

Tabu Search and Hybrid Genetic Algorithms for Quadratic Assignment Problems

Zvi Drezner

*California State University, Fullerton
U.S.A.*

1. Introduction

In this chapter experience with solving quadratic assignment problems is reported. The results reported in this chapter are the best results for heuristic solutions of the quadratic assignment problem available to date and can serve as bench mark results for future researchers who propose new approaches for solving quadratic assignment problems.

The most effective method to date for solving quadratic assignment problems heuristically is the hybrid genetic algorithm. The offspring produced by the genetic algorithms are improved by tabu search before considering them for inclusion into the population. Six different tabu searches are described and are embedded in a special genetic algorithm whose merging process is the most effective for heuristically solving quadratic assignment problems.

The most successful merging process (the crossover operator) used in the genetic algorithm is described. This specific merging process exploits the special structure of quadratic assignment problems and is especially effective when the distance matrix consists of “real” distances rather than random values.

A short cut suggested by Taillard (1995) is described. This short cut reduces the time required for the evaluation of all $O(n^2)$ values of the objective function by all pair-wise exchanges of facilities from $O(n^4)$ to $O(n^2)$ (i.e. $O(1)$ per pair exchange) where n is the number of facilities.

Grey pattern problems are quadratic assignment problems with a special structure. For these problems a special merging process and a special tabu search are developed (Drezner, 2006).

Several improvement schemes for genetic algorithms (or hybrid genetic algorithms) are described and discussed. These include: gender specific genetic algorithms, distance based approach to selecting parents in genetic algorithms, a distance based rule for removing population members, and compounded genetic algorithms. These improvement schemes can help researchers who work on other problems as well to improve the performance of their genetic or hybrid genetic algorithms.

The chapter concludes with summary tables of computational experiments with various techniques. These include the best known results for 32 “pure” quadratic assignment problems and 127 grey pattern quadratic assignment problems. All pure quadratic assignment problems have between 36 and 150 facilities. Smaller problems, with a few

Source: Local Search Techniques: Focus on Tabu Search, Book edited by: Wassim Jaziri, ISBN 978-3-902613-34-9, pp. 278, October 2008, I-Tech, Vienna, Austria

exceptions, were optimally solved and thus not reported. One hundred and twenty six grey pattern problems have 256 facilities. One grey pattern problem with 64 facilities and 6 grey pattern problems with 256 facilities are optimally solved.

2. The quadratic assignment problem

The quadratic assignment problem (QAP) is considered one of the most difficult optimization problems to solve optimally. The QAP is a combinatorial optimization problem proposed by Koopmans & Beckmann (1957).

The problem is defined as follows. A set of n possible sites are given and n facilities are to be located on these sites, one facility at a site. Let c_{ij} be the cost per unit distance between facilities i and j and d_{ij} be the distance between sites i and j . The cost f to be minimized over all possible permutations, calculated for an assignment of facility i to site $p(i)$ for $i=1, \dots, n$, is:

$$f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{p(i)p(j)} \quad (1)$$

Optimal algorithms can solve relatively small problems ($n \leq 36$). Nug30, Kra30b, Tho30 were solved by Anstreicher et al. (2002); Kra30a by Hahn & Krarup (2001); Ste36a by Brixius and Anstreicher (2001); Ste36b, Ste36c by Nystrom (1999). Consequently, considerable effort has been devoted to constructing heuristic algorithms for its solution. The first heuristic algorithm proposed for this problem was CRAFT (Armour & Buffa, 1962) which is a descent heuristic. More recent algorithms use metaheuristics such as tabu search (Batiti & Tecchiolli, 1994; Skorin-Kapov, 1990; Taillard, 1991), simulated annealing (Burkard & Rendl, 1984; Wilhelm & Ward, 1987; Connoly, 1990; Misevicius, 2003), simulated jumping (Amin, 1999), genetic algorithms (Ahuja et al., 2000; Fleurent & Ferland, 1994; Tate & Smith, 1995; Drezner, 2003, 2005c), ant colony search (Gambardella et al., 1999), scatter search (Cung et al., 1997), or specially designed heuristics (Drezner, 2002; Li et al., 1994).

For a complete discussion and list of references see Burkard (1990), Cela (1998), Rendl (2002), Taillard (1995), and Drezner et al. (2005).

2.1 Grey pattern problems

Grey Pattern problems (Taillard, 1995) are a special class of quadratic assignment problems. These quadratic assignment problems have a special structure that can be exploited in the design of hybrid genetic algorithms.

The grey pattern problem (Taillard 1995) is based on a rectangle of dimensions n_1 by n_2 . A grey pattern of m black points is selected from the $n = n_1 \times n_2$ slots in the rectangle while the rest of the slots remain white. This forms a "grey pattern" of density m/n . The objective is to have a grey pattern where the black points are distributed as uniformly as possible. This objective is achieved by defining a distance between pairs of points according to some rule. For more details see Taillard (1995).

Two grey pattern problems are available at QAPLIB <http://www.seas.upenn.edu/qaplib>. These are called Tai64c and Tai256c. Tai64c is a grey pattern problem in a square of 8 by 8 slots ($n=64$) and $m=13$ black points. Tai256c is a grey pattern problem in a square of dimensions 16 by 16 ($n=256$) and $m=92$ black points. Taillard and Gambardella (1997) define 126 grey pattern problems with the same distance matrix as Tai256c for $n=256$ selecting $3 \leq m \leq 128$ black points.

The grey pattern quadratic assignment problems can be formulated in a simpler way (Taillard, 1995; Drezner, 2006). m slots out of n available slots need to be selected. d_{ij} is the distance between slots i and j . Let M of cardinality m be the subset of selected slots. The objective function, to be minimized by selection of the best subset M , is $f(M) = \sum_{i,j \in M} d_{ij}$. In

Drezner (2006) the grey pattern problem is described as a special case of a more general problem. Consider n objects such as points in the plane or nodes of a network with a given distance between every pair of points. We wish to find a cluster of m points which minimizes the total distance between all pairs of points in the cluster. This cluster can be interpreted as the "tightest" cluster of m points.

Since this quadratic assignment formulation has a special structure, it is easier to solve as pointed out by Taillard (1995). Taillard (1995), Misevicius (2003a,b, 2004, 2005), and Drezner (2006) used this special formulation rather than the general QAP formulation for its solution.

3. Tabu searches for quadratic assignment problems

Tabu search procedures were suggested by Glover (1986). For a review of tabu searches the reader is referred to Glover & Laguna (1997). The search starts as a steepest descent algorithm but continues after the steepest descent algorithm has been terminated. Unlike the steepest descent, tabu search may take upward moves in the hope that a sequence of upward moves will lead to subsequent downward moves and eventually lead to a better solution. The direction of the search is determined by the recent history of moves that are "memorized". Once a move is performed, the reverse move (i.e. moving back to the previous combination) is forbidden for some iterations called tabu tenure (hence the name tabu which can also be spelled as taboo), thus pushing the search away from previous combinations. Imagine a search on a plane with many craters. One of these craters is the deepest one, and that one is the desired solution (the global optimum). The steepest descent performs only downward moves and may land at a shallow crater (a local optimum) and not at the global one. Tabu search attempts to get out of a shallow crater in the hope of getting to a better one. Therefore, when the steepest descent algorithm terminates at a bottom of a crater, upward moves are taken in tabu search while sliding back into the same crater is disallowed with the hope of sliding into deeper craters and eventually reaching the global optimum.

In this section six different tabu searches are presented: the robust tabu (RT) proposed by Taillard (1991), the modified robust tabu (MRT) suggested by Drezner & Marcoulides (2008), the simple tabu (ST) suggested in Drezner (2003) and improved by Drezner (2008a), the concentric tabu (CT) proposed by Drezner (2002), the ring moves (RM) and all moves (AM) suggested by Drezner (2005c).

3.1 Robust Tabu (RT)

Robust tabu (RT) was introduced by Taillard (1991) and is also described in Taillard (1995). The defined tabu list commonly used is set to contain pairs of facility-site (i.e., there are n^2 possible entries in the tabu list). There is a short term and long term tabu memory.

Short Term Memory: When a facility is removed from a site, the iteration number is recorded (meaning that the facility was at that site one iteration earlier). An exchange

between two facilities is disallowed (unless the objective function is better than the best one found so far) if at least one of the facilities moves back to a site it was removed from in the last u iterations. The tabu tenure u is randomly generated in $[0.9n, 1.1n]$ every iteration.

Long Term Memory: Every iteration after t iterations (we use $t=3n^2$): if there is an exchange between two facilities such that each facility moves to a site it was never there in the last t iterations, such an exchange preempts any other exchange and is executed. The long term memory serves as a diversification of the tabu search.

3.2 Modified Robust Tabu (MRT)

The modification suggested by Drezner & Marcoulides (2008) is replacing the range of $[0.9n, 1.1n]$ for the tabu tenure in the short term memory with the expanded range of $[0.2n, 1.8n]$. This modification yielded statistically proven superior results in computational experiments.

3.3 Simple Tabu (ST)

The simple tabu was suggested in Drezner (2003) and was modified to a wide range for generating the tabu tenure in Drezner (2008a).

1. The terminal solution of the descent heuristic is defined as the current solution and the best-known solution. The number of iterations of the descent heuristic is h . Empty the tabu list.
2. The following is repeated $\max\{2h, 50\}$ times:
 - All pair-wise exchanges of facilities in the current solution are checked.
 - If a solution better than the best-known solution is found, the best improving exchange is performed, the tabu list is emptied, and the next iteration starts.
 - The tabu tenure TT is randomly generated in $[0.05n, 0.45n]$ and the tabu list consists of the facilities added to it in the last TT iterations.
 - If no exchanged solution is better than the best-known solution, the best exchange (whether improving or not) between two facilities, *both* not in the tabu list, is performed.
 - The two exchanged facilities are added to the tabu list.

3.4 Concentric Tabu (CT)

Concentric tabu search was introduced in Drezner (2002). One iteration of the concentric tabu search is very similar to the variable neighborhood search (Mladenovic & Hansen, 1997; Hansen & Mladenovic, 2001). The search is performed in "rings" around the center solution, proceeding from one ring to a larger one, and so on, until a pre-specified radius is obtained. A starting solution is selected as the center solution. Every feasible solution of the quadratic assignment problem is a permutation p of the center solution. A "distance" Δp is defined for each solution p (permutation of the center solution). The distance Δp is the number of facilities in p that are not in their center solution site. Note that $\Delta p \leq n$. The tabu list consists of solutions that are not farther than Δp from the center solution, thus forcing the search away from the center solution.

For each Δp three solutions are recorded: s_0 , s_1 , and s_2 . The solution s_0 is the best encountered solution with distance Δp . Similarly, s_1 and s_2 are the best encountered solutions with distances $\Delta p+1$, and $\Delta p+2$, respectively. The depth of the search is set to $d \leq n$.

One Iteration of CT

1. Set $\Delta p=0$. The starting solution s_0 is the center solution and the best found solution.
2. All pair exchanges of s_0 are evaluated.
3. If the exchanged solution is better than the best found solution, the best found solution is updated and the rest of the exchanges are evaluated.
4. If the distance of an exchanged solution is Δp or lower, it is in the tabu list. It is ignored and the rest of the exchanges are evaluated.
5. If its distance is $\Delta p+1$ or $\Delta p+2$, s_1 or s_2 are updated if necessary.
6. If a new best found solution is found by scanning all the exchanges of s_0 , the starting (center) solution is set to the new best found solution. Go to Step 1.
7. Otherwise, set $s_0=s_1$, $s_1=s_2$, and s_2 is emptied. Set $\Delta p=\Delta p+1$.
8. If $\Delta p=d+1$ stop the iteration. Otherwise, go to Step 2.

3.5 Two extended concentric tabu searches

Two variants of the concentric tabu search are suggested in Drezner (2005c): ring moves (RM) and all moves (AM). These extended concentric tabu searches are based on the following observations.

Consider all possible changes in Δp ($\Delta\Delta p$) when facilities $f1$ and $f2$ are exchanged. The nine possible changes are depicted in Table 1. "Back" means that the facility is moved back to its center site (decreasing Δp by 1); "Out" means that a facility is removed from its center site (increasing Δp by 1), and NC (No Change) means that the facility was neither in its center site nor moved into its center site so Δp is not changed. The combination of one "Back" and one "Out" is impossible. If facility $f1$ is moved out of its center site, facility $f2$ could not be moved into its center site because this site is the center site of facility $f1$.

$f1$	$f2$	$\Delta\Delta p$
Back	Back	-2
Out	Back	*
NC	Back	-1
Back	Out	*
Out	Out	+2
NC	Out	+1
Back	NC	-1
Out	NC	+1
NC	NC	0

* Impossible

Table 1. The nine possible moves

In Table 2 we summarize the possibilities in a different way.

$\Delta\Delta p$	Move
-2	Both Back
-1	One Back, the other NC
0	Both NC
+1	One out, the other NC
+2	Both Out

Table 2. Summary of moves

In the original concentric tabu (CT) only +1 and +2 moves are considered (unless a solution better than the center solution is found). Moves of -2, -1, and 0 are in the tabu list. This means that a facility that was taken out from its center location, cannot be moved back into its center location throughout the iteration because “Back” moves are not considered (except, of course, when a better best known solution is found). Also, every move must include a facility taken out of its center location. These restrictions are a bit “harsh”. Two possible modifications to the concentric tabu are suggested. These modifications reduce the tabu list and yet guarantee that no cycling is possible.

3.6 Ring Moves (RM)

In the RM algorithm, we consider also moves “inside the ring”. A change $\Delta p = 0$ means that the exchanged facilities did not move in or out of their center location. The solution following the exchange has the same Δp (is in the same ring).

One Iteration of RM

1. Set $\Delta p = 0$. The starting solution s_0 is the center solution and the best found solution.
2. All pair exchanges of s_0 are evaluated.
3. If the exchanged solution is better than the best found solution, the best found solution is updated and the rest of the exchanges are evaluated.
4. If the distance of an exchanged solution is lower than Δp , it is in the tabu list. It is ignored and the rest of the exchanges are evaluated.
5. If its distance is Δp , $\Delta p + 1$ or $\Delta p + 2$, s_0 , s_1 or s_2 are updated if necessary. Note that the original s_0 is still used for the rest of the pair exchanges.
6. If a new best found solution is found by scanning all the exchanges of s_0 , the starting (center) solution is set to the new best found solution. Go to Step 1.
7. Otherwise,
 - If s_0 has changed, go to Step 2.
 - If s_0 has not changed, set $s_0 = s_1$, $s_1 = s_2$ and s_2 is emptied. Set $\Delta p = \Delta p + 1$.
 - If $\Delta p = d + 1$ stop the iteration. Otherwise, go to Step 2.

The algorithm allows for exchanges between two facilities, both with “No Change” that improve the present value of the objective function. Note that when a facility was taken out from its center location, it cannot be moved back into its center location throughout the iteration. Therefore, the ring moves do not rectify this issue encountered in concentric tabu.

3.7 All Moves (AM)

In the AM algorithm the tabu list is eliminated and replaced by a different approach. A list of the best encountered solution for each $0 \leq \Delta p \leq n$ is maintained (at the beginning only the center solution is in the list). Members in the list are tested to find whether their neighbors are better than other members in the list or themselves. A list member whose neighbors were not tested yet, is flagged. Once none of the members in the list are flagged, the iteration terminates. This way the value of Δp can change up and down while in CT it can only increase and in RM it can increase or stay the same.

One Iteration of AM

1. Set $\Delta p = 0$. The best encountered solution for each $2 \leq \Delta p \leq n$ is emptied and all flags removed. The starting solution is the best encountered for $\Delta p = 0$ and is flagged.
2. The flagged solution, if there is one, with the lowest Δp is selected for pair exchanges.

3. If there are no flagged solutions left, the iteration terminates with the center solution.
4. All pair exchanges of the selected solution are evaluated and its flag removed.
5. If the exchanged solution is better than the best found solution, the best found solution is updated and the rest of the exchanges are evaluated.
6. If an exchanged solution is better than the best encountered solution for the appropriate Δp , it replaces it and flagged. (If the improvement is for the same Δp , the original selected solution is kept for the remaining pair exchanges).
7. If a new best found solution is found by scanning all the exchanges of the selected solution, the starting (center) solution is set to the new best found solution. Go to Step 1.
8. Otherwise, go to Step 2.

This variant has no tabu list. It allows moves to a ring with a smaller Δp , if they improve the best encountered value of the objective function for that Δp . This variant may allow a facility that was removed from its center location to move back to it.

3.8 The improvement procedure

Robust Tabu (RT), the modified robust tabu (MRT) and the simple tabu (ST) constitute the improvement procedure. One iteration of concentric tabu (CT), ring moves (RM), or all moves (AM) is repeated L times called "levels" in the post merging improvement procedure of the hybrid genetic algorithm. In Drezner (2003) concentric tabu with $L=3$ levels (CT3) was used.

The Improvement Procedure for CT, RM, and AM

1. The result of the merging procedure is the center solution. It is also the best found solution.
2. Set a counter $c=0$.
3. Select d randomly in $[n-4, n-2]$ (other ranges for d may be used). Perform one iteration of CT, RM, or AM on the center solution.
4. If the iteration improved the best found solution go to Step 2.
5. Otherwise, advance the counter $c=c+1$, and
 - If $c \leq L$ and is odd use the best solution with depth d as the new center solution and go to Step 3.
 - If $c \leq L$ and is even use the best solution found throughout the scan (the previous center solution is not considered) as the new center solution and go to Step 3.
 - If $c=L+1$ stop and report the best found solution.

In order to reduce run time, a shorter depth d randomly generated in $[0.3n, 0.9n]$ defined as a "short" search was suggested in Drezner (2005a). It produced improved results when the number of levels was relatively small but was inferior for larger number of levels reported in this chapter.

3.9 Selecting among equal values

In many heuristic optimization algorithms (such as tabu search), each iteration the best "move" to another solution to be employed in the next iteration is selected. It is very common that there are several moves with exactly the same value for the objective function. Which of the tying moves should be selected? If we select a move only if it is better than the best move found so far, the first tying move will always be selected. If we select a move as long as it is not worse than the best move found so far, the last one will be selected. This

may bias the search (for example, if the moves are scanned by the order of the facilities) giving a preference to either early scanned moves or late scanned moves. One possible approach is to evaluate the possible moves in random order. Another way is to save the tying moves and once the process is completed, when we know how many tying moves there are, a tying move is selected at random. Both of these approaches are cumbersome and require extra code and memory space. The following approach (Drezner, 2008b) is a simple approach which is very easy to implement.

When the moves are evaluated et seriatim, we do not know how many tying moves there will eventually be. The strategy is to select tying move number K with a probability of $1/K$. The first move which is better than the best move found so far is selected with probability 1. The second tying move is selected with probability of $1/2$, the third with probability of $1/3$ and so on. This rule is obvious for one or two tying moves (if there are 2 tying moves, each is selected with a probability of 50%). It is proven in Drezner (2008b) by mathematical induction that if eventually there are K tying moves, each of them is selected with a probability of $1/K$.

3.10 A short cut for reducing the run time in tabu searches

Taillard (1995) suggested an effective short cut for reducing the run time necessary for evaluating the values of the objective function for all pair exchanges. There are $n(n-1)/2$ possible pair exchanges and evaluating each value of the objective function requires $O(n^2)$ time leading to a total of $O(n^4)$ time. Taillard (1995) suggested the following procedure that calculates *all* these values of the objective functions in $O(n^2)$ time. Run times of the various tabu searches and the hybrid genetic algorithms were reduced by a substantial factor using this technique. This short cut can be applied for the calculations of pair exchanges in all six tabu searches described above.

Since we experimented only with symmetric problems, we present this short cut for symmetric problems with zero diagonal (i.e., the cost between a facility and itself, and the distance between the same two locations is zero). It can be easily generalized to non symmetric problems.

Let Δf_{rs} be the change in the cost f , calculated by Equation (1), by exchanging the sites of facilities r and s . This is a concept similar to the derivative of f . There are $n(n-1)/2$ such values. It can be easily verified by examining Equation (1) that:

$$\begin{aligned} \Delta f_{rs} &= 2 \sum_{i=1}^n \left\{ c_{ir} \left[d_{p(i)p(s)} - d_{p(i)p(r)} \right] + c_{is} \left[d_{p(i)p(r)} - d_{p(i)p(s)} \right] \right\} = \\ &= 2 \sum_{i=1}^n \left\{ \left[c_{ir} - c_{is} \right] \left[d_{p(i)p(r)} - d_{p(i)p(s)} \right] \right\} \end{aligned} \quad (2)$$

Calculating Δf_{rs} by using Equation (2) requires only $O(n)$ time rather than $O(n^2)$ time required to calculate f by Equation (1). Taillard (1995) points to yet a faster formula for calculating Δf_{rs} .

Let $\Delta_{uv} f_{rs}$ be the change in the value of the objective function between the exchanged permutation by uv , and an additional exchanged pair rs when rs and uv are mutually exclusive. Note that $\Delta_{uv} f_{rs} = \Delta_{rs} f_{uv}$. This is a concept similar to the second derivative of f . This change in the value of the objective function can be calculated in $O(1)$ time (starting from the second iteration) if the pairs rs and uv are mutually exclusive. The formula is based on Δf_{uv}

(the change in the value of the objective function from the previous permutation by exchanging the pair uv). Therefore, one needs to keep all the values of Δf_{ij} for all i, j . Saving these Δf_{ij} values requires $O(n^2)$ time for each evaluation of all pair exchanges of s_0 .

Since

$$\Delta f_{uv} = 2 \sum_{i=1}^n \left\{ [c_{iu} - c_{iv}] [d_{p(i)p(u)} - d_{p(i)p(v)}] \right\}$$

it can be easily verified that:

$$\Delta_{uv} f_{rs} = \Delta f_{uv} + 2 [c_{su} + c_{rv} - c_{sv} - c_{ru}] [d_{p(s)p(u)} + d_{p(r)p(v)} - d_{p(s)p(v)} - d_{p(r)p(u)}] \quad (3)$$

which is calculated in $O(1)$. Note that only $2n - 3$ pairs are not mutually exclusive and formula (2) can be used in these cases to evaluate $\Delta_{uv} f_{rs}$. Therefore, evaluating the change in the value of the objective function for all $n(n-1)/2$ possible pair exchanges (which is required for one step of each of the tabu search algorithms described above) requires $O(n^2)$ time rather than $O(n^4)$ time by calculating each f directly or $O(n^3)$ time by using only reduction scheme (2).

4. Hybrid genetic algorithms

Genetic algorithms (Holland, 1975; Goldberg, 1989) simulate evolution and survival of the fittest. A population (made of individual permutations) evolves over time (generations). Pairs of population members (permutations) mate and produce an offspring (two permutations are merged to produce a new permutation). Good offspring are kept in the population whereas unfit population members are discarded (the survival of the fittest). The population evolves and at the end of the process, the population usually consists of fairly good solutions (without a guarantee that the optimal solution is found). Hybrid genetic algorithms, sometimes called memetic algorithms (Moscato, 2002), incorporate some improvement heuristic on every offspring before considering its inclusion into the population. For a review see Drezner & Drezner (2005).

The following is a short description of the specific hybrid genetic algorithm used for the computational experiments.

1. A starting population of size P is randomly selected, and the improvement procedure is applied on each starting population member.
2. Steps 3-6 are repeated for a pre-specified number of G generations.
3. Two population members are randomly selected and merged by a crossover operator to produce an offspring.
4. The improvement procedure is applied on the merged solution, possibly improving it.
5. If the value of the objective function of the offspring is not better than the value of the objective function of the worst population member, the offspring is ignored and the process of the next generation starts.
6. Otherwise,
 - If the offspring is identical to an existing population member, it is ignored and the process of the next generation starts.

- If the offspring is different from all population members, the offspring replaces a population member selected according to one of the rules described in Section 5.4. In most variants the worst population member is removed.

There are many genetic algorithms suggested for the solution of the quadratic assignment problem (Ahuja et al., 2000; Fleurent & Ferland, 1994; Tate & Smith, 1995; Drezner, 2003, 2005c). Drezner (2003) suggested a merging process that proved to be most effective for the solution of quadratic assignment problems. There are two merging processes suggested in Drezner (2003): the cohesive merging process and the scrambled merging process. The cohesive merging process was slightly better than the scrambled merging process and was used in subsequent algorithms. We present here the cohesive merging procedure.

4.1 The cohesive merging procedure

The most crucial part of a successful genetic algorithm is the merging process of two parents to produce an offspring. For the process to be effective an offspring should be significantly better (in terms of its value of the objective function) than a randomly generated solution. Otherwise, we do not gain by the merging process. It is true that such an algorithm may find a good solution, but it does not have a significant advantage over repeating the improvement process from randomly generated solutions the same number of times. Therefore, it is essential to find a merging rule that exploits the structure of the problem and is likely to use “good features” of the parents when creating an offspring.

The cohesive merging process for the solution of the quadratic assignment problem is similar to the successful merging procedure used in Drezner & Salhi (2002) for solving a network design problem. Drezner & Wesolowsky (1997) suggested the problem of designing a network so that each link can be either a two way link or a one-way link in one direction. The model was extended in Drezner & Wesolowsky (2003) to include the option of eliminating links. The problem is to determine the design of each link to minimize an objective function. A hybrid genetic algorithm is proposed in Drezner & Salhi (2002) for the solution of the problem. Suppose that two parents are selected and an offspring is designed. The network is divided into two cohesive parts by selecting a pivot node, assigning a count of 1 to all links directly connected to it, a count of 2 to all links connected to a link of count 1, and so on. Each link gets a count. The median of these counts for all links is calculated. The design of links with a count below the median is taken from the first parent and the design of links above the median is taken from the second parent. The design of links that have a count equal to the median is randomly selected from one of the parents. This way each part of a parent is a connected part of the network. Drezner & Salhi (2002) suggest considering the n possible partitions (one for each pivot node) and selecting the best offspring of these n partitions for an improvement procedure and possible inclusion in the population. For the quadratic assignment problem a similar merging process is suggested (Drezner, 2003).

In the cohesive merging process we attempt to divide the sites into two cohesive parts where each has all its facilities from the same parent. A pair of parents is randomly selected. The parent with the better value of the objective function is selected as the first parent. If the two parents tie in the value of the objective function, one of them is arbitrarily selected as the first parent. The following is executed for every pivot site.

1. The median distance from the pivot site to all sites is calculated (this is done in the preamble and not at every iteration).

2. A site that is closer than the median to the pivot site is assigned the facility from the first parent.
 3. All other sites are assigned the facility from the second parent.
 4. It is likely that some facilities are assigned twice and some are not assigned at all. Therefore,
 - Create a list of unassigned facilities.
 - Find all facilities that are assigned twice, and replace the site that is farther than the median (i.e., from the second parent) with a facility from the list.
 - Remove the selected facility from the list.
 5. This completes the merge of the two selected parents for one pivot site.
- The merged solution with the lowest value of the objective function is the offspring selected for the improvement algorithm.

4.2 Merging processes for the grey pattern problems

Grey pattern problems (Taillard, 1995; Drezner, 2006) are special cases of the quadratic assignment problem and special merging processes designed for it are warranted. We present the descent merging process and its extension to a tabu merging process (Drezner, 2006). These merging processes do not resemble neither the standard crossover operator nor the hybrid genetic algorithm approach. They combine elements of both. The tabu merging process provided the best results.

4.3 The descent merging process for grey pattern problems

The descent merging process is similar to the merging process suggested in Berman and Drezner (2007).

1. The two parents are M_1 and M_2 , each represented by a set of m slots.
2. The intersection between M_1 and M_2 is: $M^I = M_1 \cap M_2$. The cardinality of the intersection is m^I .
3. The union of M_1 and M_2 is: $M^U = M_1 \cup M_2$. The cardinality of M^U is $2m - m^I$.
4. K different slots not in M^U are randomly selected to form M^K (if $2m - m^I + K > n$, only $n - 2m + m^I$ points are selected).
5. All the points in M^U which are not in M^I define M^E . The cardinality of M^E is $2m - 2m^I$.
6. Define $M^D = M^E \cup M^K$. The cardinality of M^D is $m^D = \min\{n - m^I, 2m - 2m^I + K\}$.
7. A starting offspring M' of cardinality m is created by randomly adding to M^I $m - m^I$ points from M^D .
8. A restricted descent process is performed on M' by adding or removing only points in M^D and keeping the points in M^I in the selected set.
9. The result of the restricted descent process is the offspring.

4.4 The tabu merging process

We also experimented with a tabu extension of the restricted descent search. Let h be the number of iterations performed by the restricted descent algorithm. A restricted tabu search for additional $5h$ iterations is performed. The value $K=3$ was used in the descent algorithm and the tabu search. The tabu tenure was randomly generated in the range $[0.02(n-m), 0.2(n-m)]$. We need to select $m - m^I$ slots out of m^D slots. If $m^D - (m - m^I) \leq 5$, the tabu search is not performed and the result of the descent algorithm is applied. For complete details see Drezner (2006).

5. Improvements of genetic algorithms

The improvements described below are described in Drezner & Drezner (2005). The compounded genetic algorithm is proposed in Drezner (2005a), the gender specific genetic algorithm is proposed in Drezner & Drezner (2006), the distance based parent selection is proposed in Drezner & Marcoulides (2003), and the modification of the removal rule of population members is proposed in Drezner (2005b). The reader is referred to these papers for a complete description of the improvements. There are many other improvements suggested by many authors. For example, mutations (Spears, 2000), invasions (Goldberg, 1989), parallel genetic algorithms (Cantu-Paz, 1998), among others.

5.1 Compounded genetic algorithms

In the compounded genetic algorithm (Drezner, 2005a) genetic algorithms are applied in two phases, generating the starting population for phase 2 by repeating genetic algorithms in phase 1. This mimics evolving parallel populations at several isolated locations. The best species in each location are moved to a common location thus creating a "high quality" starting population. Suppose that a population of P members is applied in phase 2. Genetic algorithms are run K times in phase 1 using K randomly generated starting populations (it is convenient but not necessary to have integer P/K). The population size of the phase 1 genetic algorithm should have at least P/K members. The best P/K population members of each run are compiled to construct the starting population for phase 2. Phase 2 genetic algorithm is run once. It is recommended that a "quick" genetic algorithm is used for phase 1 and an "effective" and possibly "slow" genetic algorithm is used for phase 2.

For example, if a population of $P=100$ members is required for phase 2, phase 1 can be run $K=20$ times (each with a population of at least 5 members), the best $P/K=5$ population members are selected from each run and compiled to create a starting population for phase 2.

Note that the best solution found in phase 1 by any of the runs can only be improved by the compounded genetic algorithm because the best solution found during phase 1 is a member of the starting population of phase 2 and can only be removed from the population by better solutions.

5.2 Gender specific genetic algorithms

In nature, most advanced species require two genders in order to mate and reproduce. The gender modification attempts to mimic this natural process. One can argue that the division into two genders was selected over time as the preferred way for producing offspring and is therefore superior to other possible mating schemes. Epelman et al. (2005) show that having only two genders maximizes long run viability. Their finding is not directly related to genetic algorithms. However, it supports our experience that it is true for genetic algorithms as well. In gender-specific genetic algorithms the diversity of the population is better maintained with no detrimental effects on run time.

It is easy to "convert" a given genetic algorithm to a gender-specific one. Three minor modifications are required (Drezner & Drezner, 2006).

1. When the starting population is generated, half the population members are designated as males and half are designated as females. The assignment of gender is done at random and no characteristic of the population member is used for such determination.

2. When selecting two parents, the first parent is randomly selected while the second is randomly selected from the pool of the opposite gender.
3. When an offspring is generated, it is randomly assigned a gender with a 50% probability of being assigned a male gender and 50% probability a female. Again, no characteristic of the offspring should be used to determine its gender.

No extra effort is required for the implementation of the gender-specific modification. A vector of genders for population members needs to be maintained, along with the gender determined for each offspring. In Drezner & Drezner (2006) it has been statistically shown that the gender-specific algorithm significantly improves the solutions on four sets of optimization problems.

Note that it is important that an offspring's gender is randomly determined. An early attempt (Allenson, 1992) for such a modification failed because it was suggested that the offspring is assigned the gender of the discarded population member. The rationale for this rule is to keep the population half males and half females. However, such a rule is inconsistent with nature. The concern is that if the population becomes all males or all females no further evolution is possible. The evolutionary process must be terminated prematurely if such a population structure evolves. In Drezner & Drezner (2006) it is shown that for a sufficiently large population (50 or more members), the probability that all population members will have the same gender is extremely low and such an event can be ignored.

5.3 Distance based parent selection

All human cultures prohibit marriage between siblings or between parents and children (genetically similar pairs). In societies where marriages are arranged, similarity in socio-economic standing, but not genetic make-up, is prevalent. Some plants avoid pollination from genetically similar or identical individuals because self-pollination or pollination by 'siblings' is typically unsuccessful, a phenomenon referred to in biology as "inbreeding depression". Mating between close relatives often results in less fit offspring. Another, less well known biological fact, is that mating between genetically distant members of the same species can lead to a decline in offspring fitness, a condition known as "outbreeding depression" or "hybrid breakdown". Some species avoid pollination from individuals that are geographically distant or genetically dissimilar, as offspring may be less suited to the local conditions and may be poorer competitors locally. Edmands (1999, 2002) observed that parental divergence (parents who are genetically distant) leads to less fit offspring.

In genetic algorithms, if dissimilar individuals mate, the offspring is more genetically diverse which is critical in maintaining a population's genetic diversity. However, parents who are too dissimilar produce less fit offspring. Using the distance criterion for parent selection Drezner & Marcoulides (2003) crafted a rule attempting to find dissimilar but not too distant parents. A parameter $K=1,2,3,\dots$ is used. The first parent is randomly selected and K candidates for mating are then randomly selected. The distance (number of variables with different values, the Hamming distance metric) between the first parent and all candidate mates is calculated. The farthest mate among these K candidates is selected as the second parent. Note that $K=1$ is the "standard" parent selection. Drezner & Marcoulides (2003) found that the efficiency of the modification for a set of test problems peaks for $K=2$,

3. Selecting the farthest population member as a mate does not work well. Two dissimilar parents may produce an offspring which is not improved compared with a randomly generated solution. It was also found that run time increases with K which further reduces the appeal of larger values of K .

5.4 Removal of population members

In most standard genetic algorithms, when an offspring is generated, it is compared with the worst population member and if the offspring is better than the worst population member, it replaces it. Some genetic algorithms employ a rule according to which if the offspring is identical to an existing population member it is not considered for inclusion in the population. This precludes the possibility of having two identical population members. Drezner (2005b) suggested a different rule for removal of population members. Two rules for removal of a population member, once a better offspring (who is not identical to an existing population member) is found, are used. Rule 1 is the standard approach and Rule 2 is a new one.

Rule 1: Remove the worst population member.

Rule 2: Hamming distances between all pairs of population members are calculated. Suppose that the shortest distance among all pairs of population members is d . All existing population members who are at distance d from another population member form a subset. This subset must have at least two members (at least one pair of population members are at distance d from one another). Remove the worst population member in this subset.

In the experiments performed in Drezner (2005b) it was found that Rule 2 is not necessarily better than Rule 1. The suggested rule is to select Rule 2 with probability p , and to select Rule 1 otherwise. Note that $p=0$ is the standard rule (Rule 1), and $p=1$ is Rule 2. The mix between the two rules by selecting $0 < p < 1$, seems to work well.

6. Computational results

All the results reported in this chapter are based on programs coded in Fortran, compiled by Intel 9.0 Fortran compiler and ran on a 2.8GHz Pentium IV desktop computer with 256MB of RAM.

6.1 Results for the first set of problems

In Table 3 we report the results obtained in Drezner (2008a) for 18 problems available in the QAPLIB. These problems are: Ste36a, Ste36c (Steinberg, 1961), Tho40 (Thonemann & Bolte, 1994), Sko49 (Skorin-Kapov, 1990), Wil50 (Wilhelm & Ward, 1987), Sko56, Sko64, Sko72, Sko81, Sko90, Sko100a-f (Skorin-Kapov, 1990), Wil100 (Wilhelm & Ward, 1987), and Tho150 (Thonemann & Bolte, 1994).

Each problem was solved twenty times. The results for the first 17 problems are by the hybrid genetic algorithm using the modified robust tabu search with 60 levels (MRT60). The results for the last problem (Tho150) are by a simple tabu hybrid genetic algorithm (Drezner, 2008a) with 100 levels. When MRT60 was applied to Tho150, the best known solution was found 6 times out of 20 runs with the average solution of 0.002% over the best known solution requiring a run time of 1223.57 minutes.

#	Problem	n	Best Known	(1)	(2)	Time (min.)
1	Ste36a	36	9,526	20	0.000%	1.55
2	Ste36c	36	8,239.11	20	0.000%	1.55
3	Tho40	40	240,516	20	0.000%	2.12
4	Sko49	49	23,386	20	0.000%	4.27
5	Wil50	50	48,816	20	0.000%	4.55
6	Sko56	56	34,458	20	0.000%	7.15
7	Sko64	64	48,498	20	0.000%	12.41
8	Sko72	72	66,256	20	0.000%	19.85
9	Sko81	81	90,998	20	0.000%	31.94
10	Sko90	90	115,534	20	0.000%	48.46
11	Sko100a	100	152,002	20	0.000%	73.57
12	Sko100b	100	153,890	20	0.000%	73.47
13	Sko100c	100	147,862	20	0.000%	73.46
14	Sko100d	100	149,576	20	0.000%	73.50
15	Sko100e	100	149,150	20	0.000%	73.47
16	Sko100f	100	149,036	18	0.001%	73.48
17	Wil100	100	273,038	20	0.000%	73.57
18	Tho150	150	8,133,398	17	0.000%	1949.05

(1) Number of times (out of 20 runs) that best known solution found

(2) Percent of average solution above best known solution

Table 3. Results for first set of problems

6.2 Results for de Carvalho and Rahmann problems

Recently de Carvalho & Rahmann (2006) introduced a new class of quadratic assignment problems that turn out to be extremely difficult to solve. There are 14 problems in this set. Seven problems called border length minimization and seven problems called conflict index minimization. The costs of the seven conflict minimization are not symmetric but the distances are and the diagonal elements are zeroes. In order to solve these problems by our symmetric program, the costs are redefined as $c'_{ij} = c'_{ji} = c_{ij} + c_{ji}$. All these 14 problems were solved by GRASP (Oliveira Pardalos & Resende, 2004) reported in de Carvalho & Rahmann (2006), GATS (genetic algorithm and tabu search) solved by the method in Rodriguez et al., (2004), EDA (estimation of distribution algorithms) reported in Pelikan et al. (2007), and Drezner & Marcoulides (2008). All these researchers report that these problems are extremely difficult to solve among the benchmark problems available on QAPLIB. Drezner & Marcoulides (2008) obtained the best results by applying the modified robust tabu search. There are two parameters that determine the total run time required by the hybrid genetic algorithm: the number of generations and the depth of the tabu search applied in each generation. The run time of the algorithm is proportional to each of these parameters and thus proportional to their product. The "standard" number of generations (Drezner, 2003) is $\max\{20n, 1000\}$ thus we used $G \times \max\{20n, 1000\}$ generations. The depth of the tabu search (the number of iterations in the tabu search) is $D \times n$. G and D are parameters. Various values of G and D such that $G \times D = 120$ were tested to determine the trade-off between them. In

Table 4 the best known results for this set of problems are reported along with average run times reported in Drezner and Marcoulides (2008). Twelve of these results are better than the best results in all previous reports.

n	Border Length Minimization		Conflict Index Minimization	
	Best Known	Time (min.)	Best Known	Time (min.)
36	3,296	1.89	168,611,971	1.89
49	4,548	4.92	236,355,034	4.92
64	5,988	13.92	325,671,035	13.91
81	7,536	37.98	427,447,820	38.14
100	9,272	95.28	523,146,366	95.53
121	11,412	235.61	653,416,978	235.29
144	13,472	524.04	795,009,899	525.57

Table 4. de Carvalho and Rahmann problems

6.3 Results for grey pattern problems

The best known values of the 126 grey pattern problems are reported in Table 5. The original best known values are reported in Taillard and Gambardella (1997). Misevicius (2003a,b, 2004, 2005) improved some best known values. Eight additional improved best known values are reported in Drezner (2006) by using the hybrid genetic algorithm with the tabu merging process. Drezner (2006) also found all previously best known values. Average run time was 3.25 minutes per problem. In Drezner (2006) it was proven that the solution of 1,855,928 for the Tai64c problem is optimal and the six solutions to Tai256c with $m=3-8$ reported in Table 5 are also optimal.

7. Conclusions

In this chapter we report the best known results for 159 quadratic assignment problems. Thirty two of these problems are "pure" quadratic assignment problems and 127 of them are grey pattern problems which are a specific type of quadratic assignment problems. These results were obtained by hybrid genetic algorithms using tabu search as its improvement procedure. The genetic algorithm and six variants of tabu search are described and implemented for obtaining these best known solutions. A short cut for calculating the change in the value of the objective function by exchanging pairs of facilities, and an effective merging procedure for genetic or hybrid genetic algorithms are described. Special hybrid genetic algorithms that exploit the special structure of grey pattern problems are designed for solving these problems.

We also describe four improvements to genetic or hybrid genetic algorithms, and an easy way to randomly select a solution among equally valued options. We also observe that increasing the range from which the tabu tenure is randomly selected is also beneficial. All these improvements can be used in designing tabu search, genetic or hybrid genetic algorithms, for heuristically solving any optimization problem.

m	BK	m	BK	m	BK	m	BK
3	7,810	35	4,890,132	67	21,439,396	99	52,660,116
4	15,620	36	5,222,296	68	22,234,020	100	53,838,088
5	38,072	37	5,565,236	69	23,049,732	101	55,014,262
6	63,508	38	5,909,202	70	23,852,796	102	56,202,826
7	97,178	39	6,262,248	71	24,693,608	103	57,417,112
8	131,240	40	6,613,472	72	25,529,984	104	58,625,240
9	183,744	41	7,002,794	73	26,375,828	105	59,854,744
10	242,266	42	7,390,586	74	27,235,240	106	61,084,902
11	304,722	43	7,794,422	75	28,114,952	107	62,324,634
12	368,952	44	8,217,264	76	29,000,908	108	63,582,416
13	457,504	45	8,674,910	77	29,894,452	109	64,851,966
14	547,522	46	9,129,192	78	30,797,954	110	66,120,434
15	644,036	47	9,575,736	79	31,702,182	111	67,392,724
16	742,480	48	10,016,256	80	32,593,088	112	68,666,416
17	878,888	49	10,518,838	81	33,544,628	113	69,984,758
18	1,012,990	50	11,017,342	82	34,492,592	114	71,304,194
19	1,157,992	51	11,516,840	83	35,443,938	115	72,630,764
20	1,305,744	52	12,018,388	84	36,395,172	116	73,962,220
21	1,466,210	53	12,558,226	85	37,378,800	117	75,307,424
22	1,637,794	54	13,096,646	86	38,376,438	118	76,657,014
23	1,820,052	55	13,661,614	87	39,389,054	119	78,015,914
24	2,010,846	56	14,229,492	88	40,416,536	120	79,375,832
25	2,215,714	57	14,793,682	89	41,512,742	121	80,756,852
26	2,426,298	58	15,363,628	90	42,597,626	122	82,138,768
27	2,645,436	59	15,981,086	91	43,676,474	123	83,528,554
28	2,871,704	60	16,575,644	92	44,759,294	124	84,920,540
29	3,122,510	61	17,194,812	93	45,870,244	125	86,327,812
30	3,373,854	62	17,822,806	94	46,975,856	126	87,736,646
31	3,646,344	63	18,435,790	95	48,081,112	127	89,150,166
32	3,899,744	64	19,050,432	96	49,182,368	128	90,565,248
33	4,230,950	65	19,848,790	97	50,344,050		
34	4,560,162	66	20,648,754	98	51,486,642		

Table 5: Best known values for grey pattern problems

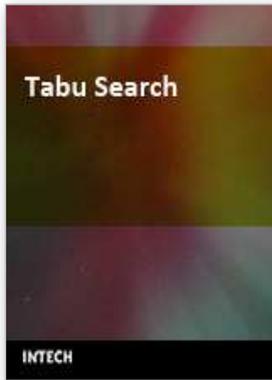
8. References

- Ahuja, R.K.; Orlin, J.B. & Tiwari, A. (2000). A Descent Genetic Algorithm for the Quadratic Assignment Problem. *Computers and Operations Research*, 27, 917-934.
- Allenson R. (1992). Genetic algorithms with gender for multi-function optimisation. Technical Report EPCC-SS92-01, Edinburgh Parallel Computing Centre, Edinburgh, Scotland, 1992.
- Amin S. (1999). Simulated Jumping. *Annals of Operations Research*, 86, 23-38.

- Anstreicher, K.; Brixius, N.; Goux, J.-P. & Linderoth, J. (2002). Solving large quadratic assignment problems on computational grids. *Mathematical Programming*, 91, 563-588.
- Armour, G.C. & Buffa, E.S. (1963). A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities. *Management Science*, 9, 294-309.
- Battiti, R. & Tecchiolli, G. (1994). The Reactive Tabu Search, *ORSA Journal on Computing*, 6, 126-140.
- Berman O. & Z. Drezner (2007). The multiple server location problem. *Journal of the Operational Research Society*, 58, 91-99.
- Brixius N.W. & K.M. Anstreicher (2001). The Steinberg wiring problem. Working Paper, The University of Iowa, October 2001.
- Burkard, R.E. (1990). Locations with Spatial Interactions: The Quadratic Assignment Problem. In *Discrete Location Theory* P.B. Mirchandani & R.L. Francis (Eds.), 387-437, Wiley, Berlin.
- Burkard, R.E. & Rendl, F. (1984). A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems. *European Journal of Operational Research*, 17, 169-174.
- Cantu-Paz, E. (1998). A Survey of Parallel Genetic Algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*, 10, 141-171.
- Cela, E. (1998). *The Quadratic Assignment Problem: Theory and Algorithms*, Kluwer Academic Publishers, Dordrecht.
- Connoly, D. T. (1990). An improved annealing scheme for the QAP. *European Journal of Operational Research*, 46, 93-100.
- Cung V.-D.; Mautor T.; Michelin P. & Tavares A. (1997). A Scatter Search based Approach for the Quadratic Assignment Problem. *Proceedings of the IEEE International Conference on Evolutionary Computation and Evolutionary Programming (ICEC'97)*, Indianapolis, 165-170.
- de Carvalho Jr., S.A. & S. Rahmann (2006). Microarray Layout as a Quadratic Assignment Problem. *Proceedings of the German Conference on Bioinformatics, Lecture Notes in Informatics*. Vol. P-83, , 11-20, D. H. Huson; O. Kohlbacher; A. Lupas; K. Nieselt & A. Zell (eds.).
- Drezner Z. (2002). Heuristic Algorithms for the Solution of the Quadratic Assignment Problem. *Journal of Applied Mathematics and Decision Sciences*, 6, 163-173.
- Drezner Z. (2003). A New Genetic Algorithm for the Quadratic Assignment Problem. *INFORMS Journal on Computing*, 15, 320-330.
- Drezner, Z. (2005a). Compounded Genetic Algorithms. *Operations Research Letters*, 33, 475-480.
- Drezner Z. (2005b). A Distance Based Rule for Removing Population Members in Genetic Algorithms, *4OR*, 3, 109-116.
- Drezner Z. (2005c). The Extended Concentric Tabu for the Quadratic Assignment Problem. *European Journal of Operational Research*. 160, 416-422.
- Drezner Z. (2008a). Extensive Experiments with Hybrid Genetic Algorithms for the Solution of the Quadratic Assignment Problem. *Computers and Operations Research*, 35, 717-736.
- Drezner Z. (2008b). Random Selection from a Stream of Events. *Communications of the ACM*. In press.
- Drezner T. & Z. Drezner (2005). Genetic Algorithms: Mimicking Evolution and Natural Selection in Optimization Models. in *Biomimetics - Biologically Inspired Technologies*, Y. Bar-Cohen (Ed.), CRC Press, Boca Raton, FL, 157-175.

- Drezner T. & Z. Drezner (2006). Gender-Specific Genetic Algorithms. *INFOR - Information Systems and Operations Research*, 44, 117-127.
- Drezner Z.; P.M. Hahn & E.D. Taillard (2005). Recent Advances for the Quadratic Assignment Problem with Special Emphasis on Instances that are Difficult for Meta-heuristic Methods. *Annals of Operations Research*, 139, 65-94.
- Drezner Z. & G.A. Marcoulides (2003). A Distance-Based Selection of Parents in Genetic Algorithms, in *Metaheuristics: Computer Decision-Making*, Mauricio G. C. Resende & Jorge P. de Sousa, (Eds.), 257-278, Kluwer Academic Publishers.
- Drezner Z. & G.A. Marcoulides (2008). On the Range of Tabu Tenure in Solving Quadratic Assignment Problems. Under review. Also presented at the 4th International Conference on Computer Science and Information Systems, Athens, Greece, July, 2008.
- Drezner Z. & S. Salhi (2002). Using Hybrid Metaheuristics for the One-Way and Two-way Network Design Problem. *Naval Research Logistics*, 49, 449-463.
- Drezner Z. & G.O. Wesolowsky (1997). Selecting an Optimum Configuration of One-Way and Two-Way Routes. *Transportation Science*, 31, 386-394.
- Drezner Z. & G.O. Wesolowsky (2003). Network Design: Selection and Design of Links and Facility Location. *Transportation Research Part A*, 37, 241-256.
- Edmands S. (1999). Heterosis and Outbreeding Depression in Interpopulation Crosses Spanning a Wide Range of Divergence. *Evolution*, 53, 1757-1768.
- Edmands S. (2002) Does Parental Divergence Predict Reproductive Compatibility? *Trends in Ecology & Evolution* 17, 520-527.
- Epelman M.; S. Pollock; B. Netter & B. Low (2005). Anisogamy, Expenditure of Reproductive Effort, and the Optimality of Having Two Sexes. *Operations Research*, 53, 560-567.
- Fleurent, C. & J.A. Ferland (1994). Genetic Hybrids for the Quadratic Assignment Problem. in *Quadratic Assignment and Related Problems* P. Pardalos & H. Wolkowicz, editors, , vol. 16, pp. 173-187. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*.
- Gambardella, L.; Taillard, E. & Dorigo, M. (1999). Ant Colonies for the Quadratic Assignment Problem. *Journal of the Operational Research Society*, 50, 167-176.
- Glover F. (1986) Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13, 533-549.
- Glover F. & Laguna M. (1997) *Tabu search*. Kluwer, Boston.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Wokingham, England.
- Hahn, P.M. & Krarup, J. (2001). A Hospital Facility Problem Finally Solved. *The Journal of Intelligent Manufacturing*, 12, 487-496.
- Hansen, P. & Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130, 449-467.
- Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- Koopmans T.C. & M. J. Beckmann (1957). Assignment problems and the location of economics activities. *Econometrica*, 25, 53-76.
- Li, Y.; Pardalos, P.M. & Resende, M. (1994). A Descent Randomized Adaptive Search Procedure for the Quadratic Assignment Problem. in P. Pardalos and H. Wolkowicz (Eds.), *Quadratic Assignment and Related Problems*, Vol. 16, pages 237-261. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*.
- Misevicius A. (2003). A Modified Simulated Annealing Algorithm for the Quadratic Assignment Problem. *Informatica*, 14, 1-18.

- Misevicius A. (2003a). Ruin and recreate principle based approach for the quadratic assignment problem. Lecture notes in computer science, vol 2723. In: Cant-Paz E, Foster JA, Deb K, et al (eds) Genetic and evolutionary computation—GECCO 2003 (Chicago, USA), Proceedings, Part I, pp 598–609. Springer, Berlin Heidelberg New York.
- Misevicius A. (2003b). Genetic algorithm hybridized with ruin and recreate procedure: application to the quadratic assignment problem. *Knowl Based Syst* 16:261–268.
- Misevicius A. (2004). An improved hybrid genetic algorithm: new results for the quadratic assignment problem. *Knowl Based Syst* 17:65–73.
- Misevicius A. (2005). A tabu search algorithm for the quadratic assignment problem. Working paper, Kaunas University of Technology, Kaunas, Lithuania *Comput Optim Appl* 30:95–111.
- Mladenovic, N. & P. Hansen (1997). Variable neighborhood search. *Computers and Operations Research* 24, 1097– 1100.
- Moscato, P. (2002). Memetic Algorithms. in *Handbook of Applied Optimization* P.M. Pardalos & M.G.C. Resende (Eds.), Oxford University Press, Oxford, U.K.
- Nystrom M. (1999). Solving certain large instances of the quadratic assignment problem: Steinberg's examples. Working paper, California Institute of Technology, 1999.
- Oliveira, C.A.S.; P.M. Pardalos & M.G.C. Resende (2004). GRASP with Path-relinking for the Quadratic Assignment Problem. In *Efficient and Experimental Algorithms*, LNCS, 3059, Ribeiro, C.C. and Martins, S.L. (eds.), 356-368, Springer-Verlag.
- Pelikan M.; S. Tsutsui & R. Kalapala (2007). Dependency Trees, Permutations, and Quadratic Assignment Problem. Missouri Estimation of Distribution Algorithms Laboratory (MEDAL) Report No. 2007003, University of Missouri in St. Louis, MO., January, 2007.
- Rendl, F. (2002). The Quadratic Assignment Problem. in *Facility Location: Applications and Theory* Z. Drezner and H. Hamacher (Eds.) Springer, Berlin.
- Rodriguez, J.M.; F.C. MacPhee; D.J. Bonham; J.D. Horton & V.C. Bhavsar (2004). Best Permutations for the Dynamic Plant Layout Problem. In *Proceedings of the 12th International Conference on Advances in Computing and Communications (ADCOM 2004)*. Dasgupta, A.R., Iyengar, S.S., and Bhatt, H.S. (Eds.), 173-178, Allied Publishers Pvt. Ltd., New Delhi Ahmedabad, India.
- Skorin-Kapov, J. (1990). Tabu Search Applied to the Quadratic Assignment Problem. *ORSA Journal on Computing*, 2, 33-45.
- Spears W.M. (2000). *The Role of Mutation and Recombination*, Natural Computing Series, Springer-Verlag.
- Steinberg, L. (1961). The Backboard Wiring Problem: a Placement Algorithm. *SIAM Review*, 3, 37-50.
- Taillard, E.D. (1991). Robust Tabu Search for the Quadratic Assignment Problem. *Parallel Computing*, 17, 443-455.
- Taillard, E.D. (1995). Comparison of Iterative Searches for the Quadratic Assignment Problem. *Location Science*, 3, 87-105.
- Tate, D.M. & A.E. Smith (1995). A Genetic Approach to the Quadratic Assignment Problem. *Computers and Operations Research*, 22, 73-83.
- Thonemann, U.W. & A. Bolte (1994). 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.
- Wilhelm, M.R. & T. L. Ward (1987). Solving Quadratic Assignment Problems by Simulated annealing. *IIE Transactions*, 19, 107-119.



Tabu Search

Edited by Wassim Jaziri

ISBN 978-3-902613-34-9

Hard cover, 278 pages

Publisher I-Tech Education and Publishing

Published online 01, September, 2008

Published in print edition September, 2008

The goal of this book is to report original researches on algorithms and applications of Tabu Search to real-world problems as well as recent improvements and extensions on its concepts and algorithms. The book's Chapters identify useful new implementations and ways to integrate and apply the principles of Tabu Search, to hybrid it with others optimization methods, to prove new theoretical results, and to describe the successful application of optimization methods to real world problems. Chapters were selected after a careful review process by reviewers, based on the originality, relevance and their contribution to local search techniques and more precisely to Tabu Search.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Zvi Drezner (2008). Tabu Search and Hybrid Genetic Algorithms for Quadratic Assignment Problems, Tabu Search, Wassim Jaziri (Ed.), ISBN: 978-3-902613-34-9, InTech, Available from:
http://www.intechopen.com/books/tabu_search/tabu_search_and_hybrid_genetic_algorithms_for_quadratic_assignment_problems

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.