A Hybrid Genetic Algorithm for the Quadratic Assignment Problem

Manuel V*á*zquez
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
vazquez@cs.colostate.edu

L. Darrell Whitley
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
whitley@cs.colostate.edu

Abstract

A heuristic technique that combines a genetic algorithm with a Tabu Search algorithm is applied to the Quadratic Assignment Problem (QAP). The hybrid algorithm improves the results obtained through the application of each of these algorithms separately. The QAP is a NP-hard problem and instances of size $n > 15$ are still considered intractable. The results of our experiments suggest that CHC combined with TS (CHC+TS), and a TS with elitist backtracking algorithm are able to obtain good near optimal solutions within 0.75% of the best-known solutions. CHC+TS produces the best-known solution in 12 of the 16 QAPLIB problems tested, where n ranges from 10 to 256.

1 INTRODUCTION

The QAP is a Combinatorial Optimization Problem (COP) introduced by Koopmans and Beckmann [1957] to model a plant location problem. Cela [Cela, 1998] gives an overview of the QAP theory and current state of the art algorithms. Cela indicates three reasons why QAP remains an active research area. First, a large number of real-world problems are modeled by QAP. Second, a number of other well-known COP's can be formulated as QAP. Third, QAP still remains a very difficult problem from a computational point of view. Only small QAP instances can be solved to optimality. Exact techniques such as Branch and Bound [Pardalos, et al., 1997], [Ramakrishnan, et al., 1995] fail to find the solution for larger problems.

Since practical applications require solving larger problems, heuristic approaches for QAP have been developed. We developed a hybrid Genetic Algorithm to solve QAP that produces good near optimal solutions.

This paper is organized as follows: Section 2 describes the QAP. In Section 3 the previous heuristic approaches used for solving the QAP are summarized. Section 4 overviews the heuristic techniques that we implemented. Section 5 describes previous algorithm comparisons in solving QAP instances. In Section 6, we present our results. Section 7 concludes our work.

2 QUADRATIC ASSIGNMENT PROBLEM (QAP)

The QAP arises in many real-world applications, such as VLSI module placement, design of factories, scheduling, manufacturing, statistical data analysis and process-processor mapping in parallel computing. An overview of QAP applications is given by [Pardalos, et al., 1994].

In the QAP n facilities have to be assigned to n locations at minimum cost. Give a set $\Pi(n)$ of all possible permutations of $\{1, 2, ..., n\}$ and two nxn matrices $A(a_{i,j})$,$B(b_{k,l})$, the goal is to minimize the equation

$$C(\Pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b_{\Pi(i),\Pi(j)} \qquad (1)$$

A can be interpreted as a distance matrix, $(a_{i,j})$ is the distance between location i and location j. B can be interpreted as a flow matrix, where $(b_{k,l})$ is the flow of materials from facility k to facility l. The term *quadratic* comes from the reformulation of the problem as an optimization problem with a quadratic objective function.

Many COPs can be formulated as a QAP. For the Travelling Salesmen Problem (TSP) let $x_{i,j}$ denote that city i is on position j of the tour; then the matrix A is given by the distances among cities and the matrix B defines a valid tour. Other problems which can be formulated as QAPs include the maximum clique problem and the graph-partitioning problem. According to Cela [Cela, 1998], the QAP is not only NP-hard and hard to approximate, but it is also practically intractable. A QAP of size n has $n!$ different placements.

3 HEURISTICS APPROACHES FOR QAP

Current exact optimization techniques cannot solve large QAPs. For this reason, several heuristic techniques have been proposed to solve the QAP. In Cela [Cela, 1998], seven different heuristic approaches are detailed. This section will cover four of these methods: Tabu Search(TS), Simulated Annealing (SA), Genetic Algorithms (GAs) and Greedy Randomize Search Procedure (GRASP). With the exception of GRASP, we implemented all of these methods.

3.1 TABU SEARCH (TS)

The general strategy of TS, as described by Glover and his colleagues [Glover, et al., 1993] is to explore all possible moves from the current to a neighboring solution. The move leading to the best neighboring solution is accepted, even if this move results in a deterioration of the objective function. A tabu list storing inverse moves is maintained to avoid cycling during the search process. Neighborhood moves in the tabu list are avoided. However, if a move on the tabu list results in the achievement of some aspiration criterion, the move may then be allowed. This can occur, for example, if the tabu move leads to a new best solution. Hertz, Taillard and de Werra described the basic ideas of TS through its application to five COPs [Hertz, et. al., 1997], including the *robust* TS implementation for the QAP.

In the robust TS implementation [Taillard, 1991], the neighborhood is defined by the 2-opt heuristic. A move is declared tabu if it places two interchanged facilities at locations they have already occupied in the last l iterations; thus, l is the length of the tabu list. The variable l is changed frequently by choosing a random value in the range $(\lfloor 0.9n \rfloor, \lceil 1.1n \rceil)$, where n is the size of the problem. The main advantage of robust TS is that it does not require tuning parameters and produces good results on the QAPLIB problems published in [Burkard, et al., 1991]. Robust TS is simple to implement.

Battiti and Tecchiolli [Battiti, et al., 1994b] proposed the *reactive TS* strategy. Reactive TS includes a mechanism for adapting the length of the tabu list according to the problem structure. Numerical results on QAPLIB problems presented by Battiti and Tecchiolli show that in the majority of the cases the reactive TS converges to the best-known solution faster than any other known tabu search scheme.

3.2 SIMULATED ANNEALING (SA)

The simulated annealing technique [Aarts, et al., 1997] belongs to a class of local search algorithms that are known as threshold algorithms. A new solution is generated and compared against the current solution. The new solution is accepted as the current solution if the difference in quality does not exceed a dynamically selected threshold. The principle is that solutions corresponding to larger increases in cost have a small probability of being accepted. A parameter that regulates the threshold is called temperature and the function that determines the values for the temperature over time is called the cooling scheduling. The idea is to decrease the temperature over time to diminish the probability of nonimproving moves; but some nonimproving moves are necessary to escape from local optima. In our experiments, the SA technique described by Connoly [Connoly, 1990] was implemented. In Connoly's QAP implementation, the neighborhood is defined by the 2-opt heuristic.

3.3 GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE (GRASP)

GRASP was introduced by Feo and Resende [Feo, et al., 1995][Resende, 1999] as a heuristic approach to solve hard COPs. GRASP is a multi-start technique that combines greedy elements with random local search elements in a two phase meta-strategy. It consists of a construction phase (greedy) where a feasible solution is built and a local improvement phase that finds a local optima.

The implementation of GRASP for QAP described in [Li, et al., 1994] yields best-known solutions for most of the small QAPLIB instances. In their implementation, the construction phase uses a randomized greedy function to assign facilities to locations one-by-one, at each step minimizing the total cost with respect to the assignments already done. They sorted the distance matrix entries and the flow matrix entries in ascending and descending order respectively. Only the first entries (determined by a parameter β) are used to computed the flow-distance product. These products are sorted in increasing order and only the first entries (determined by another parameter α) are kept to form a candidate list. Two entries are randomly selected from the candidate list and form the first two assignations of the QAP. The rest of the facilities are assigned to locations; one facility is assigned to one location at each step according to the greedy function described above.

3.4 GENETIC ALGORITHMS (GAs)

GAs are probabilistic search algorithms based on biological principles of natural selection, recombination, mutation and survival of the fittest. During each iteration of the algorithm, the selection phase determine the survival individuals. A mating selection process chooses the parents to reproduce and the individuals to cross and to mutate. There can also be a policy to decides which individuals out of the parent and offspring generation will survive to reproduce in the next generation. A fitness function is applied to the individuals to determine the probability of survival.

Several GA approaches have been used in solving the QAP. As indicated by Cela [Cela, 1998], pure GAs show some drawbacks; GAs have been improved by combining them with local search techniques. Merz and Freisleben [Merz, et al., 1997], [Merz, et al., 1999] developed a Genetic Local Search (GLS) strategy by applying a variant of the 2-opt heuristic as a local search technique. A next descent search is used. Merz and Freisleben claim that their algorithm is superior to TS and to Ant Systems (AS), in terms of the quality of solution found within a given time limit. Fleurent and Ferland [Fleurent, et al., 1994] combined a GA with Taillard's robust TS; the algorithm is known as Genetic Hybrids (GH). They improved the best solutions known at that time for most of the large scale QAPLIB problems. However the computation time for these improvements is very large (almost 24 hours for $n = 100$). Fleurent and Ferland proposed the use of two heuristics: one (H_1) for improving the individuals of the initial population and the other (H_2) for improving the offsprings. For improving the search, Fleurent and Ferland defined an entropy measure for the QAP and give several suggestions to guide the search process: initial populations with high entropy, H_1 must be more powerful than H_2, etc. Their best results were obtained using robust TS for n^2 iterations as H_1 and applying robust TS for $10n$ iterations as H_2.

Ahuja, Orlin and Tiwari [Ahuja, et al., 1995] obtained very promising results on large scale QAPs using a greedy genetic algorithm. The greedy GA combines techniques from GRASP algorithms and GAs. The ideas they incorporated in the greedy GA include 1) the initial population is generated using a random construction heuristic, 2) use of path crossover operators (a modified version of the path relinking operator described in [Glover, 1994]), 3) use of a special immigration scheme that promotes diversity 4) periodic local hillclimbing of a subset of the population and 5) tournaments among subpopulations. The greedy GA was tested on all QAPLIB problems, obtaining the best-known solutions for 103 out of 132 test problems. In the remaining cases the greedy GA solutions are within 1% of the best-known solutions.

4 HEURISTICS IMPLEMENTED

The neighborhood operator is the 2-opt heuristic. In the QAP, the neighborhood is defined as the set of all permutations that can be reached by swapping any two elements from the current solution. Taillard indicated in his TS implementation [Taillard, 1991] that although the neighborhood size is $O(n^2)$, the change in the cost (*delta*) produced by a movement can be calculated in linear time.

4.1 TABU SEARCH

We implement Taillard's Robust TS (RoTS) technique [Taillard, 1991] described in the previous section. In order to improve robust TS we devise a backtracking mechanism inspired by the Nowicki and Smutnicki results for Job Shop Scheduling Problems [Nowicki, et al., 1996]. The backtracking technique stores a fixed number of the best solutions found during the execution of the algorithm and backtracks to these solutions when a fixed maximum number of iterations is reached. While exploring a backtracked solution, robust TS is executed for 1/5 of the maximum number of iterations.

4.2 SIMULATED ANNEALING

Connoly's [1991] SA algorithm was implemented. First, the standard local swap based local search is executed a number fixed of times (Connoly suggested 10% of the total number of iterations) to determine parameters δ_{min} and δ_{max}. Let *delta* represents the change in the cost when a swap is executed. δ_{min} represents the minimum *delta* for those swaps that increase the cost, and δ_{max} represents the maximum *delta*. Second, the temperature schedule is produced according to the following relationships:

$$t_0 = \delta_{min} + (\delta_{max} - \delta_{min})/10$$

$$t_\infty = \delta_{min}$$

$$\beta = (t_0 - t_\infty)/(n_i)\ t_0 t_\infty$$

where n_i is the number of iterations, and

$$t_i = t_{i-1}/(1 + \beta t_{i-1}).$$

4.3 GENETIC ALGORITHMS

We used three main variants of Genetic Algorithms: 1) the CHC algorithm, with local search, modified for a permutation representation, 2) the genetic local search algorithm described in [Merz, et al., 1997], 3) order-based GA, based on Davis' order-based uniform crossover operator and scramble mutation operator [Davis, 1991] in the permutation space. All the individuals produced by our implementations are improved using the 2-opt heuristic under a steepest descent method. ([Merz, et al., 1997] use next descent).

4.3.1 CHC Description

CHC [Eshelman, 1991] is a generational genetic search algorithm that uses truncation selection; the best individuals from the combined parents and offsprings populations are preserved. CHC uses a variant of Uniform Crossover where exactly half of the different alleles are interchanged (HUX). Offsprings produced by this operator are maximally different from their parents. Parents are mated in a random fashion, but only those individual pairs that exceed a mating threshold are allowed to reproduce (incest prevention strategy). In CHC, regular mutation is not used. But whenever convergence is detected, the search is restarted by reinitializing the population by mutating the current best individual (e.g., 35% mutation) to regenerate the entire population. Originally, CHC was designed to work with binary representations, but it can be extended to permutation representations. Eshelman [1991] used a permutation representation with CHC for the Traveling Salesman Problem. Extensions to CHC in our implementation include the use of the permutation distance metric (for Incest Prevention) and the Distance Preserve Crossover (DPX) operator described in [Freisleben, et al., 1996].

4.3.2 Order-Based GAs

GAs using order-based representations have been applied successfully for problems such as scheduling, where the objective is to determine the order in which jobs are released by the schedule. The main principle behind order-based operators is that they preserve the relative order present in parents. We implemented Davis' order-based operators (uniform crossover and scramble mutation) because they shown best results for Job Shop Scheduling problems [Vázquez, et al., 2000].

```
    Parent 0:   8 6 4 2 1 5 9 3  7 10
    Parent 1:   2 3 4 6 7 1 5 9 10  8
    Template:   0 1 0 1 1 0 0 0  0  1

Offspring 1a:   _ 3 _ 6 7 _ _ _  _  8
     Missing:   4   2     1 5 9 10
 Offspring 1:   4 3 2 6 7 1 5 9 10  8

Offspring 0a:   8 _ 4 _ _ 5 9 3  7  _
     Missing:     2   6 1          10
 Offspring 0:   8 2 4 6 1 5 9 3  7 10
```

Figure 1: Davis' Uniform Order-Based Crossover. Assign "B" either 0 or 1. Let $\bar{B}$ denote the complement of B. Offspring Ba is constructed by selecting elements corresponding to B positions in the template from Parent B. The order of the missing elements for Offspring B are taken from Parent $\bar{B}$.

a) Uniform Order-based Crossover : A number of elements are selected from one parent and copied to the offspring. The missing elements are taken from the other parent in order. This operator is shown in Figure 1.

b) Order-based Scramble Mutation : A sub-list of elements is selected from the parent by selecting position x and y. The sublist of elements between positions x and y is randomly permuted.

4.3.3 Implementation details

The natural encoding for the QAP is the permutation of n integers where the i^{th} element in the permutation $\pi(i)$ denotes that the facility $\pi(i)$ is located on location i.

We tested two schemes to generate initial populations, 1) random permutation, and 2) the construction technique used in GRASP [Li, et al., 1994] [Ahuja, et al., 1995]. We only publish the results obtained by using the GRASP construction technique, which proved to be the best of the two methods.

The order-based and genetic local search implementations considered two selection mechanisms. A fitness biased scheme selection in a generational GA and random selection for reproduction in a steady state strategy, similar to GENITOR [Whitley, et al., 1988]. In our generational GA, mating individuals are selected in a fitness biased fashion (rank proportional). After genetic operators are applied, the best individuals (from the set including the old population plus the new offsprings) are selected for survival. Our steady state implementation picks two parents randomly and recombines them to create two offspring. The best offspring will replace the worst individual of the pop-

Table 1: Summary of the Genetic Algorithms implemented. SS-OB (Steady-State Order-Based), SS-DPX (GLS Steady-State using DPX), GEN-OB (Generational Order-Based), GEN-DPX (Generational GLS using DPX)

ALGORITHM	Cross. Oper	pc	Mut. Oper	pm	pop size	Rep.Sel	Surv.Sel
CHC	HUX	—	Rand. Perm.	.35	50	copy	elitist
SS-OB	OBC	—	OBM	.1	100	random	elitist
SS-DPX	DPX	—	Rand. Perm.	.1	100	random	elitist
GEN-OB	OBC	.7	OBM	.1	100	fit. biased	elitist
GEN-DPX	DPX	.7	Rand. Perm.	.1	100	fit. biased	elitist

ulation.

Three crossover operators were implemented: OBC, HUX and DPX. We modify HUX for the permutation space in the following way: the similar positions are repeated so that facilities are assigned to same locations, and half of the different positions are preserved. For the remaining positions a random permutation is generated over the missing elements. In the DPX (Distance Preserve Crossover) only the similar positions are preserved and the rest are generated as a random permutation.

Order-based mutation (OBM) is used with the generational and steady-state algorithm. The restart mechanism of CHC preserves 35% of the positions and the rest are changed through random permutation.

Table 1 summarizes the characteristics of the GAs implementations.

4.4 HYBRID GENETIC ALGORITHMS

Borrowing the ideas described by Fleurent and Ferland [Fleurent, et al., 1994] a hybridization between CHC and robust TS was created. CHC is used because it produced the best GA results during our experimentation phase (see Results Section). CHC without hybridization restarts the population five times, the hybrid technique restarts the population only three times and then runs robust TS for $100n$ iterations (n is the problem size).

5 PREVIOUS ALGORITHM COMPARISON

Taillard [Taillard, 1995] compared robust TS, reactive TS and genetic hybrids algorithms. Taillard concluded that the efficiency of these methods strongly depends on the problem type and that no one method is better than all the others.

Battiti and Techiolli [Battiti, et al., 1994a] compared SA with TS techniques. Their empirical research suggests that SA could achieve a reasonable solution quality with fewer evaluations than TS, but they showed that these conclusions must be changed if the task is hard or a very good approximation of the optimal solution is desired.

Miagkikh and Punch [Miagkikh, et al., 1999] compared the Merz and Freisleben GLS algorithm, the Ant System (AS) by Maniezzo and Colorni [Maniezzo, et al., 1998] and their population of Reinforcement Learning (RL) agents. They concluded that RL outperforms AS and it is competitive to GLS (RL is better in some benchmarks problems and worse in others).

In the following section, we present our results. Other results were taken from the literature. We implemented several of these algorithms. We were able to replicate Robust Tabu Search and Simulating Annealing results, but we were not able to replicate the Global Local Search (GLS) results. (When replicating results, we used the same parameters and mechanisms described by the authors of the algorithms and not those described in Table 1). We compare our CHC+TS algorithm against published results for global local search, robust TS, reactive TS, GRASP and Genetic-Hybrids.

6 RESULTS

There exist two sets of problems in the QAPLIB that represent a challenge for the best heuristics available. These problems were introduced by Skorin-Kapov [Skorin-Kapov 1990] and Taillard [Taillard 1991]. We selected 10 problems from Taillard and 6 problems from Skorin-Kapov. Skorin-Kapov problems were selected due to their large size. Taillard problems were used mainly because their difficulty and their wide range of sizes. In Table 2, the best solution produced by each algorithm over 20 different trials is shown. Table 3 shows the deviation in percentage from the best-known solution.

We tested CHC with three different recombination operators: HUX, OB and DPX. HUX found better results in 10 of the 16 problems tested, OB and DPX found 3 each. The hybrid CHC+TS algorithm only uses the HUX recombination operator.

We also tested two different versions for each of the GAs, steady-state and generational. In the order-based GA, the steady-state strategy always produced

Table 2: Best solutions obtained for the algorithms implemented over 20 different trials. The first column contain the QAP instance, the size of the problem is implicit in the name (example tai60a is a problem with $n = 60$). The second column contains the Best-Known Solution (BKS). Bold results indicate best solutions found during the experimentation (in several cases a match with the BKS is reached, an ***** indicates the match)

Problem	BKS	RoTS	RoTS+back	SA	CHC	GLS	OB	CHC+TS
tai10a	135028	**135028(*)**	**135028(*)**	**135028(*)**	**135028(*)**	**135028(*)**	**135028(*)**	**135028(*)**
tai20a	703482	**703482(*)**	**703482(*)**	**703482(*)**	**703482(*)**	**703482(*)**	**703482(*)**	**703482(*)**
tai30a	1818146	**1818146(*)**	**1818146(*)**	1849696	1829802	1864322	1863722	**1818146(*)**
tai40a	3139370	3150192	3145124	3212692	3168766	3188226	3215382	**3139370(*)**
tai50a	4941419	4982214	4977406	5041058	5004626	5082892	5067904	**4952270**
tai60a	7208572	7265202	7243867	7381442	7303264	7417156	7422318	**7226857**
tai80a	13557864	13610428	13599322	13852100	13695390	13869310	13814562	**13590398**
tai100a	21125314	21279646	21198564	21499478	21440312	21603028	21501554	**21217167**
tai150b	498896643	502830484	501392878	502651565	501116144	505991323	506448890	**498896643(*)**
tai256c	44759294	44893126	44787321	44814014	44793457	44899131	45023121	**44759294(*)**
sko100a	152002	152082	152050	154210	152378	152750	153090	**152002(*)**
sko100b	153890	153932	153921	154262	154068	154252	155030	**153890(*)**
sko100c	147862	147894	147887	149542	148920	148776	149948	**147862(*)**
sko100d	149576	149670	149650	151746	150096	150708	150828	**149576(*)**
sko100e	149150	149182	149617	150426	150260	149578	150598	**149150(*)**
sko100f	149036	149138	149074	150738	149370	149844	150402	**149036(*)**

Table 3: Percentage above the Best Known Solutions (for the same results shown in Table 2)

Problem	RoTS	RoTS+back	SA	CHC	GLS	OB	CHC+TS
tai10a	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
tai20a	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
tai30a	**0.000**	**0.000**	1.735	0.641	2.539	2.506	**0.000**
tai40a	0.344	0.183	2.335	0.936	1.556	2.421	**0.000**
tai50a	0.825	0.728	2.016	1.279	2.863	2.559	**0.219**
tai60a	0.785	0.489	2.398	1.313	2.893	2.965	**0.253**
tai80a	0.387	0.305	2.170	1.014	2.297	1.893	**0.239**
tai100a	0.730	0.346	1.771	1.491	2.261	1.780	**0.434**
tai150b	0.788	0.500	0.752	0.444	1.422	1.513	**0.000**
tai256c	0.299	0.062	0.122	0.076	0.312	0.589	**0.000**
sko100a	0.052	0.031	1.452	0.247	0.492	0.715	**0.000**
sko100b	0.027	0.020	0.241	0.115	0.235	0.740	**0.000**
sko100c	0.021	0.016	1.136	0.715	0.618	1.410	**0.000**
sko100d	0.062	0.049	1.450	0.347	0.756	0.837	**0.000**
sko100e	0.021	0.011	0.855	0.744	0.286	0.970	**0.000**
sko100f	0.068	0.025	1.142	0.224	0.542	0.916	**0.000**

better results and in the GLS case the steady-state strategy found better results in 12 of the 16 problem.

Table 2 and 3 show that TS and CHC produce good near optimal results for the QAP instances tested. SA, OB and GLS did not work very well. In the case of OB and GLS parameter tuning may be necessary to improve the results. Our experimentation also reflects that the Taillard QAP instances are harder than the Skorin-Kapov instances. For Skorin-Kapov instances, robust TS is able to produce results within 0.07% above the best-known solution.

To improve the TS and CHC results, we designed two mechanisms. First we added a backtracking mechanism borrowed from a Job Shop Scheduling implementation to Taillard's robust TS. The robust TS with backtracking improves the robust TS solution and produces near optimal solutions within 0.73 % above the best-known solution and consistently solves the Skorin-Kapov QAP instances (see the RoTS+back column of Table 3). Second we combined robust TS with CHC. Our CHC+TS hybrid solution outperforms all other algorithms and finds the best-known solution in 12 of the 16 cases. The worst solution is only 0.5 % above the best-known solution (see the CHC+TS column of Table 3).

In order to compare our CHC+TS hybrid with the state of the art heuristics, Table 4 shows the published results of: a)GLS algorithm [Merz, et al., 1997], b) Robust TS [Taillard, 1995], [Taillard, 1991], c) Reactive TS [Battiti, et al, 1994b], d) GRASP [Feo, et al., 1995], [Resende, et al., 1996], [Li, et al., 1994] and e) Genetic Hybrids [Fleurent, et al. 1994]. For the Taillard problems, the results of algorithm e) and algorithm c) were taken from [Taillard, 1995].

Table 4 indicates that no algorithm is absolutely better than the others. Our results suggest that our hybrid technique is very competitive and produces good near-optimal results. CHC+TS only was beaten in three cases and found the best-known solution in 12 of the 16 QAPLIB problems. The reactive TS is a good strategy in Taillard problems. Unfortunately we do not have results of its application for Skorin-Kapov

Table 4: Best solutions obtained for the algorithms compared. The first column contain the QAP instance, the size of the problem is implicit in the name (example tai60a is a problem with $n = 60$). The second column contains the Best-Known Solution (BKS). Bold results indicate best solutions for the algorithms compared (in several cases a match with the BKS is reached). GLS stands for Genetic Local Search, RoTS for Robust Tabu Search, ReTS for Reactive Tabu Search and GENHyb for Genetics Hybrids. ***** indicates a match with the BKS. n/a means that no results were published (Notice that this Table does not reflect the results or the heuristics implemented here -except for our CHC+TS-).

Problem	BKS	GLS	RoTS	ReTS	GRASP	GENHyb	CHC+TS
tai10a	135028	**135028(*)**	**135028(*)**	**135028(*)**	n/a	**135028(*)**	**153028(*)**
tai20a	703482	**703482(*)**	**703482(*)**	**703482(*)**	n/a	**703482(*)**	**703482(*)**
tai30a	1818146	n/a	**1818146(*)**	**1818146(*)**	n/a	**1818146(*)**	**1818146(*)**
tai40a	3139370	n/a	3146514	3141702	n/a	31441702	**3139370(*)**
tai50a	4941419	n/a	4951186	4948508	n/a	**4941419(*)**	4952270
tai60a	7208572	7265232	7272020	7228214	n/a	7254564	**7226857**
tai80a	13557864	n/a	13582038	**13558710**	n/a	13574084	13590398
tai100a	21125314	21334458	21245778	**21160946**	n/a	21235842	21217167
tai150b	498896643	500945216	n/a	n/a	n/a	n/a	**498896643(*)**
tai256c	44759294	44812252	n/a	n/a	n/a	n/a	**44759294(*)**
sko100a	152002	152070	n/a	n/a	152926	**152002(*)**	**152002(*)**
sko100b	153890	n/a	n/a	n/a	154688	**153890(*)**	**153890(*)**
sko100c	147862	n/a	n/a	n/a	148484	**147862(*)**	**147862(*)**
sko100d	149576	n/a	n/a	n/a	150676	**149576(*)**	**149576(*)**
sko100e	149150	n/a	n/a	n/a	150118	**149150(*)**	**149150(*)**
sko100f	149036	n/a	n/a	n/a	150118	**149036(*)**	**149036(*)**

problems (we downloaded the reactive TS code from Battiti's web page, but we could not replicate the results for Taillard problems nor obtain competitive solutions for Skorin-Kapov problems). The Genetic Hybrid algorithm also works very good in these QAPLIB problems, but their main limitation is its running time [Fleurent, et al., 1994] (although we are not reporting running times, we never take more than 5 hours to produce the results shown here).

7 CONCLUSIONS

We investigate several heuristic techniques for solving the QAP. We developed two algorithms based on TS and GAs that produce good near optimal solutions. The first one, a slight modification of the Taillard's robust TS, which incorporates a backtracking mechanism, produced results 0.75% above the best-known solutions. The second one, a combination of the CHC algorithm with Robust TS gives us the best results during our experimentation and was able to find 12 of the 16 best-known solutions. CHC+TS results are only 0.5% above the best-known solution in the worst case. The key idea of this algorithm (also applied in [Fleurent, et al., 1994]) is to use GAs to explore in parallel several regions of the search space and to use a good mechanism as TS to intensify the search around some selected regions.

Acknowledgments

Manuel Vázquez is a graduate student at Colorado State University supported by PDVSA under the Program "Becas 1998-2000". This work was also supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-97-1-0271. The U.S. Government is authorized to reproduce and distribute reprints notwithstanding any copy notation thereon.

References

[Aarts, et. al., 1997]: Emile H. L. Aarts, Jan H.M. Korst and Peter J. M. van Laarhoven. "Simulated Annealing". Local Search in Combinatorial Optimization. Edited by E.Aarts and K.Lenstra. Chapter 4, pp. 91 to 120. John Wiley and Sons Ltd.

[Ahuja, et al., 1995]: R.K.Ahuja, J.B.Orlin, and A.Tivari. "A Greedy Genetic Algorithm for the Quadratic Assignment Problem". Working paper 3826-95. Sloan School of Management, MIT, 1995.

[Battiti, et al., 1994a]: R.Battiti and G. Tecchioli. "Simulated Annealing and Tabu Search in the Long Run: a Comparison on QAP Tasks". Computer and Mathematics with Applications, 28 (6):1-8, 1994.

[Battiti, et al., 1994b]: R.Battiti and G. Tecchioli. "The Reactive Tabu Search". ORSA Journal of Computing, Vol. 6, N. 2 (1994). pp. 126-140.

[Burkard, et al., 1991]: R.E. Burkard, S.E.Karisk, F.Rendl. "QAPLIB – A Quadratic Assignment Problem Library". European Journal of Operational Research, Vol. 55, pp. 115, 119.

[Cela, 1998]: Eranda Cela. "The Quadratic Assignment Problem: Theory and Algorithms". Kluwer Academic Publishers, Dordrecht, 1998.

[Connoly, 1990]: D. T. Connoly. "An improved annealing scheme for the QAP". European Journal of Operational Research 46, 1990, pp. 93-100.

[Davis, 1991]: L.Davis. "Order-Based Genetic Algorithm and the Graph Coloring Problem". In Handbook of Genetic Algorithms, Chapter 6.

[Feo, et al., 1995]: T.A.Feo and M.C.G. Resende. “Greedy Randomized Adaptive Search Procedures”. Journal of Global Optimization, vol. 6, pp. 109-133. 1995.

[Fleurent, et al., 1994]: Charles Fleurent and Jacques A. Ferland. “Genetic Hybrids for the Quadratic Assignment Problem”. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Volume 16, 1994. Quadratic Assignment and Related Problems. Panos Pardalos and Henry Wolkowicz editors. pp. 173, 187.

[Freisleben, et al., 1996]: B. Freisleben and P. Merz. “New Genetic Local Search Operators for the Traveling Salesman Problem”, in Parallel Problem Solving from Nature - PPSB IV. pp. 890-900.

[Glover, 1994]: Fred Glover. “Genetic algorithms and scatter search: Unsuspected potential”. Statistics and Computing 4, pp. 131-140.

[Glover, et al., 1993]:F.Glover, E.Taillard, D. de Werra. “A user’s guide to Tabu Search”. Annals of Operations Research, 41, 3-28, 1993.

[Hertz, et. al., 1997]: A.Hertz, E.Taillard, D. de Werra. “Tabu Search”. Local Search in Combinatorial Optimization. Edited by E.Aarts and K.Lenstra. pp. 121 to 136. Wiley.

[Koopmans, et al., 1957]: T.C.Koopmans and M.J. Beckmann. “Assignment Problems and the Location of Economic Activities”, Econometrics, vol. 25, pp.53-76.

[Li, et al., 1994]: Y.Li, P.M.Pardalos and M.C.G. Resende. “A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem”.DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Volume 16, 1994. Quadratic Assignment and Related Problems. Panos Pardalos and Henry Wolkowicz editors. pp. 237-261.

[Maniezzo, et al., 1998]:V.Maniezzo and A.Colorni. “The Ant System Applied to the Quadratic Assignment Problem”. To appear in IEEE Transactions on Knowledge and Data Engineering.

[Merz, et al., 1999]: Peter Merz and Bernd Freisleben. “A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem”. International Congress on Evolutionary Computation (CEC’ 99), IEEE Press, pp 2063-2070, 1999.

[Merz, et al., 1997]: Peter Merz and Bernd Freisleben. “A Genetic Local Search Approach to the Quadratic Assignment Problem”. In International Conference on Genetic Algorithms (ICGA) 97. pp. 465,472.

[Miagkikh, et al., 1999]: Victor V. Miaghkikh and Willian F.Punch. “An Approach to Solving Combinatorial Optimization Problems Using a Population of Reinforcement Learning Agents”. In Genetic and Evolutionary Computation Conference (GECCO). pp. 1358-1365.

[Nowicki, et al., 1996]:E. Nowicki and C. Smutnicki. “A Fast Taboo Search Algorithm for the Job-Shop Problem”. Management Science, 42(6), pp. 797-813.

[Pardalos, et al., 1997]:P.M. Pardalos, K.G. Ramakrishnan, M.C.G. Resende, and Y.Li. “Implementation of a variance reduction-based lower bound in a branch and bound algorithm for the Quadratic Assignment Problem”. SIAM J.Optim, 7(1) pp. 280-294, Feb 97.

[Pardalos, et al., 1994]: Y.Li, P.M.Pardalos, F. Rendl and H.Wolkowiccz. “The Quadratic Assignment Problem: A Survey and Recent Developments”.DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Volume 16, 1994. Quadratic Assignment and Related Problems. Panos Pardalos and Henry Wolkowicz editors. pp. 1-41.

[Ramakrishnan, et al., 1995]:K.G. Ramakrishnan, M.C.G. Resende, and P.M. Pardalos. “A Branch and Bound Algorithm for the Quadratic Assignment Problem using a Lower Bound based on Linear Programming”. Proceedings of the state of the art in Global Optimization: Computational Methods and Applications.

[Resende, 1999]: M.C.G. Resende. “Greedy Randomized Adaptive Search Procedures (GRASP)”. To appear in Encyclopedia of Optimization, Kluwer Academic Press, 1999.

[Skorin-Kapov, 1990]: J.Skorin-Kapov. “Taboo Search applied to the quadratic assignment problem”, ORSA Journal of Computing 2 (1990), pp. 33-45.

[Taillard, 1995]:E.Taillard. “Comparison of Iterative Searches for the Quadratic Assignment Problem”. Location Science 3. 1995. pp. 87-105.

[Taillard, 1991]:E.Taillard. “Robust Taboo Search for the Quadratic Assignment Problem”. Parallel Computing 17, 1991, pp 443-455.

[V*á*zquez, et al., 2000]:Manuel V*á*zquez and L. Darrel Whitley. “A comparison of Genetic Algorithms in solving the Dynamic Job Shop Scheduling Problem”. In Genetic and Evolutionary Computation Conference, GECCO 2000.

[Whitley, et al., 1988]: D. Whitley and J. Kauth: “GENITOR: A different Genetic Algorithm”. In Proc. Rocky Mountain Conf. on Artificial Intelligence.