Electronics and Communications in Japan, Part 2, Vol. 90, No. 12, 2007
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J86-D-I, No. 11, November 2003, pp. 791–799

Effectiveness of an Evolutionary Algorithm for the Multi-objective Rectangular Packing Problem

Shinya Watanabe,[1,*] Tomoyuki Hiroyasu,[2] and Mitsunori Miki[2]

[1]Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, 135-0064 Japan

[2]Faculty of Engineering, Doshisha University, Kyo-tanabe, 610-0321 Japan

SUMMARY

This paper considers the rectangular packing problem and investigates the effectiveness of the neighborhood cultivation genetic algorithm (NCGA). NCGA, which we proposed, is a new algorithm in which the unique mechanism of neighborhood crossover is combined with effective mechanisms for search in multi-objective GA proposed in the past. The effectiveness of the proposed method in typical test problems has already been investigated with satisfactory results in past studies. The rectangular packing problem, on the other hand, is applied to floor planning, such as chip area minimization in large-scale integrated circuits. It is a kind of discrete combinatorial problem in which it is known that the search is difficult and a very long time is required until the solution is obtained. This paper formulates the rectangular packing problem as a multi-objective problem with the vertical and horizontal lengths of the placement configuration as the objectives. The sequence-pair is used as the block placement representation, and PPEX is used as the crossover procedure. Using these processes, the effectiveness of NCGA is investigated. For comparison, three other methods—NSGA-II, SPEA2, and non-NCGA (NCGA without neighborhood crossover)—are investigated. Electron Comm Jpn Pt 2, 90(12): 111–120, 2007; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ecjb.20427

Key words: rectangular packing problem; multi-objective optimization; genetic algorithm; neighborhood cultivation GA; sequence-pair.

1. Introduction

The application of the genetic algorithm (GA) to multi-objective optimization started with VEGA [1] proposed by Schaffer and colleagues. The technique has been more intensively investigated recently, with marked progress [2]. Many studies of multi-objective GA have presented excellent original algorithms with excellent results [2–5]. Among typical algorithms, NSGA-II [3] proposed by Deb and colleagues and SPEA2 [4] proposed by Zitzler and colleagues are considered to have markedly superior performance.

We have proposed the neighborhood cultivation genetic algorithm (NCGA), a technique for the multi-objective GA, which has a feature not found in the above methods [5]. NCGA is an algorithm which has a unique mechanism of neighborhood crossover, and also uses effective mechanisms of superior methods proposed in the past, such as

Contract grant sponsor: Doshisha University Science Frontier Research Project "Intelligent Information Science and Its Application," based on the Science Frontier Promotion Plan of the Ministry of Education and Science.

*Presently affiliated with the Department of Computer Science and Systems Engineering, Muroran Institute of Technology

NSGA-II and SPEA2. The effectiveness of the proposed method in typical test problems has already been investigated with satisfactory results in past studies [5].

There are many applications of the multi-objective GA to real problems [2]. This paper attempts to apply NCGA to the rectangular packing problem, in which multiple rectangular blocks are to be placed in the minimum area. The rectangular packing problem is applied to a wide range of problems, such as floor planning [6, 7], where the chip area in large-scale integrated circuit (LSI) is to be minimized, and the placement of installations and offices in a plant [8]. The rectangular packing problem is a kind of combinatorial optimization problem which has the property that the number of possible combinations increases exponentially with the number of blocks.

This paper considers the rectangular packing problem and attempts a formulation as a multi-objective optimization problem, with the vertical and horizontal lengths of the finally obtained placement area as the objective variables. The purpose of this formulation is that the solution selector can select the aspect ratio of the configuration to place the blocks, in addition to minimizing the area. This paper uses the sequence-pair as the representation for block placement [7], and PPEX [9] as the crossover procedure.

As examples of numerical calculation, the proposed NCGA [5] is applied to the multi-objective rectangular packing problem. The effectiveness of NCGA compared to SPEA2 and NSGA-II is evaluated. The discussion of the usefulness of the problem formulation, the packing representation, and the crossover procedure in this study will be pursued further in the future.

2. Multi-objective Optimization Problem

The multi-objective optimization problem is defined as the problem in which k mutually competing objective functions $\vec{f}(\vec{x})$ are to be minimized under m inequality constraints [2]. In the multi-objective optimization problem, it is difficult to arrive at a unique optimal solution if the objective functions are in a trade-off relation. In order to handle such a situation, the concept of the Pareto optimal solution instead of the optimal solution is introduced.

The Pareto optimal solution is defined on the basis of the dominance relation of the solutions in the multi-objective optimization problem. The dominance relation of the solutions in minimization in the multi-objective optimization problem is defined as follows.

Definition (dominance relation): Let $\vec{x}_1, \vec{x}_2 \in R^n$.

If $f_i(\vec{x}_1) \leq f_i(\vec{x}_2)$ ($\forall i = 1, \ldots, k$) and $f_i(\vec{x}_1) < f_i(\vec{x}_2)$ ($\exists i = 1, \ldots, k$), it is defined that $\vec{x}_1$ dominates $\vec{x}_2$.

Based on the above dominance relation, the Pareto optimal solution is defined as follows.

Definition (Pareto optimal solution): Let $\vec{x}_0 \in R^n$.

(a) If there is no $\vec{x} \in R^n$ which dominates $\vec{x}_0$, $\vec{x}_0$ is called a (strong) Pareto optimal solution.

(b) If there is no $\vec{x}^* \in R^n$ such that $f_i(\vec{x}^*) < f_i(\vec{x}_0)$($\forall i = 1, \ldots, k$), $\vec{x}_0$ is called a weak Pareto optimal solution.

3. Multi-objective Genetic Algorithm

The application of GAs to multi-objective optimization is a subject of intensive study, and many algorithms have been proposed with excellent results [2–5]. In particular, powerful algorithms such as NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm) [3] and SPEA2 (Strength Pareto Evolutionary Algorithm 2) [4] have been proposed recently, demonstrating better results than the preceding methods.

NSGA-II is an algorithm proposed in 2001 by Deb and Agrawal as an improved version of NSGA [2]. NSGA-II includes additional features such as the preservation of nondominant solutions, crowded tournament selection, and the crowding distance, compared to NSGA. SPEA2 is an algorithm proposed in 2001 by Zitzler and colleagues [4] as an improved version of SPEA. Compared to SPEA, matching degree assignment and the archive truncation method are added in SPEA2, and the handling of the set of archive individuals is also modified.

These algorithms have many mechanisms in common. Among these, the following mechanisms are especially helpful in the search:

(a) Preservation of found superior solutions
(b) Reflection of preserved superior solutions on the search
(c) Reduction of preserved superior individuals
(d) Normalization of objective scales

Comparing the multi-objective GA to the single-objective GA, which is used in single-objective optimization, there is a great difference between the two in the search process by the GA. In the single-objective case, global search is performed in an early stage of search, maintaining the diversity of the ensemble, and local search is performed in the latter stages of the search, arriving at a unique optimal solution. In the case of multi-objective GA, on the other hand, the purpose is to derive the set of multiple Pareto optimal solutions. Thus, it is required to maintain diversity from the early stage to the last stage of search, and to perform local search at each search point.

The reason for the above search mode is that the search for the whole region containing the set of Pareto optimal solutions cannot be realized if diversity is lost, and a highly accurate Pareto optimal solution cannot be obtained unless local search is performed.

In order to achieve fast and highly accurate solution search for the whole region containing the set of Pareto optimal solutions, the following three mechanisms will be necessary:

(1) A mechanism to perform fast solution search near the set of Pareto optimal solutions

(2) A mechanism to maintain the diversity of the ensemble

(3) A mechanism to perform local search

In past algorithms, items (a) and (b) correspond to mechanism (1), and items (c) and (d) correspond to mechanism (2).

4. Neighborhood Cultivation Genetic Algorithm

As pointed out in the previous section, many multi-objective GAs presented recently use similar mechanisms to achieve the same purpose. It is expected that more accurate solution will be obtained by adding a procedure to achieve local search to such algorithms.

Consequently, we proposed a new method called the neighborhood cultivation genetic algorithm (NCGA) [5], which includes a new mechanism of neighborhood crossover to achieve local search. Neighborhood crossover is a crossover procedure in which the closeness between the individuals is considered in the selection of parent individuals for crossover. The neighborhood in this paper is defined not as the structural closeness between the individuals, but as the distance between the individuals in the objective function space remaining within the specified value. Thus, NCGA is an algorithm in which a new procedure of neighborhood search is added to a mechanism that is useful in the search in the multi-objective GA discussed in the previous section.

4.1. Outline of neighborhood cultivation genetic algorithm

The flow in the proposed model is as follows.

Step 1

The initial individuals are generated. The generation number is set as $t = 1$. The individuals are evaluated. The set of initial individuals is defined as the set of archive individuals (A_t).

Step 2

The set of archive individuals (A_t) is copied to the set of search individuals (P_t). P_t is sorted according to one of the objective function values. The objective function $f_i(x)$ to be used as the reference is determined from the remainder when the generation number t is divided by the number M of objective functions ($t \equiv i(\mathrm{mod}\ M)$). The variable i representing the individual number to be used in steps 3 and 4 is initialized as 0.

Step 3

From the set of search individuals (P_t) sorted in step 2, the two adjacent individuals, that is, the i-th and the (i + 1)-th, are selected as the set of pair individuals.

Step 4

For the set of selected pair individuals, crossover, mutation, and evaluation are applied, and the set of pair individuals is updated. To variable i is added 2. Steps 3 and 4 are repeated until i is equal to the number of individuals. As a result, the set of search individuals is completely updated (P_{t+1}).

Step 5

The set of search individuals (P_{t+1}) and the set of archive individuals (A_t) are compared, and the set of archive individuals is updated (A_{t+1}). In the process, the environmental selection process in SPEA2 is used.* In the same way as in SPEA2 and NSGA-II, the set of archive individuals retains the preset number N of individuals as superior individuals.

Step 6

It is decided whether the termination condition is satisfied. The termination condition is defined in this paper as the arrival at the preset number of generations (termination generation). If the termination condition is satisfied, the procedure ends. If not, the generation number is updated as $t = t + 1$ and the procedure goes back to step 2.

Thus, the proposed NCGA performs genetic manipulations as follows. The set of search individuals is sorted on an arbitrary objective function axis. Then, two adjacent individuals are selected as the set of pair individuals, and genetic manipulations are applied. The adjacent individuals in the sorted set of individuals are close together in the objective function space. Consequently, by applying crossover between two adjacent individuals, neighborhood crossover is achieved.

It should be noted, however, that if the above sort is iteratively applied on the same objective function axis, there is a danger that crossover will be iteratively applied only within the same pair. In order to avoid this situation, the following procedure is used.

- The objective function to be used as the reference is changed for each generation.
- Neighborhood shuffling is applied with an approximately 10% width of the ensemble size to the set of search individuals after sorting.

*Generally, when the preserved set of archive individuals is updated on the basis of the newly obtained set of search individuals, the new set of superior individuals is generated through selection from the sum of the set of search individuals and the set of superior individuals. This selection process is called environmental selection [4].

The objective function $f_i(x)$ which is to be used as the reference in sorting is determined from the remainder when the generation number t is divided by the number M of objective functions ($t \equiv i \pmod{M}$). By this process, the reference changes for each generation, and the pair to which crossover is applied, easily changes. Neighborhood shuffle is the process in which individuals are rearranged at random within a certain range. Consider, for example, a set of 100 individuals. The individuals are shuffled using a random variable in the range of 10 individuals at the maximum. By this process, the pair is modified even in the final stage of search in which there is little change in the ensemble.

Furthermore, in the updating of the set of archive individuals (A_t), the environmental selection of SPEA2 is applied, which uses the matching degree assignment and the truncation process [4]. The environmental selection of SPEA2 is outlined as follows.

Step 1: The sum set R_t of the set of search individuals (P_{t+1}) and the set of archive individuals (A_t) is formed.

Step 2: The matching degree assignment of SPEA2 is applied to R_t [4].

Step 3: Using the assigned matching degree and the truncation process,* the same number of individuals as the number of individuals in the archive set are selected from R_t to form the new set of archive individuals A_{t+1}.

On the other hand, mating selection,† which is used in SPEA2 and NSGA-II to select the set of search individuals (P_t) from the set of archive individuals (A_t), is not applied. Mating selection selects especially superior individuals in duplication from the set of archive individuals. This process is not used in NCGA, since there can be an adverse effect on neighborhood crossover if there the same individuals occur in duplication. In NCGA, the copy of the set of archive individuals (A_t) is used as the set of search individuals (P_t).

5. Rectangular Packing Problem

The rectangular packing problem is applicable to a wide range of activities, such as the floor planning, which aims at minimization of the chip area in large-scale integrated circuits (LSI) [6, 7], and the placement of installations and offices in a plant [8]. The rectangular packing problem is a kind of combinatorial optimization problem, and has the property that the number of possible combinations increases exponentially with the number of blocks to be handled.

In this paper, the rectangular packing problem is not handled as the simple minimization of the area, but as the multi-objective optimization problem in which the vertical and horizontal lengths of the finally obtained placement are considered as the objectives. The purpose is to let the solution selector select the aspect ratio of the area for the block placement to be finally realized. This section describes the representation of the data structure, which is closely related to the definition of the problem, and also the formulation of the problem.

5.1. Representation of data structure

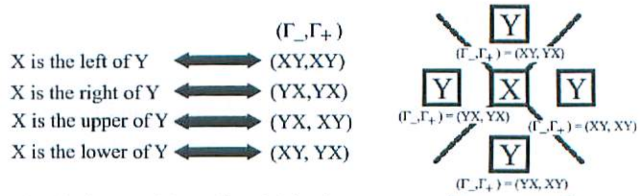
One of the most important aspects of the rectangular packing problem is the representation of the solution, that is, the representation of the data structure. The representation of the data structure in the rectangular packing problem affects not only the block placement configuration, but also such essential aspects of the problem as the accuracy of the finally obtained solution.

Various approaches to this aspect of the problem have been presented [6–10]. By the use of the sequence-pair [7, 9] and BSG [10], which were developed in LSI floor planning, it is possible to derive the optimal packing in a finite solution space. This paper uses the sequence-pair, which can realize a more efficient solution search than BSG. The mechanism of the sequence pair is described in detail below.

Sequence-pair

The sequence-pair is used to represent the rectangular placement based on the arrangement (Γ_-, Γ_+) of two sequences composed of the blocks under consideration. The sequence-pair specifies the relative position relation between any two blocks, based on the permutation of (Γ_-, Γ_+). Figure 1 shows the conceptual diagram of the sequence-pair.

Figure 1(a) shows the permutation of two arbitrary blocks X and Y in (Γ_-, Γ_+), and the corresponding position



(a) relative position of each block (b) concept figure

Fig. 1. Concept of sequence-pair.

*This is the reduction process for the excess nondominated individuals used in SPEA2 [4].

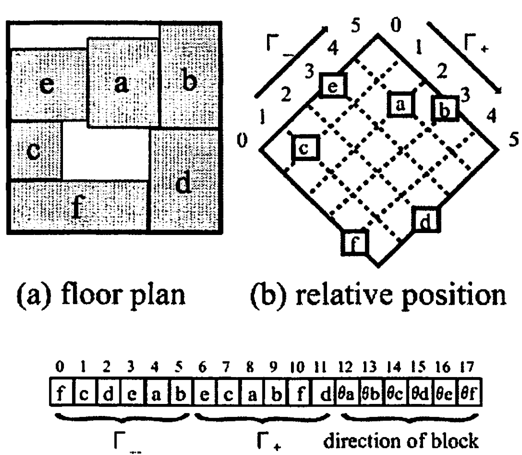
†The selection process used to select the set of search individuals for genetic operation is called mating selection. In mating selection, the set of search individuals is formed by duplicate selecting of the especially superior individuals from the set of archive individuals in order to achieve higher search efficiency.

relation of blocks X and Y. That is, the block permutation $(\Gamma_-, \Gamma_+) = (XY, XY)$ indicates that X precedes Y in Γ_- and Γ_+. Such an order implies that X is placed to the left of Y. Basically, the Γ_- axis represents the vertical relation between the blocks, and the Γ_+ axis represents the horizontal relation between the blocks. Thus, Γ_- determines the placement from lower to the upper side, and Γ_+ determines the placement from left to right side. Visually, Fig. 1(a) indicates the placement for (a).

In the sequence-pair representation, for any permutation (Γ_-, Γ_+), there is a rectangular placement that satisfies the indicated position relation. Conversely, for any rectangular placement, there is a permutation (Γ_-, Γ_+) that represents the placement [7]. Thus, by using the sequence-pair, any rectangular placement can be represented, including a nonslicing structure.

Figure 2 shows an example of the sequence-pair for the case of six blocks. The example of the individuals in Fig. 2(c) represents the permutation (Γ_-, Γ_+), and also the orientation θ of each block. In this paper, the orientation of the block in placement is restricted to either vertical or horizontal. Figure 2(b) representing the relative position of the blocks is constructed from the permutation (Γ_-, Γ_+) in Fig. 2(c).

Based on Fig. 2(b) representing the relative position, the relative positions are determined for all blocks based on Fig. 1, and the vertical and horizontal constraint graphs in Fig. 3 are constructed. In the vertical and horizontal constraint blocks, the width and height of the block are given as the weight of the point. Then the longest route is determined by calculating the sum of the weights (block lengths) of the points from the start to the end points. By the above weight calculation, the placement diagram of the blocks in Fig. 2(a) is determined.



(a) floor plan (b) relative position

(c) coding of solution

Fig. 2. Coding example of sequence-pair.

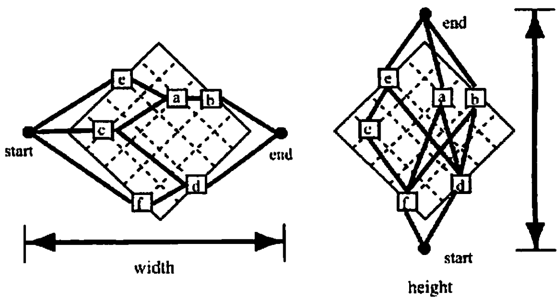


Fig. 3. Horizontal/vertical constraint graphs.

5.2. Problem formulation

The objectives are defined as the following two in the numerical experiment:

$$\min f_1(x) = width \tag{1}$$

$$\min f_2(x) = height \tag{2}$$

In the above expression, *width* is the horizontal length, and *height* is the vertical length.

The above two objectives have an explicit trade-off relation, but the area occupied by the block is minimized by performing optimization for the two objectives. By multi-objective processing, the solution selector can flexibly decide the aspect ratio of the packing area. After completing the block placement, the width and height of the whole region can be derived from the vertical and horizontal constraint graphs in Fig. 3.

6. Numerical Experiment

The effectiveness of the proposed NCGA is investigated in this study for the rectangular packing problem described in the previous section. For comparison with NCGA, SPEA2 proposed by Zitzler and colleagues, NSGA-II proposed by Deb and colleagues, and NCGA without neighborhood crossover (non-NCGA) are also investigated.

6.1. Solution by GA

6.1.1. Crossover

In the proposed method, placement-based partially exchanging crossover (PPEX) [9] is used as the crossover process for the rectangular packing problem. This method is a procedure based on the relative placement of the sequence-pairs, making it possible to apply local crossover to blocks in close placement.

The PPEX algorithm is as follows.

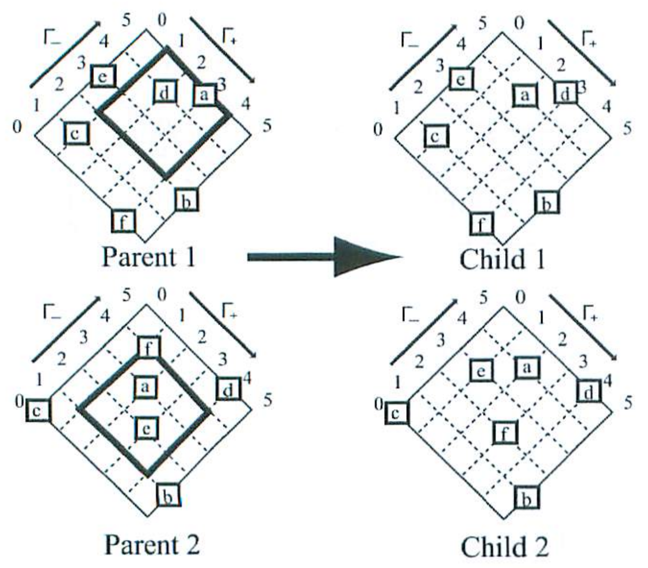


Fig. 4. PPEX.

Step 1: One block is selected at random from each of the parents.

Step 2: The window region is formed with the selected block as the center. The set of blocks contained in the window is defined as the crossover target M_c. The other set of blocks is denoted as M_{nc}. The blocks of M_c are sorted following the order of (Γ_-, Γ_+), and the result is copied to the child.

Step 3: Each block of M_{nc} is directly copied to the child.

The window region mentioned above is the continuous partial region on the oblique grid[*] representing the relative placement of the sequence-pairs. In this paper, the window region is handled as a square partial region.

Figure 4 shows an example of PPEX, where the edge length of the window region is set as 4. In this case, blocks *a* and *b* contained in the randomly generated window region for parent 1 are the target block of crossover (M_c). Before crossover, the order was (Γ_-, Γ_+) = (da, da), but the order is modified to the order of parent 2, that is, (Γ_-, Γ_+) = (ad, ad) after crossover. The blocks other than the target of crossover (M_{nc}) are directly copied from parent 1 to child 1. The orientation is also modified to the orientation of the mate parent.

6.1.2. Mutation

As mutation, the orientation of a randomly selected block is reversed.

[*]Basically the oblique grid, see Fig. 2(b).

6.2. Sample problems

In our study, NCGA was applied to four different problems (ami33, ami49, pcb146, pcb500) with various numbers of blocks, and the performance was compared to the conventional methods. ami33 and ami49 are benchmark problems for MCNC, composed of 33 and 49 blocks, respectively. pcb146 and pcb500 are sample problems used by Murata and colleagues, composed of 146 and 500 blocks, respectively [7]. In this paper, only the results for ami33 and pcb500 are shown, due to space limitations.

In the experiment, four methods are used, namely, SPEA2 proposed by Zitzler and colleagues [4], NSGA-II proposed by Deb and colleagues [3], and NCGA proposed by us, plus non-NCGA without neighborhood crossover (non-NCGA).[*] By comparing NCGA and non-NCGA, the effect of search using neighborhood crossover is clearly indicated.

6.3. Evaluation of obtained solution candidate

In this study, I_{LI}, which samples the intersection with the approximate Pareto front, is used as a means of evaluation of the set of solutions obtained by the search. The distribution diagram of the solution set is also used in order to show visually the distribution range of the solution set. Below, I_{LI}, which samples the intersection with the approximate Pareto front, is described.

Intersection I_{LI} with approximate Pareto front

This is a technique proposed by Knowels and Corne to compare two or more nondominated solution sets [11].[†] In this method, nondominated solution sets are compared in terms of dominance along multiple directional vector axes, and the degree to which the nondominated solution dominates is decided. Figure 5 shows a conceptual diagram of the method when two nondominated solution sets (*X*, *Y*) are compared in the two-objective problem.

In this method, the arrival front (dotted line in Fig. 5) formed by the obtained set of nondominated solutions is calculated. The sampling line is defined so that the Pareto region is uniformly sampled. The intersections of the sampling line and the arrival front of the nondominated solution are determined and compared. When there are two sets (*X*, *Y*) to be compared, $I_{LI}(X, Y)$ represents the percentage of cases in which *X* dominates *Y* in the comparison of intersections. Consequently, the best result for *X* is that $I_{LI}(X, Y) = 1$ and $I_{LI}(Y, X) = 0$.

[*]Ordinary crossover is applied in non-NCGA.

[†]A nondominated solution is a solution which is not dominated by any solution candidate obtained by the search. The set of solutions obtained by the search are called nondominated solutions.

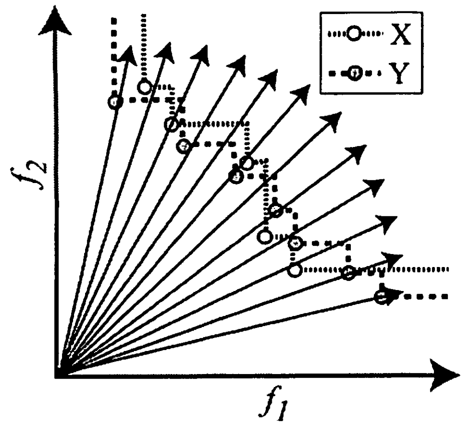


Fig. 5. Schematic of I_U.

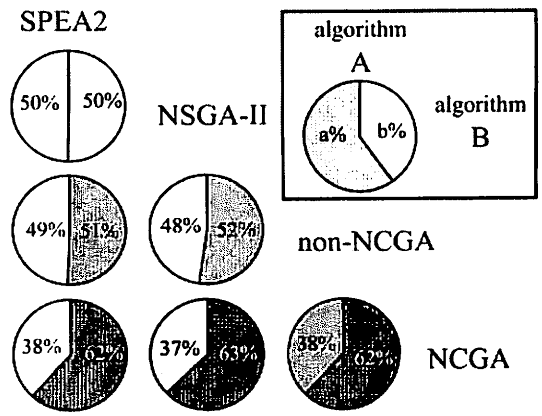


Fig. 6. Results of I_U (ami33).

By applying this process, the obtained nondominated solution for the Pareto optimal solution can be evaluated from the two viewpoints of width and accuracy. For problems with three or more objectives, however, this method is difficult to apply, since the formation of the arrival front becomes difficult.

6.4. GA parameters

Table 1 shows the GA parameters used in this experiment.

6.5. Numerical results

For the four sample problems, the numerical results for ami33 and pcb500 are shown in this paper. The four methods NCGA, SPEA2, NSGA-2, and non-NCGA were used. Each of these methods is applied to each sample problem 30 times. The results for these problems and the placement diagram for ami33 obtained by NCGA are presented below.

Table 1. GA parameters

number of blocks	33, 50, 100, 500
population size	200
crossover rate	1.0
mutation rate	1/bit length
terminal generation	400
number of trials	30

6.5.1. ami33

The result for I_U for ami33 is shown in Fig. 6. Figure 7 shows the plot of the set of nondominated solutions obtained by each method. In Fig. 7, the set of nondominated solutions is summarized in the plot.

As can be seen from Figs. 6 and 7, NCGA can derive a wide range of solutions closer to the Pareto optimal solution than any other method. In particular, when the vertical and horizontal lengths of the rectangle differ greatly [$f_1(x) < 750, f_1(x) < 2500$ in Fig. 7], NCGA can search the range for Pareto optimal solutions, which cannot be achieved by other methods. non-NCGA, on the other hand, derives a wider range of nondominated solutions than SPEA2 and NSGA-II, but is worse than NCGA in terms of the closeness to the Pareto optimal solution.

Thus, neighborhood crossover in NCGA is a very effective method of search in the rectangular packing problem.

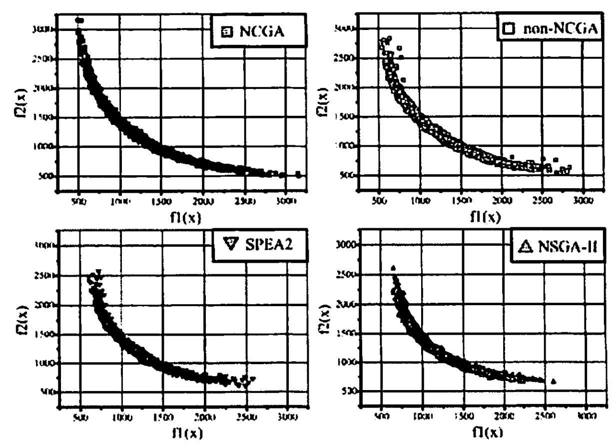


Fig. 7. Nondominated solutions (ami33).

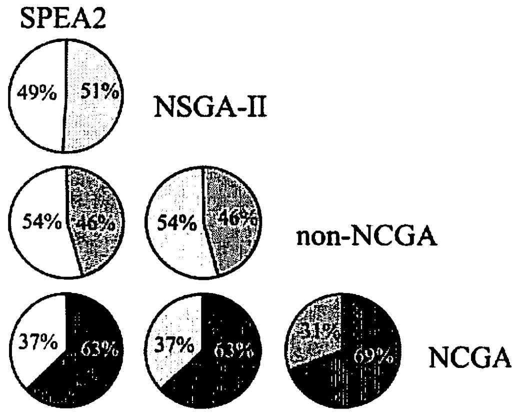


Fig. 8. Results of I_U (pcb500).

6.5.2. pcb500

The result for I_U for pcb500 is shown in Fig. 8. Figure 9 shows the plot of the set of nondominated solutions obtained by each method. It is seen from Figs. 8 and 9 that NCGA can derive better solutions than other methods, as in the case of ami33. However, observing in detail the region which is close to a square in Fig. 9 ($1400 < f_1(x) < 1600$), better solutions are obtained by SPEA2 and NSGA-II than by NCGA, although the difference is only slight.

It can also be seen that SPEA2 and NSGA-II give a solution distribution that is extremely concentrated near the center, compared to NCGA and non-NCGA. The reason seems to be that pcb500 is a difficult problem with large scale, making the search concentrate in a region close to the square, for which better solutions are easily obtained in the early stage. In pcb500, the number of possible permutations is more than 10^{2000} times as great as that of ami33. Because of this scale difference, the search in NSGA-II and SPEA2

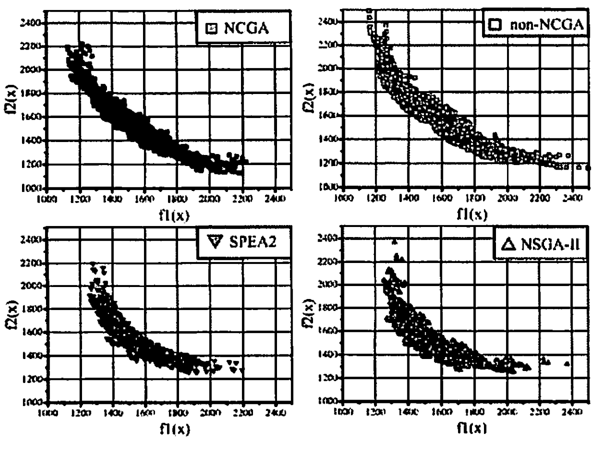


Fig. 9. Nondominated solutions (pcb500).

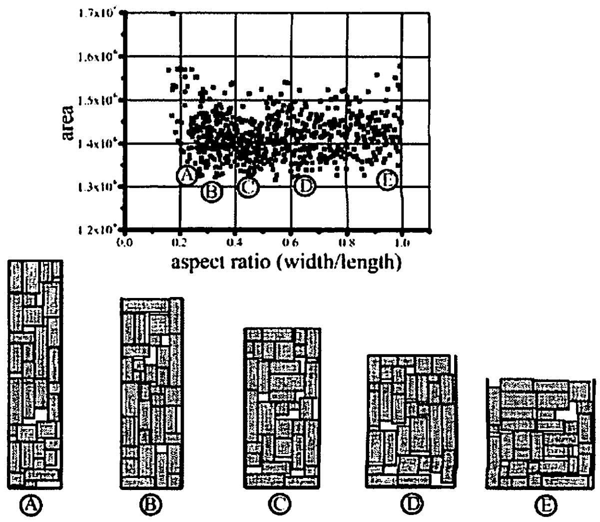


Fig. 10. The placement of the blocks (ami33).

concentrates in a region that is close to a square and better results are obtained for such a region, compared to NCGA and non-NCGA, which search for the solution over a wider range.

It should be noted that NCGA attempts a very wide range of search even for problems in which search is difficult, while providing nondominated solutions, which is only slightly worse in the square region in which the search is concentrated in NSGA and SPEA2. Comparing NCGA and non-NCGA, non-NCGA is more markedly worse than in ami33. Thus, neighborhood crossover is more effective in highly difficult problems with a wider search region.

6.5.3. Placement diagram

The above experimental results show that NCGA derives the best solution in almost all cases. Thus, the validity of the placement diagram for the nondominant solution obtained by NCGA is investigated in this study.

Figure 10 shows a plot of NCGA for ami33, together with placement diagrams corresponding to several plot points. In the figure, the axes are not the vertical and horizontal lengths of the area, but the area and aspect ratio. The reason is that the solution selector at the final stage will select the solution balancing the area and the aspect ratio.

It can be seen from the figure that almost all solutions give adequate placements without gaps. It is also seen that solutions are obtained for a wide range of aspect ratios from 0 to 1. It is evident from the placement diagrams in the figure that the area of placement remains almost constant from "A" to "E," regardless of the aspect ratio. It is inferred that there is no strong effect of the aspect ratio on the gaps in the placement diagram (i.e., the total area for wiring).

7. Conclusions

This paper has considered the rectangular packing problem, which occurs in office placement problems and in chip area minimization problems in large-scale integrated circuits (LSI), and has presented a method of multi-objective optimization by a genetic algorithm. In this paper, the problem is formulated as a two-objective optimization problem, with the height and width of the overall placement as the objectives. The sequence-pair is applied to placement representation, PPEX is applied to crossover, and NCGA is applied as the genetic algorithm.

Among four sample problems for numerical experiment, the application of NCGA is shown for two problems, ami33 and pcb500, and the results are compared to those of the conventional method. As targets of comparison, SPEA2 and NSGA-II, which are typical procedures in multi-objective GA, are considered in this paper. In order to investigate the effectiveness of neighborhood crossover in the rectangular packing problem, NCGA and non-NCGA without neighborhood crossover are also compared.

The following facts were revealed by numerical experiments.

- In all sample problems, NCGA gave better results than the other methods. Thus, NCGA is an effective method in the multi-objective rectangular packing problem.
- Comparing the cases with and without neighborhood crossover, the former gave much better results in all problems. Thus, neighborhood crossover is effective in the search in the multi-objective rectangular packing problem.
- For complex problems with a large number of blocks, there is a tendency in SPEA2 and NSGA-II for the solutions to concentrate to the middle range of the Pareto front. In contrast, diverse solutions can be derived in NCGA and non-NCGA. This indicates the effectiveness of NCGA in retaining diversity, regardless of the problem scale.
- Examining the placement diagrams for the solution obtained by NCGA, it is seen that reasonable solutions with few gaps are obtained. Thus, NCGA is an effective method for the rectangular packing problem.

Acknowledgment. This study was performed as a part of the Doshisha University Science Frontier Research Project "Intelligent Information Science and Its Application," based on the Science Frontier Promotion Plan of the Ministry of Education and Science. The authors are grateful to those concerned.

REFERENCES

1. Schaffer J. Multiple objective optimization with vector evaluated genetic algorithms. Proc 1st Int Conference on Genetic Algorithms and Their Applications, p 93–100, 1985.
2. Deb K. Multi-objective optimization using evolutionary algorithms. Wiley; 2001.
3. Deb K, Agrawal S, Pratab A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL Rep 200001, Indian Institute of Technology, Kanpur, 2000.
4. Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the performance of the strength Pareto evolutionary algorithm. Tech Rep 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 2001.
5. Watanabe S, Hiroyasu T, Miki M. Multi-objective optimization by neighborhood cultivation genetic algorithm. Trans Inf Process Soc, "Mathematical modeling and applications," 2002;43:183–198.
6. Nojima T, Sakanushi K, Takahashi A, Kajitani H. Placement using sequence-pair guaranteeing connections. Tech Rep IEICE 2001;VLD2001-54.
7. Murata H, Fujiyoshi K, Nakatake S, Kajitani Y. VLSI module placement based on rectangle-packing by the sequence-pair. IEEE Trans Comput Aided Des Integr Circuits Syst 1996;15:1518–1524.
8. Shirai Y, Matsumoto N. Performance evaluation of ES type genetic algorithms for solving block layout problems with floor constraints. Trans JSME (C) 1999;65:296–304.
9. Nakaya S, Wakabayashi S, Koide T. Floor-planning technique based on adaptive genetic algorithm and sequence-pair. Tech Rep Inf Process Soc SLDM93-16, 1999;99:119–126.
10. Nakatake S, Murata H, Fujiyoshi K, Kajitani Y. Module placement on bsg-structure and ic layout applications. Proc Int Conference on Computer Aided Design '96, p 484–491.
11. Knowles J, Corne D. Approximating the nondominated from using the Pareto archived evolution strategy. Evol Comput 2000;8:149–172.

AUTHORS (from left to right)

Shinya Watanabe received a Ph.D. degree from Doshisha University in 2003 and became a lecturer at the College of Information Science and Engineering, Ritsumeikan University, in 2004. Since 2007, he has been a lecturer in the Department of Computer Science and Systems Engineering, Muroran Institute of Technology. His current research interests focus on evolutionary multi-objective optimization, machine learning, and data mining. He is a member of IEEE, IPSJ, and ISCIE.

Tomoyuki Hiroyasu (member) received a D.Eng. degree from Waseda University in 1998. He has been an associate professor in the Department of Intelligent Information Systems Engineering, Doshisha University, since 2003. His research interests include the development of novel evolutionary algorithms, parallel computing, grid, and intelligent systems. He is a member of IEEE, JSME, JSCES, IPSJ, SICE, and IEICE.

Mitsunori Miki received a Ph.D. degree from Osaka City University in 1978. He has been a professor in the Department of Computer Science, Doshisha University, since 1994. His major research areas are optimization, parallel processing, and intelligent systems. He is the leader of the Academic Frontier, Intelligent Information Science (AFIIS) project at Doshisha University sponsored by the Ministry of Education, Culture, Sports, Science and Technology, and a member of IEEE, AIAA, IPSJ, JSAI, JSME, and ISCIE.