An Empirical Study of Extended Guided Local Search on the Quadratic Assignment Problem

Patrick Mills and Edward Tsang

Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom
{millph,edward}@essex.ac.uk
http://cswww.essex.ac.uk/CSP/

Abstract. In this paper, we show how an Extended Guided Local Search can be applied to the Quadratic Assignment Problem and show the extensions can improve its performance. GLS is a general, penalty-based meta-heuristic, which sits on top of local search algorithms, to help guide them out of local minima. GLS has been shown to be successful in solving a number of practical real life problems, such as the travelling salesman problem, BT's workforce scheduling problem, the radio link frequency assignment problem, the SAT problem, the weighted MAX-SAT problems, and the vehicle routing problem. We present empirical results of applying several extended versions of Guided Local Search to the Quadratic Assignment Problem, and show that these extensions can improve the range of parameter setting within which Guided Local Search performs well. Finally, we compare the results of running our Extended Guided Local Search with some state of the art algorithms for the QAP.

1 Introduction

Guided Local Search [35] has been applied to a number of real life problems, including the SAT problem and the weighted MAX-SAT problems [20], the vehicle routing problem [11], the radio link frequency assignment problem [40], function optimization [41] and the travelling salesman problem [42].

GLS is a meta-heuristic that sits on top of local search procedures and helps them escape from local minima. GLS can be seen as a generalization of the GENET neural network [32,5,6] for solving constraint satisfaction problems and optimization problems. Recently, it has been shown that GLS can be put on top of a specialized Genetic Algorithm, resulting in the Guided Genetic Algorithm [18]. GGA has been applied to a number of problems, including the processor configuration problem [13,14,16], the generalized assignment problem [15] and the radio link frequency assignment problem [17,19]. In this paper, we show how GLS and some extensions of GLS can successfully applied to the Quadratic Assignment Problem.

1.1 The Quadratic Assignment Problem

The Quadratic Assignment Problem [3] is one of the hardest groups of problems in combinatorial optimization, with many real world applications and has been the focus of a lot of successful research into heuristic search methods. The problem can be formally stated as in (1).

$$\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j} \qquad (1)$$

where:

- n is the size of the problem (i.e. number of facilities or locations),
- π is a permutation, where π_i is the i^{th} element in permutation π,
- a and b are the n * n distance and flow matrices.

The problem is to find a permutation π (which represents which facilities are placed at which locations), which minimizes the sum of the distance times the flow between different facilities. The element a_{ij} of matrix a, represents the distance between location i and location j. The element $b_{\pi_i\pi_j}$ represents the flow between facility πi and πj. When a_{ij} is multiplied by $b_{\pi_i\pi_j}$, the cost of placing facility πi at location i and facility πj and location j, is obtained. Thus, by summing all the terms together, the total cost of the whole permutation of location-facility assignments is obtained.

Both exact and inexact algorithms have been proposed for solving the Quadratic Assignment Problem (for a survey, see [21]). The exact algorithms have the disadvantage that they can only solve relatively small QAPs (n • 20), whereas the inexact methods can deal with much larger problems. The heuristic methods, which have been used to solve the QAP, include Robust Tabu Search [27,28], Reactive Tabu Search [2], Simulated Annealing [43,30], a Genetic Hybrid Algorithm [10], ant algorithms [26, 29] and various others [1]. In this paper, we show how Guided Local Search can be applied to the Quadratic Assignment problem, and present empirical results showing two extensions of Guided Local Search which can increase the range of parameters under which good results are obtained.

2 Guided Local Search

Guided local search (GLS), (see [35] for a more detailed description) sits on top of a local search algorithm. When the given local search algorithm settles in local optimum, GLS changes the objective function using a scheme that will be explained below. Then the local search will search using the modified objective function, which is designed to bring it out of local optimum. The key is in the way that the objective function is modified.

Solution features are defined to distinguish between solutions with different characteristics, so that bad characteristics can be penalized by GLS, and hopefully removed by the local search algorithm. The choice of solution features therefore depends on the type of problem, and also to a certain extent on the local search algorithm. Each feature, f_i defined must have the following components:

- An Indicator function, I_i indicating whether the feature is present in the current solution or not:

$$I_i(s) = \begin{cases} 1, & \text{solution } s \text{ has property i} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

- A cost function $c_i(s)$, which gives the cost of having the feature present in a solution.
- A penalty p_i, initially set to 0, used to penalize occurrences of the feature, in local minima.

2.1 Selective Penalty Modifications

When the Local Search algorithm returns a local minimum, *s*, which is not a legal solution, GLS penalizes (increments the penalty of the feature) all the features present in that solution which have maximum utility, $util(s,f_i)$, as defined in (3).

$$util(s, f_i) = I_i(s) \cdot \frac{c_i(s)}{1 + p_i} \quad (3)$$

The idea is to penalize features, which have high costs first, although the utility of doing so decreases as the feature is penalized more and more times.

2.2 Augmented Cost Function

GLS uses an augmented cost function (3), to allow it to guide the Local Search algorithm out of the local minimum, by penalizing features present in that local minimum. The idea is to make the local minimum more costly than the surrounding search space, where these features are not present.

$$\boldsymbol{h}(\boldsymbol{s}) = \boldsymbol{g}(\boldsymbol{s}) + \lambda \cdot \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{I}_{\boldsymbol{i}}(\boldsymbol{s}) \cdot \boldsymbol{p}_{\boldsymbol{i}} \tag{4}$$

The parameter λ may be used to alter the intensification of the search for solutions. A higher value for λ will result in a more diverse search, where plateaus and basins in the search are searched less carefully; a low value will result in a more intensive search for the solution, where the basins and plateaus in the search landscape are searched with more care.

2.3 Local Search for the QAP

The Quadratic assignment problem can be formulated as a local search algorithm, using the objective function in the previous section, and searching the space of possible permutations. The local search neighborhood is simply the set of possible permutations resulting from the current permutation with any two of the elements swapped around.

2.4 Efficient Local Search and Neighbourhood Updating for the QAP

The new value of the objective function after a swap can be efficiently incrementally updated in approximately $O(n^2)$ time using (5) and (6) (for asymmetric QAPs, see [28]) or (7) and (8) (for symmetric QAPs, the symmetry in the matrices can be taken advantage of to speed up neighborhood updating by a factor of about 4, see [2]).

$$\Delta(\pi', r, s) = \begin{matrix} (a_{rr} - a_{ss})(b_{\pi_s\pi_s} - b_{\pi_r\pi_r}) + (a_{rs} - a_{sr})(b_{\pi_s\pi_r} - b_{\pi_r\pi_s}) + \\ \sum_{k=1, k\neq r,s}^{n} ((a_{kr} - a_{ks})(b_{\pi_k\pi_s} - b_{\pi_k\pi_r}) + (a_{rk} - a_{sk})(b_{\pi_s\pi_k} - b_{\pi_r\pi_k})) \end{matrix} \tag{5}$$

$$\forall u,v \bullet u,v \neq r,s : \Delta(\pi', u, v) = \begin{matrix} \Delta(\pi,u,v) + (a_{ru} - a_{rv} + a_{sv} - a_{su})(b_{\pi'_s\pi'_u} - b_{\pi'_s\pi'_v} + b_{\pi'_r\pi'_v} - b_{\pi'_r\pi'_u}) + \\ (a_{ur} - a_{vr} + a_{vs} - a_{us})(b_{\pi'_u\pi'_s} - b_{\pi'_v\pi'_s} + b_{\pi'_v\pi'_r} - b_{\pi'_u\pi'_r}) \end{matrix} \tag{6}$$

$$\Delta(\pi', r, s) = 2 \sum_{k=1, k\neq r,s}^{n} (a_{rk} - a_{sk})(b_{\pi_s\pi_k} - b_{\pi_r\pi_k}) \tag{7}$$

$$\forall u,v \bullet u,v \neq r,s : \Delta(\pi', u, v) = \begin{matrix} \Delta(\pi,u,v) + \\ 2(a_{ru} - a_{rv} + a_{sv} - a_{su})(b_{\pi'_s\pi'_u} - b_{\pi'_s\pi'_v} + b_{\pi'_r\pi'_v} - b_{\pi'_r\pi'_u}) \end{matrix} \tag{8}$$

$$\Delta(\pi', r, s) = 2 \sum_{k=1, k\neq r,s}^{n} (b_{\pi_s\pi_k} - b_{\pi_r\pi_k}) \tag{9}$$

$$\forall u,v \bullet u,v \neq r,s : \Delta(\pi', u, v) = \Delta(\pi,u,v) + 2(b_{\pi'_s\pi'_u} - b_{\pi'_s\pi'_v} + b_{\pi'_r\pi'_v} - b_{\pi'_r\pi'_u}) \tag{10}$$

(where π' = the permuation π with elements r and s swapped)

In addition to this, for Taillard's Grey density problems the neighborhood may be restricted to swapping elements from the first m values in the permutation with the last n-m values in the permutation and may be calculated and updated more efficiently using (9) and (10) (as explained in [28]).

2.5 Features for the QAP

The QAP was found to be a difficult problem to solve for GLS. This is because there is only one obvious choice for the feature set: facility-location assignments[1]. These can be penalized and all the penalties can be kept in an N by N matrix. The cost of a particular facility-location assignment is the sum of the constituent parts of the objective function that it is involved in (see below):

$$C(i,\pi_i) = \sum_{j=0}^{n} a_{i,j} b_{\pi_i,\pi_j} \qquad (11)$$

2.6 A basic GLS for the QAP

```
GLSQAP(λcoeff)
{
           N   N            N   N
           Σ   Σ  a_ij  ×   Σ   Σ  b_ij
          i=0 j=0          i=0 j=0
 K =  ------------------------------------
                      n^4
 π = randomly generated permutation with no repetitions

 Do
 {
   //the second argument is a function
                                        n-1
   π = LocalSearch(π,  h + λcoeff × K ×  Σ  p(i) )
                                        i=0

   For each (i,πi), such that C(i,πi)/(1 + p(i,πi)) is maximised
     p(i,πi) = p(i,πi) + 1
 }
 While (Not termination criteria)

 Return π
}

LocalSearchQAP(π, g)
{

 sideways = 0

 While ((there is a downwards move w.r.t. g(π)) or
        (there is a sideways move w.r.t. g(π) and sideways < 2)
         and termination criteria is not met )
 {
   π = π with the elements πi and πj swapped such that Δg(π with πi and
```

[11] We did try to use pairs of facility-location assignments, with the flow between the facilities as the cost, but this meant there were a too many (N^4) features to store and incremental updating of the neighbourhood also became too expensive (more than 4 times slower) for larger problems, even with very "lazy" schemes for neighbourhood updating.

```
        πj swapped) is minimised (ties are broken randomly)

    If (Δg(π) == 0)
     sideways = sideways + 1
    Else
     sideways = 0

    Update the neighbourhood
  }

  return π

 }
```

3 Guided Local Search extensions

Whilst applying Guided Local Search to the QAP, we tried various schemes in an attempt to further improve Guided Local Search and try to understand why those schemes might work.

3.1 Aspiration criteria

Aspiration criteria (as used in the tabu search framework, [8,9]), is when a move is allowed, even when it would normally be tabu, usually due to it improving the current best solution. Intuitively, this is a good idea, since it would be stupid to avoid making a move just because it was tabu, if it gave us a new best solution. In GLS, we have penalties rather than a tabu list, so aspiration criteria means ignoring the penalties, if there is a move which can produce a new best solution. More formally, we can modify the local search algorithm to look like this with aspiration criteria:

```
LocalSearchQAPAspiration(π, g, h)
{
  sideways = 0
  While((there is a downwards move w.r.t. g(π)) or
        (there is a sideways move w.r.t. g(π) and sideways < 2)
         and termination criteria is not met )
  {
    //NOTE: the first term with the original objective function
    //is the aspiration criteria, the second is the standard
    //GLS, minimizing the augmented objective function
    If there exists a move, such that
    h(π) + Δh(π with πi and πj swapped) < best cost so far
      π = π with the elements πi and πj swapped such that Δh(π with πi
          and πj swapped) is minimised (ties are broken randomly)
    Else
      π = π with the elements πi and πj swapped such that Δg(π with πi
          and πj swapped) is minimised, ties are broken randomly

    If (Δg(π) == 0)
      sideways = sideways + 1
    Else
      sideways = 0

    Update the neighbourhood
   }
   return π
}
```

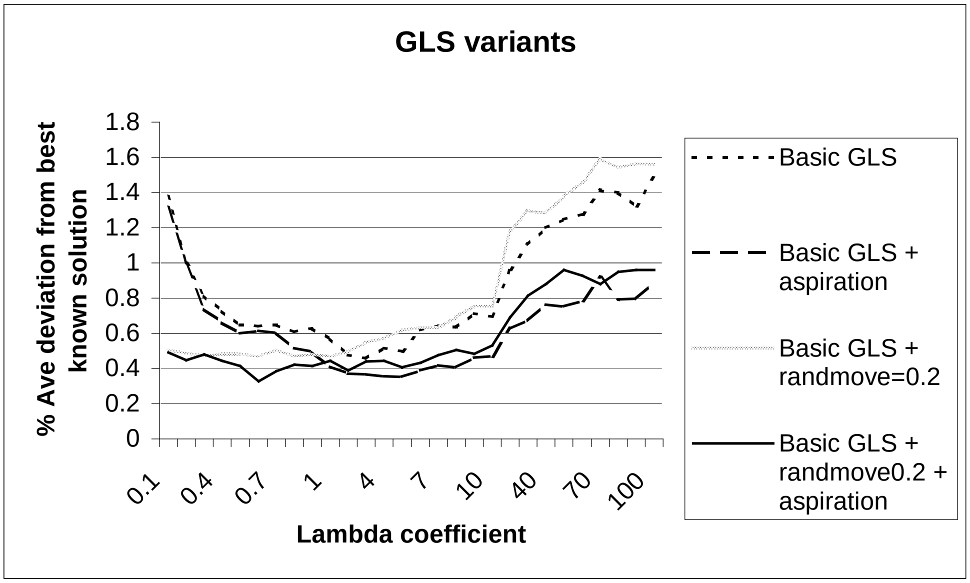

Fig. 1. GLS variants over a range of lambda coefficients values on small to medium sized QAPLib [4] problems, average of 10 runs, 1000 * N repairs (N = problem size)

We have found that aspiration criteria improves the performance of GLS in terms of the best cost solution found, particularly when large values of lambda are used (we have found using paired sample t-tests that these results were statically significant), and also in terms of how many iterations were required to find a solution of a given cost.

We theorised that *aspiration criteria works because it allows us to focus on minimising the original objective function at critical points during the search and also allows us to find more new best found solutions, that might otherwise be ignored.* This is particularly important when the lambda coefficient is large, as any penalty will have a larger effect on the local search algorithm, and is a plausible explanation for the improved performance of GLS when the lambda coefficient is large.

We ran controlled experiments, to try and substantiate this theory of why aspiration criteria works. The first was to allow a GLS without aspiration to follow the standard GLS scheme p% of the time and 100-p% of the time choose a move according the original objective function. We found that simply allowing GLS to ignore penalties p% of the time, does not result in an increase in performance, and so is not the sole reason for the success of aspiration criteria. The second experiment was to run GLS, allowing a random move to be chosen from the neighbourhood p% of the time, with the normal GLS scheme being followed the rest of the time. This was to check whether or not GLS was simply able to move into areas of the search space which would otherwise have been difficult to reach, due to penalties restricting GLS moves, when aspiration criteria was added. These experiments gave rise to an increase in the performance of GLS when small values of lambda were used, although the increase in performance when the lambda coefficient was large, that occurred with aspiration criteria did not occur. In fact, from looking at the average entropy (this is a measure of the spread, 0 only one facility-location assignment was visited for a particular element in the permutation, 1 would mean all labels were facility-location assignments were present in the same quantities, see, e.g. [3] for the definition of entropy) of facility-location assignments, we observed that random moves had a completely different effect from aspiration criteria in that they allowed GLS to diversify its search, when lambda was too small, whereas aspiration criteria left the entropy almost identical to before.

So the reason that aspiration criteria produced better results was not just because it allowed GLS to occasionally ignore the penalty term in the augmented objective function. During the runs of GLS with and without aspiration criteria we also recorded the average cost of each of the best-found solutions over each run of GLS. We found that when aspiration criteria was used that this value was substantially lower than when aspiration criteria was not used. We also found that GLS with aspiration found more “best found so far" solutions, than GLS without aspiration. This suggests that GLS with aspiration works as we theorised, because it allows GLS to find new best-found solutions that it might otherwise simply ignore due to penalties imposed on those solution. We recorded several statistics about the quality of solutions during the search (including the average cost local minima visited, the average cost of solutions visited) and only the statistics on the quality of new best-found solutions varied between GLS with and without aspiration. This also suggested that it was precisely when and what aspiration does that is critical in it's success.

3.2 Random moves

As already mentioned, whilst trying to understand more precisely why aspiration criteria gave a performance improvement to GLS, we tried an additional scheme, whereby with probability p, we allowed GLS to make a move at random. In addition to this, a local search was tried, with random moves as the sole means of escaping from local minima and plateaus, and was found to be much inferior to the combination of GLS with random moves or indeed GLS without random moves. This shows that GLS is enhanced by random moves. We found that GLS without random moves, at low values of lambda produced a much less diverse search, than GLS with random moves, resulting in a better performance with respect to the best cost of solution of GLS with random moves. This suggests that the role of random moves is to help GLS move out of local minima, when either random moves or GLS on it's own might not be able to do so.

4 Comparison with state of the art QAP algorithms

In this section, we compare GLS against two of the best QAP inexact algorithms, Reactive Tabu Search [2] and Robust Tabu Search [28]. We allowed each algorithm a maximum of $1000 \times N$ repairs (where N is the number of variables in the problem) and 10 runs each, taking the average deviation from the best known solution in every case and taking the average result of those 10 runs. We set GLS to use a lambda coefficient of 0.6, Pr(random move) = 0.2, and allowed GLS to make aspiration moves. The parameters for reactive tabu search and robust tabu search were the standard parameters suggested in [2] and [28], although we used our own implementation (which we have found performs similarly to the original results in the papers). The results are shown in table 1 below:

Problem group	GLS	ReTabu	RoTabu	Real problem?
Chr*	1.495	1.909	1.516	No
Lip*	0.386	0.231	0.077	No
Nug*	0.004	0.009	0.002	No
Rou*	0.038	0.015	0.016	No
Scr*	0.011	0.000	0.000	No
Sko*	0.160	0.209	0.130	No
Tai*a	0.827	0.430	0.680	No
Tai*b	0.587	1.318	0.420	No
Tho*	0.146	0.221	0.141	No
Bur*	0.000	0.084	0.002	Yes
Els19	0.000	1.684	0.193	Yes
Esc*	0.015	0.710	0.000	Yes
Ste*	0.465	0.739	0.075	Yes
Tai*c	0.063	0.022	0.039	Yes
Artificial problems	0.502	0.591	0.406	
Real problems	0.080	0.535	0.020	
All problems	0.387	0.576	0.300	

Table 1. GLS verses reactive tabu search and robust tabu search: Mean percent excess from best known solution, over 10 runs, 1000 x N repairs per problem

These results show GLS gives a comparable performance to both reactive tabu search and robust tabu search overall, and in some cases (the chr* problems) outperforms one or other or both of them. In addition to this, it should be noted Guided Local Search is a general meta-heuristic. Given our understanding of the extensions, we believe they should also generalize to other problems.

5 Conclusion

In this paper, we have presented an Extended Guided Local Search algorithm and its application to the Quadratic Assignment Problem. We have shown how two simple extensions of Guided Local Search, can dramatically increase the range of parameters under which GLS performs well. We have also studied and provided evidence on why they work. Finally, we have shown that Guided Local Search with these two extensions gives comparable results to reactive tabu search and robust tabu search (two of the most famous heuristic methods for solving the Quadratic Assignment Problem), in some cases outperforming them, given the same number of iterations for each algorithm.

Bibliography

1. Amin, A.: Simulated Jumping. Appears in Annals of Operations Research, 1998.
2. Battiti, R., Tecchiolli, G.: The Reactive Tabu Search. In ORSA Journal on Computing, 6(2):126-140, 1994.
3. Bishop, C.M., Hinton, G.: Neural Networks for Pattern Recognition. Clarendon Press 1995.
4. Burkard, R.E., Karisch, S.E., and Rendl, F.: QAPLIB - A Quadratic Assignment Problem Library. In Journal of Global Optimization 10:391-403, 1997.
5. Davenport A., Tsang E.P.K., Zhu, K., Wang C.J.: GENET: A Connectionist Architecture for Solving Constraint Satisfaction Problems by Iterative Improvement. In Proceedings of AAAI, 1994, p.325-330.
6. Davenport, A.: Extensions and Evaluation of GENET in Constraint Satisfaction. PhD Thesis, Department of Computer Science, University of Essex, Colchester, UK, July, 1997.
7. Gent, I. & Walsh, T.: Towards an Understanding of Hill-climbing Procedures for SAT. In Proceedings of AAAI-93.
8. Glover, F.: Tabu search Part I. In Operations Research Society of America (ORSA) Journal on Computing, Vol. 1, 109-206, 1989.
9. Glover, F.: Tabu search Part II. In: Operations Research Society of America (ORSA) Journal on Computing, Vol. 2, 4-32, 1990.
10. Fleurent, C., Ferland, J.A.: Genetic Hybrids for the Quadratic Assignment Problem. In Pardalos, P., Wolkowicz, H., (eds.), Quadratic Assignment and Related Problems, Vol. 16, pages 173-187. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1994.
11. Kilby, P., Prosser, P., Shaw, P.: Guided Local Search for the Vehicle Routing Problem. In Proceedings of the 2nd International Conference on Metaheuristics, July 1997.
12. Jiang Y., Kautz H., Selman B.: Solving Problems with Hard and Soft Constraints Using A Stochastic Algorithm for MAX-SAT, 1st International Joint Workshop on Artificial Intelligence and Operations Research, 1995.
13. Lau, T.L., Tsang, E.P.K.: Applying a Mutation-Based Genetic Algorithm to Processor Configuration Problems. In Proceedings of 8th IEEE Conference on Tools with Artificial Intelligence (ICTAI'96), Toulouse, France, November 1996 .
14. Lau, T.L., Tsang, E.P.K.: Solving the Processor Configuration Problem with a Mutation-Based Genetic Algorithm. In International Journal on Artificial Intelligence Tools (IJAIT), World Scientific, Vol.6, No.4, 567-585, December 1997.
15. Lau, T.L., Tsang, E.P.K.: The Guided Genetic Algorithm and its application to the General Assignment Problems. In IEEE 10th International Conference on Tools with Artificial Intelligence (ICTAI'98), Taiwan, November 1998.
16. Lau, T.L., Tsang, E.P.K.: Solving Large Processor Configuration Problems with the Guided Genetic Algorithm. In IEEE 10th International Conference on Tools with Artificial Intelligence (ICTAI'98), Taiwan, November 1998.

17.Lau, T.L., Tsang, E.P.K.: Solving the Radio Link Frequency Assignment Problem with the Guided Genetic Algorithm. In Proceedings, NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace), Aalborg, Demark, October 1998, Paper 14b.
18.Lau, T.L.: Guided Genetic Algorithm. PhD Thesis, Department of Computer Science, University of Essex, 1999.
19.Lau, T.L., Tsang, E.P.K.: Guided Genetic Algorithm and its application to Radio Link Frequency Assignment Problems. To appear in Journal of Constraints.
20.Mills, P., Tsang, E.P.K.: Guided Local Search for solving SAT and Weighted MAX-SAT Problems. In Journal of Automatic Reasoning, Special Issue on Satisfiability Problems, Kluwer, Vol.24, 2000, 205-223.
21.Pardalos,P.M., Rendl,F., Wolkowicz, H.: The Quadratic Assignment Problem: A Survey of Recent Developments. In P. Pardalos and H. Wolkowicz, editors, Quadratic Assignment and Related Problems, volume 16, pages 1-42. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1994.
22.Selman, B., Kautz, H.: An Empirical Study of Greedy Local Search for Satisfiability Testing. In Proceedings AAAI-93. 1993.
23.Selman, B., Kautz, H.: Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems. In Proceedings IJCAI-93. 1993.
24.Selman, B., Kautz, H., Cohen, B.: Noise Strategies for Improving Local Search. In Proceedings AAAI-94. 1994.
25.Selman, B., Levesque, H. & Mitchell, D.: A New Method for Solving Hard Satisfiability Problems. In Proceedings AAAI-92. 1992.
26.Stutzle T.: MAX-MIN Ant System for Quadratic Assignment Problems. Research Report AIDA-97-04, Department of Computer Schience, Darmstadt University of Technology, Germany, 1997.
27.Taillard, E.D.: Robust Tabu Search for the Quadratic Assignment Problem. In Parallel Computing, 17:443-455, 1991.
28.Taillard, E.D.: Comparison of Iterative Searches for the Quadratic Assignment Problem. Location Science, 1994.
29.Taillard, E.D., Gambardella, L.M.: Adaptive Memories for the Quadratic Assignment Problem. Research Report, IDSIA Lugano, Switzerland, 1997.
30.Thonemann, U.W., Bolte, A.: An Improved Simulated Annealing Algorithm for the Quadratic Assignment Problem. Working paper, School of Business, Department of Production and Operations Research, University of Paderborn, Germany, 1994.
31.Tsang, E.P.K.: Foundations of Constraint Satisfaction, Academic Press, London, 1993.
32.Tsang, E.P.K., Wang, C.J.: A Generic Neural Network Approach for Constraint Satisfaction Problems. In Taylor, J.G. (ed.), Neural network applications, Springer-Verlag, p.12-22, 1992.
33.Tsang, E.P.K., Voudouris, C.: Fast Local Search and Guided Local Search and their application to British Telecom's Workforce Scheduling Problem. In Operations Research Letters, Elsevier Science Publishers, Amsterdam, Vol.20, No.3, 119-127, March 1997.
34.Tsang, E.P.K., Wang, C.J., Davenport, A., Voudouris, C., Lau,T.L.: A Family of Stochastic Methods for Constraint Satisfaction and Optimization. In The First International Conference on the Practical Application of Constraint Technologies and Logic Programming (PACLP), London, 359-383, April 1999.
35.Voudouris, C.: Guided Local Search for Combinatorial Optimisation Problems. Ph.D. thesis, Department of Computer Science, University of Essex. 1997.
36.Voudouris, C., Tsang, E.P.K.: The Tunneling Algorithm for Partial CSPs and Combinatorial Optimisation Problems. Technical Report CSM-213, University of Essex, Colchester, UK, September 1994.
37.Voudouris, C., Tsang E.P.K.: Function Optimisation using Guided Local Search. Technical Report CSM-249, University of Essex, Colchester, UK, September 1995.
38.Voudouris, C., Tsang, E.P.K.: Partial Constraint Satisfaction Problems and Guided Local Search. In Proceedings of Practical Application of Constraint Technology (PACT'96), London, 337-356, April 1996.
39.Voudouris, C., Tsang, E.P.K.: Guided Local Search joins the Elite in Discrete Optimisation. In Proceedings, DIMACS Workshop on Constraint Programming and Large Scale Discrete Optimisation, Rutgers, New Jersey, USA, September 1998.

40. Voudouris, C., Tsang, E.P.K.: Solving the Radio Link Frequency Assignment Problem using Guided Local Search. In Proceedings of NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace), Aalborg, Demark, October 1998, Paper 14a.
41. Voudouris, C.: Guided Local Search - An Illustrative Example in Function Optimisation. In BT Technology Journal, Vol.16, No.3, July 1998, 46-50.
42. Voudouris, C., Tsang, E.P.K.: Guided Local Search and its application to the Travelling Salesman Problem. In European Journal of Operational Research, Anbar Publishing, Vol.113, Issue 2, March 1999, 469-499.
43. Wilhelm, M.R., Ward, T.L.: Solving Quadratic Assignment Problems by Simulated Annealing. In IIE Transaction, 19/1:107-119, 1987.