Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research 34 (2007) 163–179

computers & operations research

www.elsevier.com/locate/cor

The quadratic assignment problem in the context of the printed circuit board assembly process

Ekrem Duman[a,*], Ilhan Or[b]

[a]Industrial Engineering Department, Doguş University, Acıbadem, Istanbul, Turkey
[b]Industrial Engineering Department, Bogazici University, Bebek, Istanbul, Turkey

Available online 29 June 2005

Abstract

The sequencing of placement and the configuration of the feeder is among the main problems involved in printed circuit board (PCB) assembly optimization. In some machine architectures, the latter problem can be formulated as the well known NP-hard quadratic assignment problem (QAP). In this study, a search is made among those metaheuristics that have recently found widespread application in order to identify a heuristic procedure that performs well with the QAP in the PCB assembly context. To this end, specific algorithms reflecting implementations of Taboo Search, Simulated Annealing and Genetic Algorithm-type metaheuristics were tested and compared using real PCB assembly data. The same set of algorithms was also tested on general QAP problems and it was observed that algorithms which performed successfully in the PCB context could perform poorly in the general situation. In the light of this, it can be concluded that ascertaining the best performing heuristic is complicated by the fact that the performance of a heuristic depends on the context of the problem, which determines the structure and relationships of problem parameters.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Metaheuristics; Combinatorial optimization; Quadratic assignment problem; Printed circuit board assembly

1. Introduction and problem definition

The use of automated placement machines has brought major improvements to printed circuit board (PCB) assembly operations when compared to the manual assembly of PCBs. Concomitant with their

* Corresponding author. Tel.: +90 216 327 11 04; fax: +90 216 327 96 31.
E-mail address: eduman@dogus.edu.tr (E. Duman).

use, however, are complicated operations research problems that need to be solved optimally (or near-optimally) in order that these machines can be better utilized. Such problems can be categorized into four classes [1–3]:

(i) Allocation of board types to assembly lines and board production scheduling.
(ii) Allocation of component types to placement machines on an assembly line.
(iii) Configuring the feeder on each placement machine.
(iv) Determining the placement sequence for each board type on each machine.

The structures of the first two problems show great variability depending on the characteristics of the manufacturing environment in question. On the other hand, the structures of the third and fourth problems can be regarded as known and fixed once the type of placement machine used is known [2]. In particular, the architecture of a typical placement machine, consisting of three main parts (feeder carriage, placement head and board carrier), and the principles underlying the operation of these parts determine the structures of the feeder configuration and placement sequencing problems.

The placement sequencing problem is mostly formulated as a traveling salesman problem (TSP) in which the distance between two placement points is found using the Chebyshev distance measure [2,4,5]. Yet there are other researchers who have formulated it as a rural postman problem [6]. While, with most of the placement machine types, the standard TSP formulation is appropriate, in some cases—due to the physical shape of the placement head—the placement sequencing problem turns out to be a precedence-constrained TSP [7,8]. In other cases, the result is a multiple TSP formulation [9].

Regarding the feeder configuration problem, in the case of some placement machine types this problem is either non-existent or very trivial [7], while in other cases it is formulated as either a linear assignment problem [10] or a quadratic assignment problem (QAP) [4,11].

In this study, the focus is on placement machine types where the feeder configuration problem can be formulated as a QAP. Such machines can be encountered in both insertion and surface mount technologies. We are concerned with a class of placement machines encountered in insertion technology in which the placement head is fixed, the board carrier is movable in two dimensions and the feeder carriage is linear and movable in one dimension. An example of this type of machine is the Panasonic variable center insertion machine, which is used to mount axial leaded components. This machine is depicted in Fig. 1 and its operation can be briefly described as follows.

The Feeder Carriage contains 60 cells. Different component types are stored in different feeder cells. The feeder carriage aligns the cell containing the component due to be placed with the placement head, by means of a single dimensional motion along the x-axis. The movement of the feeder carriage is not fast, and the time it takes for the desired cell to be aligned with the placement head depends on the position of the feeder carriage. This position is itself contingent on the identity of the previously placed component and the proximity of the desired cell to the insertion head.

The PCB onto which various electronic components are to be placed is set on the *Board Carrier*. Through a two-dimensional motion along the x and y axes, the Board Carrier aligns the exact location on the PCB at which the current component is to be placed with the placement head. The motions along the two axes are independent and high speed. Nevertheless, since many placements occur on a single PCB, the total distance traversed by the Board Carrier becomes important. This distance is directly related to the sequence in which components are placed on the PCB.

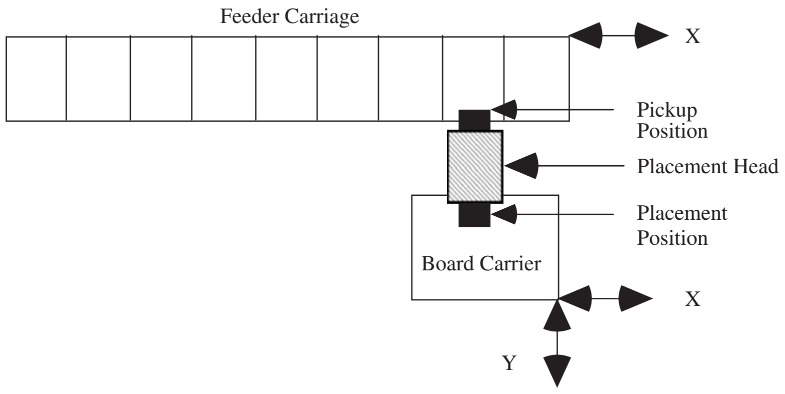

Fig. 1. Diagram of a numerically controlled component insertion machine.

The *Placement Head* picks up a component from the aligned feeder carriage cell and places it on the PCB set on the board carrier, which is aligned in such a way as to enable placement in a precise location. The desired leg spreads or placement angles of components may vary, so, if necessary, the placement head adjusts the leg spread and/or rotates before executing a placement. Leg spread adjustment is a slow operation and its duration depends on the previous leg spread setting of the placement head (i.e. the leg spread required for the component just placed). Rotation can be made concurrently with the downward movement of the head, but this requires no extra time.

As can be deduced from the above explanation, among the important factors contributing directly to the productivity of the assembly system of a PCB are the determination of the component placement sequence and the assignment of components to the feeder carriage cells in such a way as to minimize overall time losses owing to carrier board and feeder carriage movements and component leg spread adjustments.

The above-mentioned decisions related to the component placement sequence and the assignment of component types to feeder cells are interdependent and mutually influential. For this reason, they need to be considered simultaneously if an exact overall optimal solution is desired. However, such an approach can lead to a very complex combinatorial problem, since these two decisions—even independently—give rise to NP-hard problems. Accordingly, the two decisions are usually treated in the literature as sequential problems and modeled using two well-known models: the QAP and the TSP [1,2,4]. In other words, the overall problem is decomposed into a QAP sub-problem and a TSP sub-problem to be solved sequentially and iteratively.

In this research, the QAP is investigated in the context of "the feeder carriage configuration and placement sequencing problem in component placement machines". As explained above, the repeated solution of the QAP is needed in this situation in an iterative manner, so a fast and effective solution procedure becomes quite important. The QAP is actually a well-investigated NP-hard problem and many solution procedures have been proposed for it, with comparable, good performance records with randomly generated data. However, the performance of most solution procedures is dependent on problem parameters, whereby in many implementations (as in the above-mentioned case) problem data bear a certain structure. Given this, the present study identifies some critical problem parameters and the characteristics thereof in the QAP, parameters and characteristics which derive from the feeder carriage configuration problem.

Following this, the performance of various QAP solution procedures is investigated and compared, using two classes of test problem:

(i) problems arising from real PCB assembly environments;
(ii) problems with general characteristics.

In the next section, the formulation of the feeder configuration problem as a QAP is elaborated and basic solution procedures proposed for it in the literature are briefly discussed. Then, in Section 3, the solution procedures investigated in this study are described. The results obtained from the application of these algorithms to the QAP arising from real PCB assembly environments are discussed in Section 4. The performances of these algorithms for non-PCB, general QAP problems are also examined and the findings are discussed in Section 5. Finally, in Section 6, the major conclusions arrived at are summarized and directions for further studies are suggested.

2. The quadratic assignment problem

The QAP is a special kind of assignment problem frequently seen in layout and location studies. The main difference of the QAP from the classic assignment problem is that in the QAP there is interaction between assignment pairs, leading to a non-linear objective function. For example, for the facility layout problem, the assignment of department i to location k is interrelated with the assignment of some other department j to location l, since the transportation cost between the departments is related not only to the amount of flow between the departments, but also to the distance between them.

Such a structure can also be considered when determining how best to configure the feeder carriage. If we assume the placement sequence to be known, the number of times component-type j is placed immediately after component-type i determines the flow between i and j (f_{ij}). Then, to determine the cost of assigning component-type i to feeder cell k and at the same time assigning component-type j to feeder cell l (C_{ijkl}), we need to multiply f_{ij} by the time the feeder needs to align cell l with the placement head if the previously aligned cell is k. Since the feeder carriage is linear in our case, the movement time between any two cells is equal to the absolute difference of the cell numbers divided by the speed of the feeder carriage (assuming it has a uniform speed for all distances). Throughout this study, the feeder speed is taken as one cell per second. There is no loss of generality in doing so, since the feeder speed is the only machine parameter used in the QAP.

As was pointed out, the QAP has been studied extensively in the literature and many exact and heuristic solution procedures have been offered for it [12–16]. Most of the exact methods are based on "Branch and Bound" and/or "Cutting Plane" philosophies. Heuristic methods, on the other hand, can be classified as improvement methods, construction methods, or combinations of the two. In improvement methods, a randomly generated initial solution is improved through systematic exchanges between the components of the assignment vector, and the procedure continues until the solution cannot be improved further. In construction methods, assignments proceed one at a time, based on some analysis, until all have been completed. Hybrid techniques start with a construction procedure, in order to develop a satisfactory initial solution, and then continue with an improvement method to obtain better solutions. Two successful examples of hybrid techniques have been presented by Resende et al. [17] and Pardalos et al. [18]. In these studies, a general solver named GRASP is applied to the dense and sparse instances of QAP, respectively.

The authors first construct a good starting solution by trying to keep departments (component-types) having a high inter-flow close to each other and then improving the solution by pairwise exchanges.

In the above QAP formulation, it is assumed that each component type can occupy only one slot in the feeder carriage. In other words, it is assumed that the number of component types is equal to that of the cells available in the feeder carriage. If this is not the case, as when some components are allowed to be placed in more than one cell in the feeder carriage, then the complexity of the PCB assembly optimization problem increases considerably [19].

3. Description of algorithms

In our research, we investigated four classes of heuristic approaches:

(i) *Exchange procedures*: Given a feasible starting assignment vector, a *k-exchange* procedure investigates one-by-one all *k*-way exchanges of its components, implementing the exchange if an improvement is detected. This process continues until no further improvement can be obtained.

(ii) *Taboo search* (*TS*): It constitutes an extension to two-way exchanges [20]. Once pairwise exchanges reach a local optimum (i.e. no further improvements can be obtained), TS continues by choosing an exchange that least degrades the objective function. In order to avoid returning to the local optimum just visited, the reverse move is forbidden. A rolling Taboo List of size "s" specifies such forbidden moves. However, the taboo list may forbid certain interesting moves (e.g. those that lead to the best solution found so far). Consequently, an aspiration criterion is defined to allow taboo moves if they seem to be promising.

(iii) *Simulated annealing* (*SA*): This procedure also augments 2-way exchanges [21]. SA implements pairwise exchanges not only when they yield an improvement, but also (with a decreasing probability) when the objective value deteriorates. Thus, it opens the way to escape from local optima. The *acceptance probability* is set to $e^{(-\Delta/T)}$, where Δ is the deterioration level and T is a decreasing parameter (corresponding to temperature in the analogy with physical annealing). In the beginning stages, T is high and pairwise exchanges with small deterioration are likely to be accepted, so as to avoid being prematurely trapped in a local optimum. As pairwise exchanges continue, T approaches zero and most uphill moves are rejected.

(iv) *Guided evolutionary simulated annealing* (*GESA*): It is a *genetic* algorithm combining *Genetic Evolution* and *Simulated Annealing* [22]. It is parallel in the sense that many candidates are evaluated in parallel. Its "Regional Guidance" search strategy depends not only on the objective value of an individual candidate solution but on the performance of a subset of feasible solutions as well. It has a mechanism for measuring the qualities of many regions (families), and automatically focuses the search onto regions with higher chances of attaining the global optimum.

GESA starts with N random solutions (parents) and, initially, each parent generates the same number of children through random mutation—the exchange of two components of the parent. At each iteration (generation), there are 2 levels of competition. In local competition, the children in the same family (i.e. generated from the same parent) compete with each other and only the best survives. This child then competes with its parent to be the parent of the next generation. The competition among the families determines the number of children each family has in the next generation. The number of children that will be generated by each family (the overall total being fixed) is set as proportional to the *acceptance index* of the family. The acceptance index of a family is computed by counting those children whose objective values are smaller than the lowest objective value found up to the previous generation. The remaining children may also be included in the family acceptance index with a probability of $e^{(-\varDelta/T)}$, where $\varDelta$ is the difference from that lowest objective value and T is the constant annealing temperature parameter.

In this study, four Exchange Procedures have been investigated. They are as follows:

(i) 2 OPT: A starting solution was given and all 2-way exchanges were investigated.
(ii) 3 OPT: A starting solution was given and all 3-way exchanges were investigated.
(iii) $(2+3)$ OPT: Application of 2 OPT, followed by 3 opt.
(iv) $(3+2)$ OPT: Application of 3 OPT, followed by 2 opt.

Regarding Taboo Search methods, two slightly different Taboo philosophies, differing in their definitions of the Taboo list, have been investigated:

(i) The Taboo list contains pairs that are not to be exchanged for "s" iterations (TABOO1 procedures in Table 1).
(ii) A move is counted as Taboo if it assigns both interchanged entities to the locations they had earlier occupied within the last "s" iterations (TABOO2 procedures in Table 1).

Instead of a constant taboo list size, we used two values $S_{\min}$ and $S_{\max}$, whereby a switch between these two values occurred at every $2 * S_{\max}$ iteration.

In the simulated annealing (SIMAN) implementations, the following parameters were varied:

(i) *Initial temperature* (T): The larger this value, the more the inferior exchanges encouraged. In numerical computations, T was set to either 100 or 1000. For the real PCB problems considered in this study, T equaling 1000 corresponded roughly to accepting 10 percent worse solutions with a probability of 0.75, whereas T equaling 100 corresponded roughly to accepting 10 percent worse solutions with a probability of 0.05.
(ii) *Temperature decrease ratio* (a): After a predetermined number of iterations, T was set to T/a (i.e., $T := T/a$). When "a" was large, the temperature decrease was faster and the acceptance of inferior exchanges became less likely at a greater rate. In numerical computations, a was set to either 1.1 or 1.5.
(iii) *Number of iterations at each temperature setting* (R): Greater values of R correspond to slower cooling, that is, more exchanges occurring when there is a greater likelihood of inferior exchanges being accepted. In numerical computations, R was set to either 5 or 20.
(iv) *Increase ratio in iteration number at each setting* (b): After a predetermined number of iterations, R was set to $R * b$ (i.e. $R := R * b$). In numerical computations, b was set to either 1.1 or 1.5.

In the genetic algorithm (GESA) implementations, the following parameters were varied:

(i) *Number of initial solutions* (*parents*) (*P)*: The larger this number, the higher the potential of a "thorough" search of the feasible region and the lesser the chance of being trapped in a local optimum. In numerical computations, P was set to either 3 or 6.

Table 1
Type and parameter settings of the heuristics employed

Procedure name	Procedure type	Number of iterations	T	R	a	b	$S_{\min}$	$S_{\max}$	P	C
H1	2 OPT	—	—	—	—	—	—	—	—	—
H2	3 OPT	—	—	—	—	—	—	—	—	—
H3	2 + 3 OPT	—	—	—	—	—	—	—	—	—
H4	3 + 2 OPT	—	—	—	—	—	—	—	—	—
H5	SIMAN	$N^3/3$	100	5	1.5	1.1	—	—	—	—
H6	SIMAN	N^3	100	5	1.5	1.1	—	—	—	—
H7	SIMAN	$N^3/3$	100	20	1.5	1.1	—	—	—	—
H8	SIMAN	N^3	100	20	1.5	1.1	—	—	—	—
H9	SIMAN	$N^3/3$	100	20	1.1	1.5	—	—	—	—
H10	SIMAN	N^3	100	20	1.1	1.5	—	—	—	—
H11	SIMAN	$N^3/3$	1000	5	1.5	1.1	—	—	—	—
H12	SIMAN	N^3	1000	5	1.5	1.1	—	—	—	—
H13	SIMAN	$N^3/3$	1000	20	1.5	1.1	—	—	—	—
H14	SIMAN	N^3	1000	20	1.5	1.1	—	—	—	—
H15	SIMAN	$N^3/3$	1000	20	1.1	1.5	—	—	—	—
H16	SIMAN	N^3	1000	20	1.1	1.5	—	—	—	—
H17	TABOO1	$N^3/3$	—	—	—	—	10	15	—	—
H18	TABOO1	N^3	—	—	—	—	10	15	—	—
H19	TABOO1	$N^3/3$	—	—	—	—	10	25	—	—
H20	TABOO1	N^3	—	—	—	—	10	25	—	—
H21	TABOO1	$N^3/3$	—	—	—	—	20	25	—	—
H22	TABOO1	N^3	—	—	—	—	20	25	—	—
H23	TABOO2	$N^3/3$	—	—	—	—	10	15	—	—
H24	TABOO2	N^3	—	—	—	—	10	15	—	—
H25	TABOO2	$N^3/3$	—	—	—	—	10	25	—	—
H26	TABOO2	N^3	—	—	—	—	10	25	—	—
H27	TABOO2	$N^3/3$	—	—	—	—	20	25	—	—
H28	TABOO2	N^3	—	—	—	—	20	25	—	—
H29	GESA	$N^3/3$	100	—	—	—	—	—	3	9
H30	GESA	N^3	100	—	—	—	—	—	3	9
H31	GESA	$N^3/3$	1000	—	—	—	—	—	3	9
H32	GESA	N^3	1000	—	—	—	—	—	3	9
H33	GESA	$N^3/3$	100	—	—	—	—	—	3	30
H34	GESA	N^3	100	—	—	—	—	—	3	30
H35	GESA	$N^3/3$	1000	—	—	—	—	—	3	30
H36	GESA	N^3	1000	—	—	—	—	—	3	30
H37	GESA	$N^3/3$	100	—	—	—	—	—	6	18
H38	GESA	N^3	100	—	—	—	—	—	6	18
H39	GESA	$N^3/3$	1000	—	—	—	—	—	6	18
H40	GESA	N^3	1000	—	—	—	—	—	6	18
H41	GESA	$N^3/3$	100	—	—	—	—	—	6	60
H42	GESA	N^3	100	—	—	—	—	—	6	60
H43	GESA	$N^3/3$	1000	—	—	—	—	—	6	60
H44	GESA	N^3	1000	—	—	—	—	—	6	60

(ii) *Number of children per generation* (C): The larger the number, the higher the number of feasible solutions investigated. In numerical computations, C was set to 9, 18, 30 or 60.
(iii) *GESA temperature range* (T): The larger the value, the higher the probability that some inferior children will be included in the computation of the family acceptance index. In numerical computations, T was set to either 100 or 1000.

In all, a total of 44 heuristic procedures, including 4 Exchange, 12 TS, 12 SIMAN and 16 GESA procedures, have been experimented with in this study. The procedure type and parameter settings of these are displayed in Table 1.

In the Exchange Procedures, there was no limit to the number of iterations, since iterations theoretically continue as long as improvements are obtained. In the remaining procedures, N^3 was the investigated number of iterations (N being the number of component types/feeder cells). Each parameter setting was also tested for $(N^3/3)$ number of iterations. This was to see the effect of the number of iterations on the performance of the solution procedure.

From the experiments, we identified the following problem parameters that may play a key role in the definition of problem types and in the performance of the heuristic procedures:

(i) *Problem size* (N): As the problem size gets larger, the number of feasible solutions increases exponentially; accordingly, the performance of limited enumeration techniques may decrease. A far greater number of feasible solutions have to be investigated in order to cover a fixed percentage of the feasible region.
(ii) *Sparsity* (SP): It is associated with the percentage of zero cells in the flow matrix. As sparsity increases, the likelihood of gaining improvements through pairwise exchanges may decrease. This is because, when the majority of cells are zero cells, a pairwise exchange becomes incapable of reflecting any change in costs). Although this means faster convergence, the risk of getting locked at a local optimum may increase.
(iii) *Mean flow value* (μ): As the flow between locations increases, expected gains from a good quality assignment increase.
(iv) *Flow dominance* (FD): This is the standard deviation of flow values divided by the mean flow value. When values of FD are very low, all solutions display a similar quality and all algorithms may perform equally well. On the other hand, as FD increases to very high levels, there exists a dominant optimal solution. In this case, the expected gain from limited enumeration methods may decrease, since the likelihood of achieving significant exchanges decreases.

Or and Atik [23] previously examined the QAP with a similar objective of identifying solution procedures. However, in that study, the authors focused strictly on PCB context and experimented only with randomly generated data.

In the next section, we analyze the performances of the 44 heuristic algorithms described above when applied to several real PCB assembly problems.

4. Experimentation runs and results

To investigate the performances of these algorithms and to identify which ones are appropriate for real PCB assembly environments, test runs were made using real data. The data were obtained from a machine at a TV set manufacturing facility that mounts axial-leaded component types onto TV main boards. The number of QAP data sets obtained this way was 11, and the average number of component types used in each problem was around 50.

As stated in Section 2, the distance between two feeder cells was taken as the absolute difference between the cell numbers, while the feeder carriage was assumed to have a uniform speed of 1 cell/s. For the construction of the flow matrix, the placement sequence currently used in the TV manufacturing facility for these specific PCBs was deployed. That placement sequence had been generated using the nearest neighbor (NN) algorithm.

Each problem was solved three times, starting with different initial solutions. Instead of displaying full results (see [2] for a full tabulation), the average costs obtained in three runs were used, as it was thought to be more meaningful to consider the average cost of three runs than the minimum cost. In addition, the percentage deviations of heuristic results from the best solution (of the 44 heuristics) were calculated and are displayed in Table 2 . In a way, it was assumed that the best solution from among the 44 heuristics was a good proxy for the optimal value. The following remarks can be made on the performances of heuristic classes.

The performances of exchange procedures were found to be rather poor as compared to most SIMAN and GESA procedures (see the last column of Table 2, where average performances of algorithms over 11 board types are displayed). As can be expected, the performances of 2 + 3 OPT or 3 + 2 OPT were better than 2 OPT or 3 OPT, since wider ranges of exchanges were considered.

With the exception of H9, H10, H15 and H16, the SIMAN procedures performed very well. The difference of H9, H10, H15 and H16 from the other procedures is that they had (a, b) values of (1.1, 1.5) as compared to the others' values of (1.5, 1.1). This phenomenon can be explained as follows. When a is low, the algorithm accepts inferior solutions for a long time and it has difficulty in bouncing back to better solutions. When the iteration limit is reached, the algorithm may still be a long way off from yielding good solutions. It should also be noted that, as can be expected, the algorithms having iteration number, (ITE) $= N^3$, $R = 20$ and $T = 1000$ performed slightly better than those having ITE $= N^3/3$, $R = 5$ and $T = 100$. The correlation between algorithm performance and number of iterations will be discussed later in this section.

The Taboo implementations performed poorly in general, whereby there seems to have been no difference between the two approaches TABOO1 and TABOO2. The only difference of the Taboo implementations from 2 OPT was the existence of a Taboo List and the formers' acceptance of inferior solutions in order to avoid being stuck at a local optima. However, when the problem size is large, as in our test problems, the probability of trying taboo moves or getting stuck at a local optima decreases and both Taboo or 2 OPT function similarly. Still, 2 OPT performed better than Taboo since there is no iteration limit for 2 OPT, while one does exist for Taboo procedures.

The GESA procedures performed fairly well, excluding the ones where the number of children to be considered at each iteration was low (H29, H30, H31, H32). This was because a new generation was started quickly and no improvement, or an insignificant degree of improvement, could be recorded at each iteration. Also, similar to the SIMAN procedures, the algorithms with a higher iteration limit and temperature value performed better than the others.

In summary, the five best-performing algorithms were SIMAN H14, SIMAN H8, SIMAN H13, SIMAN H12 and GESA H36, respectively. A closer inspection of these algorithms reveals that they all share two key characteristics. First, the number of iterations at each temperature setting is large ($R = 20$ vs. $R = 5$ in SIMAN and a large C in GESA). Second, the temperature decrease ratio is large ($a = 1.5$ vs. $a = 1.1$).

Table 2
Deviation from best solution

		B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	Avg.
	N	60	54	52	52	50	48	48	47	34	35	35	
	SP	95	95	95	95	92	94	92	91	90	92	88	
	FD	4.97	5.03	4.86	4.78	4.51	4.58	4.39	4.29	4.37	3.67	1.51	
2 OPT	H1	0.12	0.15	0.09	0.14	0.07	0.09	0.03	0.04	0.08	0.10	0.03	0.08
3 OPT	H2	0.08	0.11	0.11	0.07	0.06	0.07	0.08	0.01	0.07	0.11	0.04	0.07
2 + 3 OPT	H3	0.12	0.07	0.11	0.06	0.07	0.03	0.06	0.04	0.06	0.06	0.03	0.06
3 + 2 OPT	H4	0.08	0.08	0.09	0.11	0.07	0.00	0.06	0.03	0.07	0.05	0.06	0.06
avg_opt		**0.10**	**0.10**	**0.10**	**0.09**	**0.07**	**0.05**	**0.06**	**0.03**	**0.07**	**0.08**	**0.04**	**0.07**
SIMAN	H5	0.02	0.30	0.00	0.00	0.04	0.02	0.02	0.17	0.02	0.05	0.17	0.07
SIMAN	H6	0.04	0.00	0.28	0.02	0.04	0.29	0.01	0.00	0.00	0.08	0.13	0.08
SIMAN	H7	0.03	0.05	0.07	0.04	0.01	0.07	0.01	0.02	0.05	0.00	0.02	0.03
SIMAN	H8	0.00	0.01	0.01	0.02	0.03	0.02	0.00	0.01	0.04	0.02	0.03	0.02
SIMAN	H9	0.23	0.39	0.37	0.35	0.15	0.36	0.21	0.20	0.27	0.48	0.23	0.29
SIMAN	H10	0.22	0.33	0.35	0.33	0.13	0.23	0.17	0.15	0.23	0.32	0.16	0.24
SIMAN	H11	0.03	0.03	0.04	0.08	0.01	0.09	0.01	0.01	0.03	0.00	0.10	0.04
SIMAN	H12	0.00	0.16	0.01	0.00	0.01	0.02	0.02	0.00	0.04	0.03	0.00	0.03
SIMAN	H13	0.09	0.02	0.00	0.06	0.00	0.01	0.03	0.03	0.01	0.05	0.00	0.03
SIMAN	H14	0.01	0.01	0.02	0.04	0.02	0.01	0.00	0.01	0.02	0.00	0.02	0.01
SIMAN	H15	0.82	0.88	0.89	0.82	0.61	0.82	0.55	0.59	0.48	0.71	0.50	0.70
SIMAN	H16	0.84	0.93	0.92	0.80	0.59	0.77	0.56	0.52	0.46	0.73	0.43	0.69
avg_siman		**0.19**	**0.26**	**0.25**	**0.21**	**0.14**	**0.23**	**0.13**	**0.14**	**0.14**	**0.20**	**0.15**	**0.19**
TABOO1	H17	0.09	0.21	0.13	0.15	0.08	0.05	0.06	0.05	0.09	0.11	0.05	0.10
TABOO1	H18	0.10	0.09	0.06	0.14	0.06	0.03	0.03	0.06	0.09	0.09	0.04	0.07
TABOO1	H19	0.11	0.11	0.11	0.17	0.07	0.09	0.04	0.03	0.04	0.14	0.05	0.09
TABOO1	H20	0.08	0.09	0.05	0.14	0.05	0.06	0.06	0.05	0.09	0.14	0.03	0.08
TABOO1	H21	0.11	0.15	0.13	0.12	0.08	0.09	0.06	0.08	0.11	0.13	0.05	0.10
TABOO1	H22	0.13	0.17	0.12	0.08	0.11	0.07	0.06	0.06	0.08	0.06	0.05	0.09
avg_taboo1		**0.10**	**0.14**	**0.10**	**0.13**	**0.08**	**0.06**	**0.05**	**0.05**	**0.08**	**0.11**	**0.05**	**0.09**
TABOO2	H23	0.16	0.10	0.16	0.15	0.05	0.09	0.08	0.04	0.08	0.08	0.06	0.10
TABOO2	H24	0.11	0.13	0.09	0.15	0.05	0.06	0.04	0.04	0.06	0.05	0.04	0.08
TABOO2	H25	0.12	0.17	0.11	0.13	0.07	0.08	0.07	0.07	0.04	0.15	0.05	0.10
TABOO2	H26	0.07	0.10	0.08	0.13	0.10	0.06	0.06	0.04	0.07	0.06	0.02	0.07
TABOO2	H27	0.09	0.14	0.13	0.12	0.09	0.07	0.05	0.04	0.08	0.17	0.06	0.09
TABOO2	H28	0.12	0.10	0.13	0.10	0.09	0.05	0.06	0.06	0.05	0.13	0.03	0.08
avg_taboo2		**0.11**	**0.12**	**0.12**	**0.13**	**0.07**	**0.07**	**0.06**	**0.05**	**0.06**	**0.11**	**0.04**	**0.09**

GESA	H29	0.69	0.77	0.53	0.40	0.42	0.51	0.67	0.29	0.33	0.73	0.16	0.50
GESA	H30	0.49	0.57	0.55	0.42	0.32	0.25	0.28	0.64	0.26	0.30	0.39	0.41
GESA	H31	0.31	0.29	0.65	0.29	0.22	0.25	0.24	0.43	0.46	0.17	0.13	0.31
GESA	H32	0.33	0.23	0.36	0.40	0.18	0.25	0.23	0.53	0.25	0.64	0.17	0.33
GESA	H33	0.12	0.11	0.14	0.20	0.05	0.11	0.12	0.10	0.08	0.12	0.08	0.11
GESA	H34	0.16	0.15	0.13	0.07	0.10	0.15	0.06	0.07	0.04	0.05	0.06	0.10
GESA	H35	0.05	0.03	0.06	0.07	0.04	0.02	0.03	0.03	0.04	0.06	0.02	0.04
GESA	H36	0.07	0.03	0.03	0.05	0.03	0.00	0.00	0.02	0.03	0.07	0.01	0.03
GESA	H37	0.25	0.13	0.20	0.17	0.12	0.19	0.12	0.17	0.07	0.12	0.05	0.14
GESA	H38	0.17	0.19	0.22	0.24	0.14	0.08	0.12	0.14	0.11	0.25	0.09	0.16
GESA	H39	0.09	0.05	0.25	0.05	0.05	0.03	0.06	0.04	0.06	0.07	0.04	0.07
GESA	H40	0.09	0.08	0.12	0.07	0.07	0.07	0.04	0.04	0.08	0.08	0.06	0.07
GESA	H41	0.08	0.08	0.04	0.08	0.04	0.01	0.06	0.02	0.04	0.03	0.02	0.05
GESA	H42	0.05	0.01	0.03	0.05	0.06	0.04	0.02	0.02	0.07	0.04	0.02	0.04
GESA	H43	0.08	0.01	0.00	0.01	0.04	0.02	0.04	0.03	0.06	0.05	0.04	0.03
GESA	H44	0.09	0.08	0.07	0.03	0.03	0.00	0.01	0.02	0.01	0.01	0.02	0.03
avg_gesa		**0.20**	**0.18**	**0.21**	**0.16**	**0.12**	**0.12**	**0.13**	**0.16**	**0.12**	**0.17**	**0.08**	**0.15**
avg_all		**0.16**	**0.18**	**0.18**	**0.16**	**0.11**	**0.13**	**0.10**	**0.11**	**0.11**	**0.16**	**0.09**	**0.14**

It can also be observed that a higher initial temperature and a higher number of iterations have positive effects in algorithmic performance.

If the problem sizes (N) of the 11 individual problems are grouped as $N = 50$ and 35 (according to which of these two values the actual value is closest to), it emerges that there are some slight differences in the ordering of the top-five heuristics for each group. For example, for $N = 50$ problems, the second best algorithm SIMAN H8 becomes slightly favorable to SIMAN H14 and occupies the first place. However, the ordering in the two groups is not very different and no surprises are yielded.

In the algorithms compared so far, the iteration limit has been either $N^3/3$ or N^3. A numerical comparison shows that N^3 iterations are better than $N^3/3$ with respect to all heuristic classes, with a comparative benefit of one percent being observed in the formed solutions. The gain is largest in TABOO implementations and is least in GESA implementations. When $5/3 * N^3$ iterations are allowed instead of N^3 iterations, on average the solutions are only superior to the degree of 0.4 percent. Thus, with regards to the increase in computation time with the number of iterations, we can say that N^3 is a good number for an iteration limit if such a limit is to be provided.

Lastly, the dependence of the average performances of algorithms on sparsity (SP) and flow dominance (FD) values was also investigated. When the problems were divided into two groups on the basis of their SP (either 95 or 90, depending on whichever was closer) it was observed that the algorithms performed on the average five percent better when SP = 90. Similarly, when problems were grouped as either FD = 5.0 or 4.5 we saw that algorithms performed approximately five percent better in the case of FD = 4.5. Although the size of the sample data set (i.e. the number of problems investigated) is not sufficient for us to arrive at a definitive statistical conclusion, it does seems that, as SP or FD increases, the average performances of algorithms slightly decrease. This result is consistent with the interpretation given of these problem parameters in the previous section.

In the next section, we discuss the performances of these algorithms on general QAP problems.

5. Performances of the algorithms on general QAP problems

Some experimentation was also carried out to test the performances of the 44 heuristic algorithms on general QAP problems. In this study, the term "general QAP problems" denotes facility-layout-like problems, where the material flows between departments are high and the sparsity levels of the flow matrices are usually lower.

A random problem generation scheme was used to obtain general QAP problems. Through this scheme, problem size, sparsity and flow dominance values were taken as input, and random problems satisfying these parameters were created. Specifically, random problems with $N = 35$ or 50 (similar sizes to those in the real PCB problems covered in the previous section), SP = 50, 85 or 95, and, FD = (approximately) 1.0, 2.5 or 4.5 were generated, while the mean flow value for all generated problems was taken as 10. Each generated problem was solved three times with different initial solutions by all algorithms. For the comparison of the algorithms, the deviations of algorithm results from the best solution in percentage form were considered again and displayed in Table 3. In this table, the problems are identified by six digits, where the first two digits show the problem size, the middle two digits show the sparsity as a percentage, and the last two digits show ten times the flow dominance value. For example, if the problem name is 358525, it means that $N = 35$, SP = 85 and FD = 2.5.

Table 3
Deviation from best solution for random problems

		355010	355025	355045	358525	358545	359545	505010	505025	505045	508525	508545	509545	Avg.
	N	35	35	35	35	35	35	50	50	50	50	50	50	
	SP	50	50	50	85	85	95	50	50	50	85	85	95	
	FD	1.0338	2.5611	4.8375	4.3961	4.8943	4.4857	1.0348	2.4574	4.2741	2.5481	4.4737	4.4667	
2 OPT	H1	0.03	0.09	1.81	0.26	0.47	0.28	0.02	0.07	0.39	0.04	0.26	0.16	0.32
3 OPT	H2	0.01	0.08	1.14	0.26	0.31	0.14	0.01	0.03	0.14	0.03	0.19	0.13	0.21
2 + 3 OPT	H3	0.01	0.07	0.86	0.05	0.22	0.16	0.01	0.03	0.15	0.03	0.18	0.08	0.15
3 + 2 OPT	H4	0.00	0.04	0.94	0.12	0.36	0.06	0.01	0.02	0.18	0.05	0.20	0.11	0.17
avg_opt		**0.01**	**0.07**	**1.19**	**0.17**	**0.34**	**0.16**	**0.01**	**0.04**	**0.21**	**0.03**	**0.21**	**0.12**	**0.21**
SIMAN	H5	0.09	0.36	4.44	1.12	2.72	0.93	0.08	0.42	1.14	0.35	1.19	1.00	1.15
SIMAN	H6	0.11	0.47	5.11	1.16	2.55	1.22	0.10	0.39	1.26	0.35	0.88	1.11	1.23
SIMAN	H7	0.08	0.24	3.12	0.80	1.09	0.50	0.07	0.27	0.80	0.24	0.49	0.54	0.69
SIMAN	H8	0.05	0.29	3.06	0.66	1.50	0.59	0.05	0.24	0.78	0.25	0.66	0.69	0.73
SIMAN	H9	0.12	0.30	1.12	0.42	0.64	0.33	0.09	0.14	0.17	0.13	0.13	0.10	0.31
SIMAN	H10	0.11	0.20	0.00	0.14	0.10	0.14	0.09	0.12	0.09	0.11	0.00	0.08	0.10
SIMAN	H11	0.12	0.38	5.61	1.14	2.26	1.05	0.11	0.38	1.23	0.25	1.17	0.68	1.20
SIMAN	H12	0.13	0.42	4.29	1.59	3.44	1.46	0.09	0.35	1.27	0.30	1.01	1.10	1.29
SIMAN	H13	0.07	0.24	3.25	0.67	1.11	0.53	0.07	0.21	0.81	0.27	0.68	0.54	0.71
SIMAN	H14	0.07	0.24	3.87	0.75	1.07	0.52	0.06	0.19	0.62	0.23	0.65	0.65	0.74
SIMAN	H15	0.12	0.20	1.21	0.30	0.28	0.14	0.09	0.08	0.00	0.08	0.12	0.06	0.22
SIMAN	H16	0.09	0.11	0.31	0.00	0.00	0.00	0.08	0.03	0.01	0.04	0.06	0.00	0.06
avg_siman		**0.10**	**0.29**	**2.95**	**0.73**	**1.40**	**0.62**	**0.08**	**0.24**	**0.68**	**0.22**	**0.59**	**0.55**	**0.70**
TABOO1	H17	0.03	0.11	1.12	0.22	0.69	0.21	0.02	0.07	0.27	0.06	0.28	0.18	0.27
TABOO1	H18	0.00	0.03	0.90	0.14	0.13	0.06	0.00	0.01	0.11	0.01	0.07	0.08	0.13
TABOO1	H19	0.02	0.08	1.34	0.30	0.41	0.22	0.02	0.08	0.26	0.06	0.24	0.16	0.27
TABOO1	H20	0.01	0.02	0.82	0.16	0.24	0.07	0.00	0.02	0.10	0.01	0.08	0.11	0.14
TABOO1	H21	0.03	0.05	1.35	0.24	0.53	0.18	0.01	0.07	0.25	0.06	0.26	0.20	0.27
TABOO1	H22	0.00	0.02	0.64	0.16	0.16	0.15	0.01	0.01	0.05	0.00	0.11	0.11	0.12
TABOO2	H23	0.02	0.06	1.30	0.30	0.60	0.17	0.02	0.06	0.33	0.04	0.22	0.19	0.28
TABOO2	H24	0.01	0.00	0.49	0.14	0.17	0.06	0.00	0.01	0.11	0.00	0.07	0.12	0.10
TABOO2	H25	0.02	0.07	1.54	0.19	0.58	0.23	0.03	0.07	0.26	0.05	0.24	0.12	0.28
TABOO2	H26	0.01	0.01	0.62	0.13	0.27	0.02	0.00	0.01	0.08	0.01	0.09	0.11	0.11
TABOO2	H27	0.02	0.11	1.24	0.27	0.31	0.22	0.02	0.06	0.26	0.05	0.25	0.18	0.25
TABOO2	H28	0.01	0.04	0.66	0.18	0.14	0.09	0.01	0.00	0.10	0.01	0.16	0.06	0.12
avg_taboo		**0.01**	**0.05**	**1.00**	**0.20**	**0.35**	**0.14**	**0.01**	**0.04**	**0.18**	**0.03**	**0.17**	**0.14**	**0.19**

Table 3 (*continued*)

		355010	355025	355045	358525	358545	359545	505010	505025	505045	508525	508545	509545	Avg.
GESA	H29	0.15	0.57	6.80	1.90	2.65	1.49	0.12	0.40	1.54	0.49	1.53	1.16	1.57
GESA	H30	0.18	0.65	6.30	1.63	2.63	1.89	0.12	0.45	1.81	0.41	1.57	1.43	1.59
GESA	H31	0.13	0.52	6.58	1.42	2.55	1.80	0.10	0.44	1.33	0.42	1.21	1.23	1.48
GESA	H32	0.14	0.63	6.39	1.30	2.84	1.20	0.11	0.44	1.99	0.42	1.51	1.40	1.53
GESA	H33	0.06	0.29	3.28	0.91	1.51	0.63	0.06	0.27	1.03	0.21	0.82	0.72	0.82
GESA	H34	0.05	0.33	3.87	0.77	1.63	1.05	0.07	0.21	0.99	0.26	0.90	0.74	0.91
GESA	H35	0.08	0.29	3.90	0.93	1.59	0.87	0.06	0.25	1.05	0.25	0.78	0.80	0.90
GESA	H36	0.08	0.31	4.26	0.87	1.81	0.59	0.07	0.28	1.00	0.22	0.92	0.90	0.94
GESA	H37	0.06	0.38	5.31	1.40	1.57	0.86	0.07	0.28	1.05	0.25	0.91	0.85	1.08
GESA	H38	0.08	0.45	5.77	0.90	2.31	0.88	0.07	0.27	1.13	0.27	0.89	0.76	1.15
GESA	H39	0.07	0.37	4.53	1.82	1.93	1.20	0.08	0.27	1.22	0.25	0.94	0.85	1.13
GESA	H40	0.08	0.37	4.58	0.80	2.04	1.14	0.07	0.28	0.99	0.28	1.05	0.88	1.05
GESA	H41	0.05	0.18	2.41	0.62	1.03	0.63	0.04	0.13	0.62	0.17	0.78	0.56	0.60
GESA	H42	0.05	0.19	3.51	0.82	1.03	0.53	0.03	0.16	0.69	0.15	0.66	0.52	0.69
GESA	H43	0.06	0.22	2.59	0.52	1.14	0.56	0.04	0.16	0.69	0.13	0.67	0.59	0.61
GESA	H44	0.05	0.23	2.80	0.56	1.66	0.63	0.04	0.14	0.59	0.12	0.67	0.57	0.67
avg_gesa		**0.06**	**0.30**	**3.90**	**0.91**	**1.60**	**0.80**	**0.06**	**0.22**	**0.92**	**0.21**	**0.83**	**0.73**	**0.88**

In order to enable a closer correspondence to the performance tests applied to the problems originating from PCB environments, the starting initial temperature values of SIMAN and GESA procedures were modified. For this purpose, it was determined that the current T values needed to be multiplied by a factor of 213 and 434 for the $N = 35$ and 50 cases, respectively, in order to accept ten percent inferior solutions (at the beginning of the procedure) with a probability of 0.75. Thus, in this section, the SIMAN and GESA procedures were run with these modified T values.

The results in Table 3 show that, similar to the PCB case, the general performances of algorithms were better in problems having smaller SP and FD values. To be more specific, the best average performances were obtained for instances where $\mathrm{SP} = 50$ and $\mathrm{FD} = 1.0$ (see the last row of Table 3). This was what was expected given the interpretations proposed for these problem parameters when they were first defined in Section 3.

The results in Table 3 also show that all SIMAN and GESA procedures performed poorly under general QAP problems. A further indicator of this observation, not shown in Table 3, is that most SIMAN and GESA procedures did not reach the iteration limit but stopped much before that, since no improvements were recorded at the last temperature setting or generation. A conjecture for this interesting phenomenon is as follows. In this case, the mean flow value between departments was quite large, and the probability of considering highly inferior pairwise exchanges was significant at a particular iteration. This may be why the algorithms terminated prematurely, failing to execute any pairwise exchanges at a particular temperature setting. This argument would explain the poor performance of these algorithms in the general context even though they were seen to perform quite well for problems arising in PCB assembly environments. It should be noted that this argument is all the more valid for problems having larger FD values, as can be seen from Table 3.

On the other hand, it was observed that SIMAN algorithms H9, H10, H15 and H16, which had performed very poorly on QAP problems arising in PCB assembly environments, now performed quite satisfactorily. The explanation of this observation is consistent with the above argument: The cooling rate a was lower (1.1) and the number of iterations allowed at each temperature setting b increased faster (1.5). Thus, considering the higher number of exchanges and accepting inferior solutions with a higher probability, the algorithm could continue right up to the iteration limit, and at the end better solutions were yielded.

In addition, the k-OPT and Taboo procedures performed better for general QAP problems. Regarding the better performance of straightforward exchange procedures, we have the following conjecture. In general QAP problems, even with high sparsity and FD values, the flow matrix is less “structured” as compared to the flow matrix of PCB environment problems. While the distribution of non-zero elements in the flow matrix is almost homogeneous in a general QAP, in a PCB QAP they are clustered in certain rows. This is because, in PCB problems, a few specific component types are placed in large numbers whereas the rest are placed only a few times. Thus, it can be argued that, in general QAP problems, the qualities of feasible solutions are not so different. Accordingly, local optima are not as inferior to the global optimum as in the PCB case, so that a very effective local optimum-seeking procedure, like the k-exchange procedure, performs better in this case.

To summarize, the difference in the performances of the algorithms in the two cases (PCB QAP and the general QAP) can mainly be attributed to two factors: the special structure of PCB problems and the (usually) higher flow values in general QAP problems.

6. Summary, conclusions and suggestions for further study

Forty-four heuristic algorithms were tested on 11 real PCB assembly data, in an attempt to solve the feeder configuration problem of machine architecture discussed above. The best performing algorithm, SIMAN H14, was chosen to be used in the iterative solution methodology of the QAP and the TSP.

With regards to the number of iterations necessary for the described procedures to perform effectively, N^3 was observed to be a reasonably realistic limit, if such a limit is to be provided.

The performances of the algorithms were also tested on general QAP problems. The results are interesting in that the algorithms which performed best with respect to the PCB-based problems performed the worst on general problems. It is thus evident that the performance of algorithms (and their settings) depends heavily on the problem parameters, pointing to the necessity to conduct a similar study in order to find algorithms with a better performance in specific problem areas.

In this study, one essential assumption made is that any component type would be assigned to a single cell in the feeder carriage. In real PCB assembly environments, however, the number of feeder cells can be larger than the number of component types, so that some of the component types could be assigned to more than one feeder cell. This possibility might well bring about a decrease in the total assembly time. In such a case, the problem would become much more complicated, but since the effort required to examine the problem can be deemed worthwhile, this case can be proposed as a future area of study.

On the other hand, as was mentioned in Section 1, the feeder configuration problem that leads to the QAP investigated in this study is actually a sub-problem of the more complex and interesting PCB assembly optimization problem. Hence, more research focusing on the integration of the feeder configuration problem with the component placement sequencing problem is called for. In this regard, other important issues that require investigation are the performance of the traveling salesman problem solution procedures under the Chebyshev distance measure and the identification of an appropriate procedure to be coupled with the QAP procedure.

References

[1] McGinnis LF, Ammons JC, Carlyle M, Cranmer L, Depuy GW, Ellis KP, Tovey CA, Xu H. Automated process planning for printed circuit card assembly. IIE Transactions 1992;24/4:18–29.

[2] Duman E. Optimization issues in automated assembly of printed circuit boards. PhD dissertation, Bogazici University, Turkey, 1998.

[3] Ji P, Wan YF. Planning for printed circuit board assembly: the state-of-the-art review. International Journal of Computer Applications in Technology 2001;14(4–6):136–44.

[4] Leipala T, Nevalainen O. Optimization of the movement of a component placement machine. European Journal of Operational Research 1989;38:167–77.

[5] Foulds LR, Hamacher HW. Optimal bin location and sequencing in printed circuit board assembly. European Journal of Operational Research 1993;66:279–90.

[6] Ball MO, Magazine MJ. Sequencing of insertions in printed circuit board assembly. Operations Research 1988;36: 192–201.

[7] Duman E, Or I. Precedence constrained TSP arising in printed circuit board assembly. International Journal of Production Research 2004;42:67–78.

[8] Chan D. Precedence constrained TSP applied to circuit board assembly and no wait flow-shop. International Journal of Production Research 1993;31:2171–7.

[9] Duman E, Or I. Constrained multiple TSP arising in printed circuit board assembly. Proceedings of the engineering technology research conference, South Africa: Vanderbijlpark; 2002. p. 9–13.

[10] Drezner Z, Nof SY. On optimizing bin packing and insertion plans for assembly robots. IIE Transactions 1984;16/3: 262–70.

[11] Grotzinger S. Feeder assignment models for concurrent placement machines. IIE Transactions 1992;24(4):31–47.

[12] Burkard RE. Locations with spatial interactions: the quadratic assignment problem. In: Mirchandani PB, Francis RL, editors. Discrete location theory. Wiley Intersection series in discrete mathematics and optimization, 1990. p. 387–437.

[13] Pardalos PM, Rendl F, Wolkowicks H. The quadratic assignment problem: a survey and recent developments. In: DIMACS series in discrete mathematics and theoretical computer science. Providence, RI: American Mathematical Society; 1994.

[14] White DJ. A parametric based heuristic program for the quadratic assignment problem. Naval Research Logistics 1993;40:553–68.

[15] Burkard RE, Cela E, Pardalos PM, Pitsoulis LS. The quadratic assignment problem. In: Du DZ., Pardalos PM, editors. Handbook of combinatorial optimization, vol. 3. Dordrecht: Kluwer Academic Publishers; 1999. p. 241–337.

[16] Pardalos PM, Pitsoulis LS, editors. Nonlinear assignment problems: algorithms and applications. Kluwer Academic Publishers, 2000.

[17] Resende MGC, Pardalos PM, Li Y. Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problems using GRASP. ACM Transactions on Mathematical Software 1996;22:104–18.

[18] Pardalos PM, Pitsoulis LS, Resende MGC. Algorithm 769: Fortran subroutines for approximate solution of sparse quadratic assignment problems using GRASP. ACM Transactions on Mathematical Software 1997;23:196–208.

[19] Crama Y, Flippo OE, van de Klundert J, Spieksma FCR. The component retrieval problem in printed circuit board assembly. International Journal of Flexible Manufacturing Systems 1996;8:287–312.

[20] Taillard E. Robust taboo search for the quadratic assignment problem. Parallel Computing 1991;17:443–55.

[21] Connaly DT. An improved annealing scheme for the quadratic assignment problem. European Journal of Operational Research 1990;46:93–100.

[22] Yip PPC, Pao YH. A guided evolutionary simulated annealing approach to the quadratic assignment problem. IEEE Transactions on Systems, Man and Cybernetics 1994;24:1383–7.

[23] Or I, Atik K. A quadratic assignment problem in the automated assembly of printed circuit boards. Journal of the Operations Research Society of Turkey 2000;11:67–88.