin. Parallel Tabu Search. 2005, Montreal, Canada

[16] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. Proc. 20th Int. Conf. Very Large Data Bases, VLDB 1994.

[17] R. Srikant and R. Agrawal. Mining Quantitative Association Rules in Large Relational Tables. SIGMOD-1996.

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

[19] G. I. Webb. Discovering Significant Rules. KDD, August 20-23, 2006, Philadelphia, Pennsylvania, USA.