EXPERIMENTS WITH TABU SEARCH FOR RANDOM QUADRATIC ASSIGNMENT PROBLEMS

Alfonsas Misevičius, Jonas Blonskis

Kaunas University of Technology, Department of Practical Informatics
Studentų St. 50–416a, LT–51368 Kaunas, Lithuania

Abstract. Tabu search (TS) is a modern highly effective meta-heuristic for solving various optimization problems. In this paper, we discuss some enhancements of TS for one of the difficult combinatorial optimization problems – the quadratic assignment problem (QAP). We implemented five variants (modifications) of TS for the random QAP instances from the library of the QAP instances QAPLIB. These random QAPs pose a real challenge for the researchers. A number of the experiments were carried out on these instances. The results obtained from the experiments demonstrate the outstanding efficiency of the modifications proposed. These modifications seem to be superior to the earlier TS algorithms for the QAP. In addition, the new best known solution has been achieved for the instance tai100a.

Keywords: combinatorial optimization, quadratic assignment problem, heuristics, meta-heuristics, tabu search, iterated tabu search.

Indroduction

The quadratic assignment problem (QAP) can be formulated as follows. Let two matrices $\boldsymbol{A} = (a_{ij})_{n\times n}$ and $\boldsymbol{B} = (b_{kl})_{n\times n}$ and the set Π of all possible permutations of $\{1, 2, ..., n\}$ be given. The goal is to find a permutation $\pi = (\pi(1), \pi(2), ..., \pi(n)) \in \Pi$ that minimizes

$$z(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)}. \qquad (1)$$

One of the interpretations of the QAP is the facility layout problem [12]. In this case, n is the number of facilities, the entry a_{ij} can be seen as the flow of materials from facility i to facility j, and b_{kl} denotes the distance between location k and location l. The permutation $\pi = (\pi(1), \pi(2), ..., \pi(n))$ represents an assignment of n facilities to n locations.

It has been proved that the QAP is NP-hard [19]. Various heuristic approaches have been applied for solving this problem. For a survey of heuristics for the QAP, see, for example, [2, 4, 17].

Tabu search (TS) meta-heuristic was introduced by Hansen and Jaumard, 1987 [10], and Glover, 1989, 1990 [7,8]. Since that time, it has been proven to be a powerful technique for difficult optimization problems, among them, the QAP (see, for example, [1, 14, 18, 20, 22, 23]). Before describing briefly the basic idea of TS, let us introduce some definitions related to combinatorial optimization (CO). A CO problem P can be defined as a pair (S, f), where $S = \{s_1, s_2, ...\}$ is a finite (or countable infinite) set of feasible solutions (a "solution space") and f: $S \to R^1$ is a real-valued objective (cost) function. Without loss of generality, we assume that f seeks a global minimum. Thus, to solve the CO problem one has to search for a solution $s_{\text{opt}} \in S$ such that

$$s_{\text{opt}} \in S_{\text{opt}} = \left\{ s^{\nabla} \mid s^{\nabla} = \arg\min_{s\in S} f(s) \right\}. \qquad (2)$$

The solution s_{opt} is called a globally optimal solution (global optimum) of (S, f). A neighbourhood function N: $S \to 2^S$ assigns for each $s \in S$ a set $N(s) \subseteq S$ – the set of neighbouring solutions (neighbours) of s (or simply the neighbourhood of s). (The 2-exchange function N_2 is a commonly used neighbourhood structure for the case where the solutions are permutations (like in the QAP). In this case, $N_2(s) = \{ s' \mid s' \in S, \rho(s, s') \le 2 \}$, where $s \in S$, and $\rho(s, s')$ is a distance between permutations s and s': $\rho(s,s') = \left|\{i \mid s(i) \ne s'(i)\}\right|$.) The solution $s_{\text{locopt}} \in S$ is said to be a locally optimal solution with respect to the neighbourhood N if $f(s') \ge f(s_{\text{locopt}})$ for every $s' \in N(s_{\text{locopt}})$.

The remaining part of this paper is organized as follows. In Section 1, we concern the basic features, principles, and extensions of the tabu search. The new modifications (enhancements) of TS for the QAP are discussed in Section 2. In Section 3, we present the results of the computational experiments with the various variants of the modifications proposed. Finally, Section 4 completes the paper with concluding remarks.

1. Tabu search meta-heuristic and its extensions

TS is based on the neighbourhood search with local-optima avoidance but in a rather deterministic way. The key idea of tabu search is allowing climbing moves when no improving neighbouring solution exists, i.e. a move is allowed even if a new solution from the neighbourhood is worse than the current one. More formally, TS starts from an initial solution s° in S. The process is then continued in an iterative way – moving from a solution s to a neighbouring one s'. At each step of the procedure, a subset $N^{\star}(s) \subseteq N(s)$ of the neighbours of the current solution is considered, and the move to the solution $s' \in N^{\star}(s)$ that improves most the objective function value f is chosen. Naturally, s' must not necessary be better than s: if there are no improving moves, the algorithm chooses the one that least degrades the objective function. In order to eliminate the returning to the solution just visited, the reverse move must be forbidden. This is done by storing the corresponding solution (move) (or its "attribute") in a memory (called a tabu list (T)). The tabu list keeps information on the last $h = |T|$ moves which have been done during the search process. h is called a tabu tenure (tabu list size). Thus, a move from s to s' is considered as tabu if s' (or its "attribute") is contained in T. This way of proceeding hinders the algorithm from going back to a solution reached within the last h steps. However, the straightforward prohibition may lessen the efficiency of the algorithm. For this reason, an aspiration criterion is introduced to permit the tabu status to be dropped under certain circumstances. Usually, a move from s to s' (no matter its status) is permitted if $f(s') < f(s^*)$, where s^* is the best solution found so far. The resulting decision rule can thus be described as follows: replace the current solution s by the new solution s' if

$$f(s') < f(s^*) \text{ or } \left(s' = \mathop{\arg\min}_{s'' \in N^{\star}(s)} f(s'') \text{ and } s' \text{ (or "attribute" of } s') \text{ is not tabu} \right). \quad (3)$$

The search process is stopped as soon as a termination criterion is satisfied. The best solution found serves as a result of the algorithm. For a more thorough discussion on the principles of TS, the reader is addressed to [5,6,9,11].

The straightforward implementation of TS typically faces severe difficulties: a huge number of local optima over the solution space, isolated local optima, presence of cycles (i.e. repeating sequences) of the search configurations, and the so-called "deterministic chaos" phenomenon. The last one is characterized by the situation in which "getting stuck" in local optima and cycles are absent but the search trajectory is still confined in a limited part of the solution space.

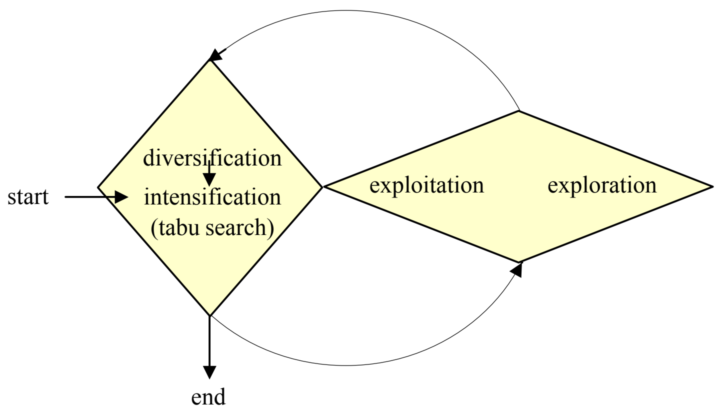


Figure 1. Generalized framework of I&D

In order to try to overcome these complications, a so-called iterated tabu search (ITS) strategy may be helpful. Roughly speaking, ITS is based on the intensification and diversification (I&D) policy (see [13,14,15]). The I&D methodology is distinguishing for four main components: intensification, diversification, exploitation, and/or exploration (see Figure 1). The intensification (local improvement algorithm) concentrates the search in localized portions of the solution space. (The standard tabu search algorithm outlined above actually plays the role of the intensification in the ITS approach.) The diversification is responsible for escaping from the current local optimum and moving towards unvisited so far solutions. Finally, exploitation and exploration may be viewed as alternative strategies for the selection of a candidate for the reconstruction (partial destruction), i.e. diversification. (The exploitation is achieved by choosing only the currently best local optimum as a candidate for the reconstruction, while in the case of exploration, each locally optimized solution can be considered as a potential candidate for the diversification.)

The intensification component remains one of the critical things by constructing competitive I&D-based algorithms. For this reason, we propose in the next section some improvements (enhancements), namely, to the intensification phase, i.e. the tabu search procedure within the ITS framework.

2. Modifications (enhancements) of tabu search for the QAP

In this section, we describe some modifications (enhancements) of the tabu search for the quadratic assignment problem. We will concern only the modifications related to the tabu search procedure itself, while the details of the ITS (I&D) paradigm can be found in [13,14,16]. The slightly modified version of the robust tabu search algorithm [20] serves as a "platform" ("starting point") for these enhancements. The new enhancements are as follows.

2. 1. Randomization of tabu search

The tabu search forbids some solutions (moves) from time to time. This fact means that certain portions of the search space are excluded from being visited. This can be seen as a serious disadvantage of the tabu search. One of the possible ways to get over this weakness is to minimize the restrictions, that is, it is desirable that the number of the tabu moves is as small as possibly. We propose a simple trick: the tabu status is ignored with a (very) small probability (even if the aspiration criterion does not hold). As to the QAP, we empirically found that the optimal value of this probability, α, is somewhere between 0.05 and 0.1 (we used $\alpha = 0.07$). As the tabu status is ignored randomly with a negligible probability, there is no danger that the cycles (cyclic trajectories of the search configuration) will occur.

2. 2. Tabu search with delay

It is obvious that in the early stages of the search (provided that the process starts from the statistically mean quality solution) the cycles are unlikely. For this reason, there is no need to restrict moves at the early iterations of the tabu search. That is the central idea of the approach. So, at the beginning, the tabu list stays unchanged, i.e. the tabu tenure is equal to zero. After K iterations are performed, the tabu search is continued in the standard way. K is defined as $\lfloor \beta \cdot n \rfloor$, where β is a user-defined parameter (delay factor) (we used $\beta = 0.7$).

2. 3. Relaxation

The tabu moves are the heart of the TS method. However, these tabu moves may also be viewed as a limitation of the search, especially in the cases when the search progresses and many moves are forbidden. As a consequence, some promising regions of the solution space remain unexplored. In order to try to diminish this negative effect, the strategy called a relaxation may be proposed. In the most effortless case, the relaxation is achieved by the simple clearing out the tabu list. For example, the tabu list is emptied each time a new locally optimal solution is encountered. After this, TS goes on in the ordinary way. The other variants of relaxation are possible, for example, the tabu list is cleared out periodically (with no taking into account the local optima). In our implementation, we applied the second variant. We wipe out the tabu list every $\lfloor \gamma \cdot \tau \rfloor$ iteration, where τ is the number of the tabu search iterations (defined a priori by the user) and γ is a parameter (relaxation factor) from the interval (0, 1) (we used $\gamma = \frac{1}{3}$).

2. 4. Alternative intensification

It was observed that TS yields slightly better results if it embeds an improvement mechanizm of the other kind. We call this policy as an alternative intensification. We obtained quite satisfactory results by using a pure descent local search (DLS) in the role of such kind intensification (the details of the DLS procedure are described in [16]). The idea of alternative intensification is to temporally interrupt the basic intensification procedure, i.e. the TS procedure and switch to the alternative procedure. After a while (for example, a local optimum is found) one returns to the basic procedure, and so on. The difference between the relaxation and alternative intensification is that, in the last case, the tabu list is not emptied (it is updated while in the alternative procedure, i.e. all the moves performed during the descent local search are stored in the tabu list); it is simply ignored during the execution of the DLS procedure, but it is taken into account again after returning to the basic procedure. The frequency of the alternative intensification is controlled by a special parameter, which depends on the tabu list size; in particular, the value of this parameter is equal to $\lfloor \delta \cdot h \rfloor$, where h is the tabu tenure and $\delta = 3$.

Encouraged the good results, we also tried another intriguing idea: we used the tabu search procedure (with a very limited number of iterations) instead of the DLS procedure, so we obtained something like using two tabu lists and switching between them periodically. So far, we unfortunately get the slightly worse results than in the case of DLS.

2. 5. Avoiding stagnation

At the beginning of the search process, the new (record-breaking) locally optimal solutions are encountered frequently; thus, the search converges very rapidly. However, as the search progresses the records, i.e. new local optima become more and more rare; it takes an extremely long time to find a new better solution in the later phases. This phenomenon is known as a stagnation of the search. A naive but seemingly helpful trick is introduced to try to lessen a detrimental influence of the stagnation. In fact, we only imitate the stagnation avoidance (prevention). So, the idea is based on the assumption that the stagnation intervals are quite small in the early periods of the search. Thus, the probability of finding of new (better) solutions in these cases is approximately the same. Let I_1, I_2, ..., I_k, I_{k+1}, ... be some (time) intervals of the search with L being an interval length. If, occasionally, a new (better) solution was not found during the interval I_k, then we assume that it will be found very likely during the interval I_{k+1}. If I_k is the last interval, we simply extend the search, i.e. increase the number of the iterations by some value. For example, if the current iteration number is equal to M, we add a value of ε to M, obtaining $M+\varepsilon$ iterations; hopefully, a new solution will be found during these extra iterations (see Figure 2). Of course, if the new local optimum was found during the last I_k iterations, the number of iterations M remains constant. In our algorithm, $L=\lfloor \omega\cdot\tau \rfloor$ and $\varepsilon=\lfloor 0.3\tau \rfloor$, where τ is the number of iterations of the tabu search and ω ($\omega \in (0, 1)$) is an a priori parameter (we used $\omega = 0.4$). The scheme described seems to bring effect only if M is relatively small. Fortunately, this is just the case in the iterated tabu search approach.

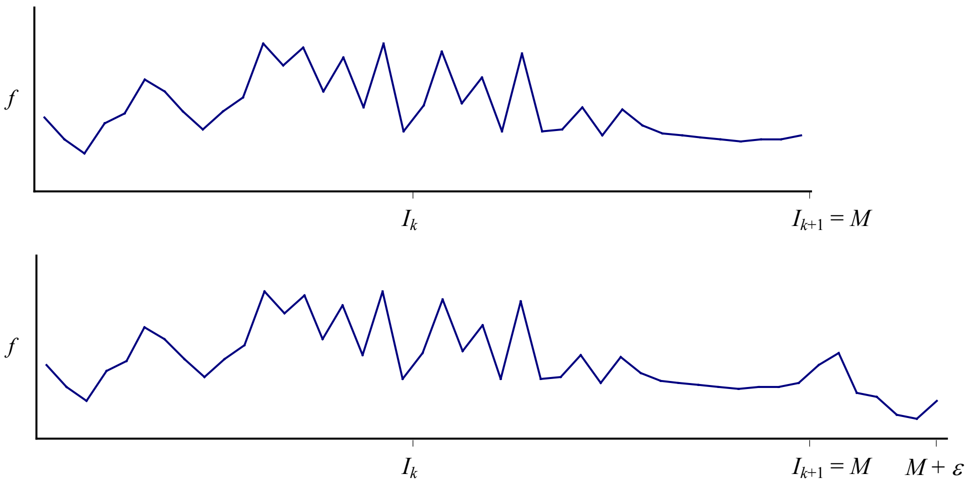


Figure 2. Avoiding stagnation: illustration of a possible situation

The following short notations will be used for the new modifications:

- ETS_{Ra} (enhanced tabu search (ETS) with randomization);
- ETS_D (ETS with delay);
- ETS_R (ETS with relaxation);
- ETS_{AI} (ETS with alternative intensification);
- ETS_{AS} (ETS with avoiding stagnation).

We also implemented the additional variant that covers all the above components (i.e. the randomization, delay, relaxation, alternative intensification, and stagnation avoidance). The resulting algorithm is called a combined enhanced tabu search – ETS_C. The detailed template of ETS_C is presented in an algorithmic language like form in Figure 3. Recall that the skeleton of this template is very similar to the one of the robust tabu search algorithm due to Taillard [20] slightly modified by Misevicius [14].

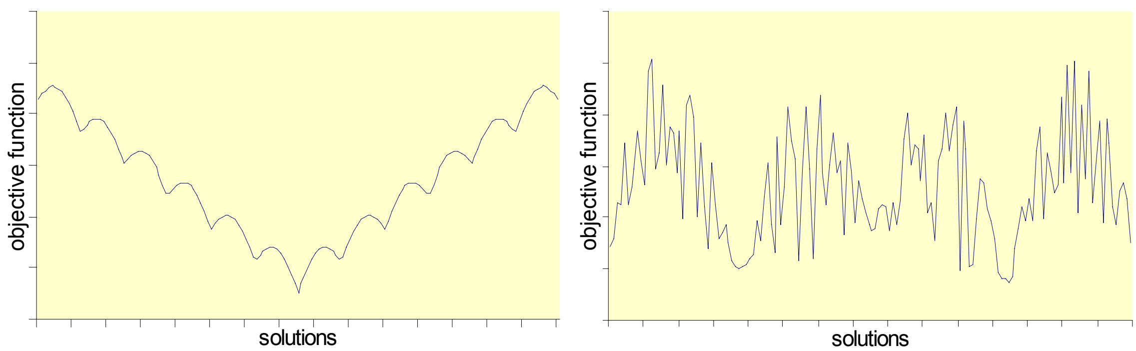
3. Computational experiments

In this section, we present the results of the experimental comparison of the algorithms outlined above. In the experiments, the instances (benchmarks) of the QAP taken from the well-known library QAPLIB [3] are used. We handle only the random instances since these problems are most difficult for the heuristics. The random instances were generated by Taillard in 1991 [20], but still remain a great challenge for the designers of the heuristic algorithms (this is especially true for the larger problems; to this date, they seem to be not practically solvable even to pseudo-optimality).

The random instances are generated randomly according to the uniform distribution. In QAPLIB, they are denoted by tai20a, tai25a, tai30a, tai35a, tai40a, tai50a, tai60a, tai80a, and tai100a (or briefly tai∗a) (the corresponding numeral (20, 25, and so on) in the instance name denotes the size of the problem). Based on Taillard's classification [21], these instances belong to the class of regular, unstructured problems. The regularity of the instances tai∗a are due to the regular, i.e. uniformly distributed values of the data matrices. On the other hand, these instances are unstructured in that sense that there is no "structureness" in the solution space. The landscapes of such problems are extremely rugged with narrow basins of attractions and a huge number of isolated local optima. This is a contrast to the structured problems with wide basins of attractions ("big valleys") (see Figure 4).

```
function CombinedEnhancedTabuSearch(π°, n, τ, h, α, β, γ, δ, ω);
   // input: π° – the initial solution, n – the problem size, τ – the number of iterations, h – the tabu tenure,  //
   //           α, β, γ, δ, ω – the control parameters  //
   // output: π• – the best solution found  //
   delay_interval := ⌊β·n⌋; relaxation_interval := ⌊γ·τ⌋; intensification_interval := ⌊δ·h⌋;
   stagnation_interval := ⌊ω·τ⌋;
   π := π°; π• := π°;
   for i := 1 to n−1 do
     for j := i+1 to n do calculate objective function difference d_ij := Δz(π,i,j);
   T := 0; i := 1; j := 1; k := 1; k' := 1; k'' := 1; k_locopt := 1; improved := FALSE;
   while (k ≤ τ) or improved = TRUE then begin  // main cycle  //
      d_min := ∞;
      for l := 1 to |N2| do begin  // find the best move that is not tabu or aspired  //
         i := IF(j < n,i,IF(i < n−1,i+1,1)) ;   j := IF(j < n, j+1,i+1) ;
        tabu := IF(t_ij ≥ k and RANDOM() ≥ α, TRUE, FALSE);
        aspired := IF(z(π) + d_ij < z(π•) and NOT(tabu), TRUE, FALSE);
        if (d_ij < d_min and NOT(tabu)) or aspired then begin d_min := d_ij; u := i; v := j end
      end;  // for l  //
      improved := IF(d_min < 0, TRUE, FALSE);
      if k − k' ≥ relaxation_interval then begin  // relaxation  // T := 0; k' := k end;
      if d_min < ∞ then begin
         π := π ⊕ p_uv ;  // replace the current permutation by the new one  //
         for l := 1 to n−1 do for m := l+1 to n do update difference d_lm;
         if k > delay_interval then t_uv := k + h  // make the move p_uv tabu  //
      end;  // if  //
      if z(π) < z(π•) then begin π• := π; k_locopt := k end;  // save the best so far solution  //
      if (improved and (k − k'' ≥ intensification_interval)) or
         ((z(π) = z(π•) and (k − k'' ≥ ½·intensification_interval)) then begin
         switch to the alternative intensification;
         k'' := k
      end;  // if  //
      k := k + 1;
      if (k > τ) and (k − k_locopt ≥ stagnation_interval) then begin  // extending the search  //
         τ := ⌊1.3·τ⌋; k_locopt := k
      end  // if  //
   end;  // while  //
   return π•
end.
```

Figure 3. Template of the combined enhanced tabu search algorithm for the QAP.
Note. p_{uv} (u, v = 1, 2, ..., n) denotes a move which simply swaps the u-th and v-th element in the given permutation; thus, $\pi' = \pi \oplus p_{uv} \in N_2(\pi)$ and $\pi'(u) = \pi(v)$, $\pi'(v) = \pi(u)$, $1 \le u, v \le n \wedge v-u \ge 1$, where π is the current permutation, and π' – the neighbouring permutation



(a) (b)

Figure 4. Fragments of the illustrative landscapes for the structured (a) and unstructured (b) problems

The algorithms used in the experimentation are as follows: the robust tabu search (RoTS) [20]; the reactive tabu search (ReTS) [1]; the original iterated tabu search (in [14], it is entitled as enhanced tabu search (ETS)), as well as the actual modifications: ETS_{Ra}, ETS_D, ETS_R, ETS_{AI}, ETS_{AS}, and ETS_C (remind that the last six variants are within the ITS framework). RoTS, ReTS, and ETS are chosen as the competitors because they belong to the most powerful heuristic algorithms, in particular, for random QAPs.

The efficiency measure for the algorithms is the average deviation of solutions obtained from the best known solution – $\bar{\delta}$ ($\bar{\delta} = 100(\bar{z} - \breve{z})/\breve{z}\,[\%]$, where $\bar{z}$ is the average objective function value over 10 restarts (single applications of a given algorithm to a given instance), and $\breve{z}$ is the best known value (BKV) of the objective function). In the experimental comparison, equated conditions are created: all the algorithms use the identical initial solutions and require approximately the same execution (CPU) time. For the sake of more fairness, we carried out three sets of experiments (namely, short time runs, medium time runs, and long time runs). The results of comparison of the algorithms are presented in Tables 1–3.

Table 1. Comparison of the tabu search algorithms for the random QAPs: shorter run results. The best results obtained are printed in bold face. CPU times per restart are given in seconds. 3 GHz PENTIUM computer was used in the experiments

Instance	BKV	$\bar{\delta}$									CPU time
		RoTS	ReTS	ETS	ETS_{Ra}	ETS_D	ETS_R	ETS_{AI}	ETS_{AS}	ETS_C	
tai20a	703482 [a]	0.279	0.272	0.362	0.312	0.216	0.242	0.185	0.273	**0.174**	*0.1*
tai25a	1167256 [a]	0.547	0.686	0.438	0.338	0.464	0.408	**0.255**	0.420	0.352	*0.2*
tai30a	1818146 [a]	0.717	0.554	0.314	**0.182**	0.276	0.371	0.190	0.335	0.330	*0.3*
tai35a	2422002 [a]	1.011	0.857	0.609	0.578	0.621	0.465	0.482	**0.448**	0.481	*0.4*
tai40a	3139370 [a]	1.028	0.926	0.836	0.653	0.636	0.609	0.762	0.623	**0.571**	*0.6*
tai50a	4941410 [a]	1.507	1.076	1.075	**0.881**	0.961	1.009	0.998	0.983	0.926	*0.9*
tai60a	7205962 [b]	1.502	1.183	1.167	**0.852**	1.082	1.135	1.041	1.086	0.969	*1.2*
tai80a	13546960 [b]	1.162	0.840	0.918	0.665	0.755	0.721	0.712	0.772	**0.639**	*2.3*
tai100a	21123042 [b]	1.016	0.684	0.800	0.639	0.700	0.652	0.697	0.653	**0.550**	*3.6*
Average:		0.877	0.708	0.652	0.510	0.571	0.561	0.532	0.559	**0.499**	

[a] comes from [3]; [b] comes from [14].

Table 2. Comparison of the tabu search algorithms for the random QAPs: medium run results. The best results obtained are printed in bold face. CPU times per restart are given in seconds. 3 GHz PENTIUM computer was used in the experiments

Instance	BKV	$\bar{\delta}$									CPU time
		RoTS	ReTS	ETS	ETS_{Ra}	ETS_D	ETS_R	ETS_{AI}	ETS_{AS}	ETS_C	
tai20a	703482	0.061	0.061	**0**	**0**	0.030	0.091	0.030	0.030	**0**	*0.5*
tai25a	1167256	0.242	0.298	0.043	0.037	0.067	0.073	**0**	0.037	0.015	*1.0*
tai30a	1818146	0.347	0.187	0.151	0.042	0.078	0.057	0.149	0.083	**0.041**	*1.5*
tai35a	2422002	0.521	0.354	0.326	0.300	0.228	0.189	**0.151**	0.217	0.205	*2.0*
tai40a	3139370	0.731	0.598	0.551	0.446	0.466	0.497	**0.394**	0.427	0.441	*3.0*
tai50a	4941410	1.206	0.789	0.882	0.674	0.746	0.750	0.643	**0.613**	0.695	*4.5*
tai60a	7205962	1.219	0.925	0.909	0.686	**0.681**	0.749	0.858	0.719	0.726	*6.0*
tai80a	13546960	0.940	0.549	0.749	0.455	0.481	**0.438**	0.457	0.444	0.454	*11.6*
tai100a	21123042	0.960	0.494	0.652	0.395	0.388	0.470	0.501	0.513	**0.356**	*19.0*
Average:		0.623	0.426	0.426	0.304	0.317	0.331	0.318	0.308	**0.293**	

Table 3. Comparison of the tabu search algorithms for the random QAPs: longer run results. The best results obtained are printed in bold face. CPU times per restart are given in seconds. 3 GHz PENTIUM computer was used in the experiments

Instance	BKV	$\bar{\delta}$									CPU time
		RoTS	ReTS	ETS	ETS_{Ra}	ETS_D	ETS_R	ETS_{AI}	ETS_{AS}	ETS_C	
tai20a	703482	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	*2.5*
tai25a	1167256	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	*5.0*
tai30a	1818146	0.057	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	*7.5*
tai35a	2422002	0.242	0.189	0.061	**0**	0.025	**0**	**0**	**0**	**0**	*10.0*
tai40a	3139370	0.536	0.398	0.324	0.300	0.307	0.297	0.293	0.275	**0.264**	*15.0*
tai50a	4941410	0.976	0.692	0.612	0.513	0.476	0.528	0.465	**0.439**	0.463	*23.0*
tai60a	7205962	1.059	0.763	0.749	0.427	0.519	0.535	0.536	**0.410**	0.416	*30*
tai80a	13546960	0.807	0.411	0.534	0.251	0.284	0.255	0.338	0.316	**0.216**	*58*
tai100a	21123042	0.837	0.269	0.446	0.224	0.195	0.164	0.142	0.226	**0.139***	*100*
Average:		0.451	0.272	0.273	0.172	0.181	0.178	0.177	0.167	**0.150**	

* during the experimentation with ETS_C, we were successful in discovering new record-breaking solution for the instance tai100a; the new objective function value is equal to **21087588**.

It can be seen from Tables 1–3 that our new modifications (enhancements) of the tabu search produce very strong results for the random QAPs. In particular, it seems that the combined modified tabu search (ETS_C) evidently outperforms the remaining versions. This is true for both short and long runs. So, this indicates a high stability of the ETS_C algorithm. Note, however, that the results of the randomized tabu search (ETS_{Ra}) are very close to the ones of ETS_C. It looks like as if the tabu search with the stagnation avoidance yields quite encouraging results, too. On the whole, all the new modified algorithms achieved better results than the earlier "pure" enhanced (iterated) tabu search algorithm due to Misevicius (ETS in Tables 1–3), which, in turn, appears to be slightly better than the reactive tabu search due to Battiti and Tecchiolli, in particular, at shorter runs; at that time, it is much more better than the robust tabu search due to Taillard. Note that the results of $ETS_{Ra} \div ETS_C$ could be improved by a more accurate tailoring of the control parameters.

The following point should also be stressed. By juxtaposing of the proposed variants, one obtains additional modifications (hybrids). For example, we can get the randomized tabu search with delay, the randomized tabu search with delay and relaxation, and so on. Totally, we can obtain $2^5 - 6 = 26$ (!) additional variants. In this work, these modifications are omitted for the sake of brevity. However, we believe that some of these hybrids may hide promising solutions (even better than those presented in the current paper). Therefore, we do hope that the investigation of these remaining modifications will be a proper topic for the future research. The testing of the performance of the new modifications on the other types of the quadratic assignment problem, in particular, the real-life (like) problems would be worthwhile as well.

4. Concluding remarks

In this paper, the issues related to the improvement of the tabu search in the context of the QAP are discussed. Investigated are five variants, more precisely, extensions to the tabu search, in particular, the randomized tabu search, the tabu search with delay, the tabu search with relaxation, the tabu search combined with the alternative intensification, and the tabu search that involves a mechanism for the stagnation avoidance (prevention). Finally, a combined tabu search algorithm which integrates all these features is proposed. The new improvements (modifications) are on the basis of the iterated tabu search framework.

All these variants, together with the earlier tabu search algorithms, were tested on the random QAP instances, which still remain a challenge for the researchers. The results from the experiments show high performance of the modifications proposed. All the new modifications outperform the pure iterated tabu search, probably, one of the most powerful heuristic algorithms for random QAPs so far. So, incorporating the additional components (features) into the standard tabu search has a considerable positive influence on the resulting efficiency (at least for the random QAPs). Also, it can be seen that the combined enhanced algorithm is superior to the variants where only single components are incorporated. The promising efficiency of the combined enhanced TS algorithm is also confirmed by the fact that the new best known solution for the instance tai100a was found.

Further extensions with the focus on the probabilistic behaviour of the tabu search may be proposed, among them, 1) using a cascade of the random levels (thresholds) instead of the single (stationary) random level, 2) applying randomization of the tabu memory (instead of the random decision rule). It is also worthy trying a so-called periodic tabu search method, i.e. changing the tabu tenure (tabu list size) periodically rather than using a purely random (chaotic) way of changing (which is the case in the robust tabu search). These extensions, as well as the development of additional innovative components of TS may be the subject of the future work.

References

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

[2] **R.E. Burkard, E. Çela, P.M. Pardalos, L. Pitsoulis.** The quadratic assignment problem. *In D.Z.Du, P.M. Pardalos (eds.), Handbook of Combinatorial Optimization*, *Kluwer, Dordrecht*, 1998, *Vol.*3, 241–337.

[3] **R.E. Burkard, S. Karisch, F. Rendl.** QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, 1997, *Vol.*10, 391–403. [*See also* `http://www.seas.upenn.edu/qaplib/`.]

[4] **E. Çela.** The Quadratic Assignment Problem: Theory and Algorithms. *Kluwer, Dordrecht*, 1998.

[5] **M. Gendreau.** Recent advances in tabu search. *In C.C. Ribeiro, P. Hansen (eds.), Essays and Surveys in Metaheuristics, Kluwer, Boston,* 2001, 369-377.

[6] **M. Gendreau.** An introduction to tabu search. *In F. Glover, G. Kochenberger (eds.), Handbook of Metaheuristics, Kluwer, Norwell*, 2002, 37–54.

[7] **F. Glover.** Tabu search: part I. *ORSA Journal on Computing*, 1989, *Vol.*1, 190–206.

[8] **F. Glover.** Tabu search: part II. *ORSA Journal on Computing*, 1990, *Vol.*2, 4–32.

[9] **F. Glover, M. Laguna.** Tabu Search. *Kluwer, Dordrecht*, 1997.

[10] **P. Hansen, B. Jaumard.** Algorithms for the maximum satisfiability problem. *RUTCOR Research Report, Rutgers University, USA*, 1987, 43–87.

[11] **A. Hertz, E. Taillard, D. de Werra.** Tabu search. *In E.H.L. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, Chichester,* 1997, 121–136.

[12] **T. Koopmans, M. Beckmann.** Assignment problems and the location of economic activities. *Econometrica*, 1957, *Vol.*25, 53–76.

[13] **A. Misevicius.** Ruin and recreate principle based approach for the quadratic assignment problem. *In E. Cantú-Paz, J.A .Foster, K. Deb et al. (eds.), Lecture Notes in Computer Science, Vol.*2723*, Genetic and Evolutionary Computation – GECCO* 2003, *Proceedings, Part* I, *Springer*, *Berlin-Heidelberg*, 2003, 598–609.

[14] **A. Misevicius.** A tabu search algorithm for the quadratic assignment problem. *Computational Optimization and Applications*, 2005, *Vol.*30, 95–111.

[15] **A. Misevičius.** Iterated tabu search for the quadratic assignment problem. *Working Paper, Kaunas University of Technology, Lithuania*, 2005.

[16] **A. Misevičius, A. Lenkevičius.** A modification of tabu search and its application to the quadratic assignment problem. *Information Technology and Control*, 2003, *No.*2(27), 12–20.

[17] **P.M. Pardalos, F. Rendl, H. Wolkowicz.** The quadratic assignment problem: a survey and recent developments. *In P.M. Pardalos, H. Wolkowicz (eds.), Quadratic Assignment and Related Problems. DIMACS Series an Discrete Mathematics and Theoretical Computer Science, Vol.*16, *AMS, Providence*, 1994, 1-41.

[18] **E. Rolland, H. Pirkul, F. Glover.** Tabu search for graph partitioning. *Annals of Operations Reserch*, 1996, *Vol.*63, 209–232.

[19] **S. Sahni, T. Gonzalez.** P-complete approximation problems. *Journal of ACM*, 1976, *Vol.*23, 555–565.

[20] **E. Taillard.** Robust taboo search for the QAP. *Parallel Computing*, 1991, *Vol.*17, 443–455.

[21] **E. Taillard.** Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 1995, *Vol.*3, 87–105.

[22] **P. Thomas, S. Salhi.** A tabu search heuristic for the resource constrained project scheduling problem. *Journal of Heuristics*, 1998, *Vol.*4, 123–139.

[23] **N.A. Wassan, I.H. Osman.** Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, 2002, *Vol.*53, 768–782.