Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems

Alev Sokea, Zafer Bingulb,*

aElectronic and Computer Education, University of Kocaeli, 41380 Kocaeli, Turkey
bDepartment of Mechatronics Engineering, University of Kocaeli, 41300 Kocaeli, Turkey

Received 30 December 2004; received in revised form 21 December 2005; accepted 24 December 2005
Available online 3 March 2006

Abstract

In this paper, genetic algorithm (GA) and simulated annealing (SA) with improved bottom left (BL) algorithm were applied to two-dimensional non-guillotine rectangular packing problems. The performance and efficiency of these algorithms on several test problems [Hopper, E., Turton, B.C.H., 2000. An empirical investigation of meta-heuristic and heuristic algorithms for two-dimensional packing problem. European Journal of Operational Research 128 (1), 34–57] were compared. These test problems consist of 17 and 29 individual rectangular pieces to place in main object limited with a size of $200 \times 200$ units. Both solution approaches were compared, based on the trim losses of the test problems. Also, the influences of the GA parameters (population sizes, mutation rates, crossover techniques) and of the SA parameters (cooling schedules, neighborhood moves, the number of inner loop, different temperature values) on the solution of these problems were examined. For considering all solutions of the test problems, the hybrid GA produces much better results than the hybrid SA.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Genetic algorithm; Improved bottom left algorithm; Simulated annealing; Two-dimensional non-guillotine rectangular packing problems

1. Introduction

Cutting and packing problems have a very similar structure with regard to their solution approaches. They are encountered in many different industries, incorporating different constraints and objectives. To a great extent, the wood, glass and paper industries are involved the cutting of regular figures, metal, textile and leather industries are concerned with the cutting of arbitrary-shaped figures.

The two-dimensional non-guillotine rectangular cutting problems consist of placing rectangular pieces of predetermined sizes into a large but finite rectangular main object or equivalently cutting rectangular pieces from rectangular main object. These are non-guillotine cutting problems since the “cuts” may not go from one end to another (Leung et al., 2001). These problems have been shown to be non-deterministic polynomial (NP) complete (Dowsland and Dowsland, 1992), in that it is impossible to optimally solve any of these problems, except in trivial cases (Callahgan et al., 1999). The objective function of NP complete problems is often multi-modal, non-smooth or even discontinuous, which means that traditional, gradient-based optimization algorithms fail. Finding a solution requires an organized search through the parameter space. An unguided search is extremely inefficient since many of these problems are NP complete (Callahgan et al., 1999). As a result, research on cutting and packing problems has focused on stochastic search techniques such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), naïve evolution (NE) and population heuristic (PH). These techniques are combined with a placement algorithm (decoding algorithm) for solution of cutting and packing problems. Therefore, this combined algorithm is referred to as the “hybrid approach” in the literature. In recent decades, non-guillotine cutting and packing problems with evolutionary methods have received considerable attention and have undergone rapid development. Some studies related to these problems in literature are given below:
GA-based methods, SA-based methods, TS-based methods, NE-based methods and PH-based methods.

For non-guillotine bin-packing problems, Smith (1985) developed order-based a GA approach using two heuristic decoding algorithms (slide algorithm and skyline algorithm). These hybrid approaches generate better layout patterns but they are computationally more expensive than other packing methods based on heuristics and dynamic programming. Jakobs (1996) developed an order-based GA with bottom left (BL) algorithm to place rectangular pieces and polygons onto a rectangular main object. However, GA in this study was combined with embedding a shrinking algorithm to place of polygons. Dagli and Poshyanonda (1997) developed two approaches for the solution of the rectangular stock-cutting problem: neuro-optimization and genetic neuro-nesting. In the neuro-optimization approach, the results show that the highest and the lowest percentages of trim loss values produced by the method are 16.42% and 0.70%, respectively. Genetic neuro-nesting approach generated layouts having packing densities in the range 95–97%. Liu and Teng (1999) developed a GA with improved BL algorithm for the orthogonal packing of rectangular pieces. The result of this study shows that the new version of the BL algorithm is more effective than proposed by Jakobs (1996). Hopper and Turton (1999), studied two GA hybridized with two placement algorithms (BL algorithm and bottom left fill (BLF) algorithm) for rectangular packing problems. The results of this study show that the difference between the two hybrids GA is less than expected. Since the performance difference between two hybrid approaches is only due to the improved heuristics, the decoder has a larger effect on the outcome of the hybrid technique than the GA. In order to improve the results, the GA encoding schemes need to incorporate more layout specific knowledge. Leung et al. (1999) applied the GA and SA methods with difference process (DP) algorithms to the two-dimensional non-guillotine cutting stock problem. In their study, the trim loss values in GA with DP algorithm range from 4.5% to 5.5% and the trim loss values in SA with DP algorithm range from 7% to 10%. The run times of the GA are shorter than the run times of the SA. Thus, the GA outperforms the SA for cutting stock problems.

Kämpke (1988) worked on SA and packing problems and compared different cooling strategies in SA. Dowsland (1993) applied the SA approach to packing problems, which consist of identical and non-identical boxes. The results of this study show that SA is only capable of producing near optimal solutions, which could be obtained by other optimization methods. Faina (1999) developed a SA approach to solve a two-dimensional cutting problems involving both guillotine and non-guillotine constraints. In his study, the algorithm used for the non-guillotine layout pattern accomplishes much higher packing densities than algorithms for the guillotine layout pattern within approximately the same computational time. Hopper and Turton (2000) studied three meta-heuristic algorithms (GA, SA, NE) and local search heuristic combined with placement algorithms (BL, BLF) for the two-dimensional rectangular packing problem. In their study, hybrid approaches were compared in terms of solution quality and computation time on a number of packing problems of different size. According to this study, SA achieved the best layout quality over all problem categories but its run time becomes larger with increasing problem size. The evolutionary algorithms, GA and NE are better in terms of run time and accomplishing results, which are slightly worse than those obtained by the SA. Leung et al. (2003) applied a pure meta-heuristic (GA) and a mixed meta-heuristic (SA–GA) to two-dimensional orthogonal packing problems. In their work, the DP algorithm was used to obtain a layout pattern and 19 cutting problems were tested. In this work, SA decides which two parents and children should be selected after crossover and mutation. This method usually produces better results in long run.

Blążewicz et al. (1993), Blążewicz and Walkowiak (1995) applied TS to irregular packing problems and two-dimensional irregular cutting problems with three different approaches. The first approach is a simple short-term memory version of TS that incorporates a tabu list, an objective function and a single criterion for optimization. The second approach introduces a new type of evaluating function, which combines several criteria to be optimized. In the last version, a probabilistic approach is used which translates the evaluation criteria into probabilities of selection. In comparison with Albano and Sapuppo’s Albano and Sappupo (1980) heuristic search algorithm, the TS achieved better results. For this approach and other variations of irregular packing problems, detail information can be found in (Blążewicz et al., 2002, 2004). Lodi et al. (1999) studied the TS for two-dimensional rectangular bin-packing problems.

Falkenauer (1998) applied NE algorithm to one-dimensional bin packing problems. In his work, NE was used entirely to establish the performance of the crossover operators implemented in various approaches to strip and bin-packing problems. Hopper and Turton (2000) compared the performance of GA, NE and SA with each other for two-dimensional rectangular packing problems. The algorithms used in the study are hybridized with two different placement algorithms (BL, BLF). The results of the two evolutionary methods (GA and NE) are very similar with NE algorithm performing slightly better for some problems. The best layouts for the hybrid algorithms have been obtained with SA in all problems. The difference between SA and the two evolutionary methods (GA and NE) lies between 1% and 8% and it is higher for the larger problems.

Beasley (2004) presented a heuristic algorithm for the constrained two-dimensional non-guillotine cutting problems. The heuristic algorithm is called population heuristic, which explicitly works with a population of solutions and combine them together in some way to generate new solutions. More information relating to
population heuristic can be found in (Beasley et al., 2003; Pinol and Beasley, 2006). The algorithm is based on a new non-linear formulation of the problem. Computational results denoted that PH algorithm could effectively solve for typical test problems taken from the literature and a number of large randomly generated problems. As problem size increases, the PH algorithm gets results that are closer to optimality. The algorithm can routinely solve problems that cannot be tackled using the best optimal procedure currently known. There exist problems for which previous heuristics can never find the optimal solution the heuristic presented in this paper has no such limitations.

The objective of the cutting and packing problems and descriptions of placement algorithms are presented and improved BL algorithm is illustrated in Section 2. Section 3 and Section 4 introduces the definition and properties of GA and SA, respectively. Methodology is outlined in Section 5. Solution approaches of GA and SA, and experimental results are presented in Section 6. Finally, a summary on the performance of each hybrid approach is provided in Section 7.

2. Placement algorithms

The objective of the cutting and packing problems is to increase the usability of main object and obtain the layout pattern with the minimum trim loss value. The trim loss is the unused area in the main object. Solution approaches of the cutting and packing problems can be divided into two sections: determination of permutation for pieces sequences and applying placement algorithm for placing all pieces. A cutting pattern or a layout pattern represented by a permutation corresponds to the sequences in which the rectangles are to place into main object. The placement algorithm using this permutation is then executed for placing of rectangular pieces into main object.

The aim of a placement algorithm is that a rectangular piece cannot be moved any further to the left and downwards without collision. There are many placement algorithms based on sliding techniques such as DP algorithm (Leung et al., 1999), BL algorithm (Jakobs, 1996), improved BL algorithm Liu and Teng (1999) and BLF algorithm. The DP algorithm can access enclosed areas in the partial layout pattern. After every new piece is placed in the layout pattern, two empty rectangular spaces rise at the top and right side of the placed piece. The algorithm keeps track of these spaces for the allocation of the next rectangular piece. In the BL algorithm, a rectangular piece is initially placed at the right upper corner of the main object. Every piece in the permutation is moved down as far as possible and then to the left until no piece is moved further. The BL algorithm a limitation in that some patterns cannot be obtained. To overcome this, the new version of the BL, called the improved BL algorithm, was developed by Liu and Teng (1999). Like a BL algorithm, the rectangular piece is moved to bottom and left as far as possible until it can go no further. The difference between the BL algorithm and the improved BL algorithm is that a rectangular piece in the improved BL algorithm is moved with checking its lower empty space. The BLF algorithm, developed by Hopper and Turton (2000), attempts to place a rectangular piece into the lowest available position of the main object and left-justifying it. Compared with the BLF algorithm and BL algorithm, the BLF algorithm has denser layout patterns. The major disadvantage, however, lies in its time complexity.

A layout pattern can be represented by a permutation \( \pi \) as shown: \( \pi = (i_1, \ldots, i_n) \), \( i \): index of rectangle \( (r_i) \). The allocation of some of rectangular pieces was illustrated in Fig. 1. In Fig. 1, piece \( r_3 \) is placed first to the bottom and then to the left as far as possible. Pieces \( r_2 \) and \( r_1 \) are placed in the same manner. The arrows in the figure depict the movement of the piece \( r_4 \) to its optimal location.

The number of possible layout patterns allocated by a BL algorithm is \( 2^n \cdot n! \) for given the number of \( n \) pieces. The number of solutions for the cutting problems becomes very high as the number of pieces increases. Therefore, it requires very large search space, rendering the cutting problems very difficult to solve.

3. Genetic algorithm

GA, one of the optimization and global search methods, is based on Darwin’s theory of evolution and simulated natural selection (Goldberg, 1989). GA was developed further by Holland in the 1970s. It is applied effectively to solve various combinatorial optimization problems and worked with probabilistic rules (Holland, 1975). Selection, crossover and mutation are the most known genetic operators.

GA searches new and better solutions to a problem by improving current population. The search is guided towards the principle known as survival of the fittest. This is obtained by extracting the most desirable characteristics from a generation and combining them to form the next generation. The population comprises a set of
chromosomes. Each chromosome in the population is a possible solution and the quality of each possible solution is measured by fitness function. First, GA generates initial population and then calculates the fitness value with fitness function for each chromosome of the population. Fitness function is specifically generated for each problem. It may be simple or complex according to problem. Then optimization criterion is checked. If optimization criteria are met, this solution can be considered as the best solution. Otherwise, new population is regenerated using GA operators (selection, crossover, and mutation). According to their fitness values, chromosomes are selected for crossover operation using any selection operator. Therefore each chromosome will contribute to the next generation in proportion to its fitness. Then crossover and mutation operation are applied to the selected population to create the next population. The process is to continue through number of generations until convergence on optimal or near-optimal solutions. However GA cannot guarantee to find an optimal solution.

GA operators are described as follows:

**Population:** It is a set of possible solution of the problem. Because size of the population varies as to problem, there is no clear mark how large it should be. In our experiments, the population size ranges from 20 to 100.

**Fitness function:** A fitness value for each chromosome of population is calculated according to fitness function as defined in Eqs. (1) and (2). The trim loss value, which is formulated in Eq. (2), represents proportion of unused area in main object. Because the aim of these problems is obtained minimum trim loss value, fitness function is created in Eq. (1).

\[
\text{Fitness value}(\pi) = \frac{1}{(\varepsilon + \text{Trimlossvalue}(\pi))}, \quad (1)
\]

\[
\text{Trimlossvalue} = \frac{\text{area of the main object} - \text{sum of areas of placed pieces}}{\text{area of the main object}}, \quad (2)
\]

where \(\varepsilon = 2.2204 \times 10^{-16}\). If the trim loss value is equal to zero, this small error in fitness function prevents the fitness value from tending to infinity.

**Selection:** Selection operator selects to be mate chromosomes according to their fitness values. *Roulette wheel selection* operator is used here. First, a fitness value is calculated for each chromosome in population. The fitness value of a chromosome is divided by the total fitness value and obtained its relative fitness value. Then two chromosomes are chosen to form the next generation according to their relative fitness value (Lindfield and Penny, 1995).

**Crossover:** Crossover operator is powerful for exchanging information between chromosomes and creating new solutions. It is hoped that good parents may produce good solutions. Six different chromosome techniques are tested here: order-based crossover (OBX), cycle crossover (CX), order crossover (OX), partially matched crossover (PMX), uniform crossover (UX) and Stefan Jakobs crossover (SJX) (Leung et al., 2001; Jakobs, 1996; Beasley, 2004; Lindfield and Penny, 1995; Houck et al., 1996).

**Mutation:** This operator is used to prevent reproduction of similar type chromosomes in population. Mutation operator randomly selects two genes in chromosome and swaps the positions of these genes to produce a new chromosome (Leung et al., 2001). This technique is called swap mutation.

### 4. Simulated annealing

SA, firstly developed by Kirkpatrick et al. (1983), is a local search algorithm. It is based on the analogy between the process of finding a possible best solution of a combinatorial optimization problem and the annealing process of a solid to its minimum energy state in statistical physics.

The searching process starts with one initial random solution. A neighborhood of this solution is generated using any neighborhood move rule and then the cost between neighborhood solution and current solution can be found with Eq. (3).

\[
\Delta C = C_i - C_{i-1}, \quad (3)
\]

where \(\Delta C\) represents change amount between costs of the two solutions. \(C_i\) and \(C_{i-1}\) represents neighborhood solution and current solution, respectively. If the cost decreases, the current solution is replaced by the generated neighborhood solution. Otherwise the current solution is replaced by the generated neighborhood solution by a specific possibility calculated in Eq. (4) or a new neighborhood solution is regenerated and steps are repeated until this step. After new solution is accepted, inner loop is checked. If the inner loop criterion is met, the value of temperature is decreased using by predefined a cooling schedule. Otherwise a new neighborhood solution is regenerated and steps are repeated until this step. The searching is repeated until the termination criteria are met or no further improvement can be found in the neighborhood of the current solution. The termination criterion (outer loop) is predetermined.

\[
e^{(-\Delta C/T)} > R, \quad (4)
\]

where \(T\) temperature is a positive control parameter. \(R\) is a uniform random number between 0 and 1.

SA operators are described as follows:

**Cost function:** The value of cost function for a possible solution is calculated according to equality as defined in Eqs. (1) and (2). In this work, the change in the cost function represents the change in the trim loss value of the layout pattern.

**Neighborhood move:** This operator is used to produce a near solution to current solution in search space. In this study, two neighborhood moves were basically employed: swapping move and shifting move (Leung et al., 2001). In
swapping move, randomly selects two genes in solution and swaps the positions of these genes. Then a neighborhood solution is produced. In shifting move, randomly selects two genes in solution and second gene is puts in front of another genes. Thus a new solution is obtained.

Cooling schedule: The performance of this algorithm is dependant on this operator. Two different cooling schedules are tested in study. These schedules are proportional decrement schedule and Lundy and Mees schedule (Lundy and Teng, 1986). In proportional decrement schedule, two temperature values, which are in \( k \)th and \( k + 1 \)th iterations, are related by Eq. (5).

\[
T_{k+1} = \alpha T_k, \quad \alpha = \sqrt{\frac{T_f}{T_i}},
\]

where \( T_k \) and \( T_{k+1} \) are temperature in \( k \)th and \( k + 1 \)th iterations respectively. \( \alpha \) is coefficient between two temperatures and varies between 0 and 1. \( M, T_f\) and \( T_i \) are number of iteration, the final temperature and the initial temperature, respectively.

In Lundy and Mees schedule; the relationship between \( T_{k+1} \) and \( T_k \) is below:

\[
T_{k+1} = \frac{T_k}{1 + \beta T_k}, \quad \beta = \frac{T_i - T_f}{MT_i T_f},
\]

where \( \beta > 0 \) is coefficient between two temperatures: \( T_{k+1} \) and \( T_k \). In this study, while initial temperature is varying between 0.1 and 0.8, final temperature is 0.01.

Inner loop and outer loop criterion: Inner loop criterion decides how many possible new solutions to produce in every temperature. Outer loop criterion is used to stop the searching process. In this study, inner loop criterion is set to 3, 5, and 10. Outer loop criterion is number of 1000.

5. Methodology

In this paper, GA and SA were used separately to obtain permutation for placing the small pieces. Improved BL algorithm was employed to place rectangular pieces.

The solution approach in this study can be summarized below:

1. GA and SA were used to find permutations with small trim loss.
2. An improved BL algorithm was used to place rectangular pieces corresponding to a particular permutation.

These rules are taken into consideration, while rectangular pieces are placed (Leung et al., 2001):

1. All pieces have fixed orientation.
2. There is no rotation when placing the pieces into the main object.
3. The length and width of each piece does not exceed dimensions of the main object.
4. The dimensions of the pieces and main object are integer.
5. The edges of the pieces do not occupy any area, when placing the pieces into the main object.
6. There is no restriction; each piece may be positioned at any place in the main object.

6. Experimental results

We have implemented simulations to test the performance and compare efficiency of the hybrid GA and hybrid SA onto various two-dimensional non-guillotine rectangular packing problems (Soke and Bingul, 2004). In these problems, six different test problems were chosen as examples. The dimensions of main object are limited with a size of 200 \( \times \) 200 units. The pieces in every test problem have different dimensions from each other. A test problem has never pieces that similar to one another. The test problems have a known optimum solution (Hopper and Turton, 2000). For instance, optimum solution of test problem 1 is shown in Fig. 2.

For all simulations, the number of iterations was set to 1000. The simulations were run total 510 times for the different parameters of algorithms, because of stochastic nature of GA and SA algorithms. The results of the simulations are given as average values. To compare the results easily in this study, the fitness values (or the values of cost function) taken from algorithms were normalized between 0 (the worst trim loss value) and 1 (the best trim loss value). The fitness values were interpolated based on the worst trim loss and the best trim loss for all cases.

6.1. GA solution approach

An order-based GA is combined with improved BL algorithm to solve the two-dimensional non-guillotine
rectangular packing problems. This solution approach is known as hybrid GA. In this approach, influences of various parameters are studied below two parts.

The influences of population size and mutation rate in hybrid GA: In the first part of GA solution approach, the influences of different population sizes and mutation rates were examined to find the best GA parameters for the packing problems. It was employed to solve 17 pieces five different test problems.

The simulations use the following GA parameters: Five varied population size: 20, 40, 60, 80, and 100. The length of a chromosome is 17. Selection operator is roulette wheel. Crossover operator is SJX technique. Mutation operator is swap mutation and mutation rate is varied as: 0.01, 0.1, 0.3, 0.6, 0.7, 0.9, and 1.

Fig. 3(a) shows normalized fitness values obtained from every one of 17 pieces five different test problems for different population sizes (from 20 to 100). Fig. 3(b) shows average normalized fitness values obtained from 17 pieces five different test problems in each population size. Mutation rate was fixed at 0.7. As can be seen from Fig. 3, population size of 80 produces the best results in all studies. Fig. 4(a) shows normalized fitness values obtained from every one of 17 pieces from five different test problems for different mutation rates (from 0.01 to 1). Fig. 4(b) shows average normalized fitness values obtained from 17 pieces five different test problems in each mutation rate. Population size was fixed at 80. Best results for these studies are obtained when mutation rate is taken as 0.7.

The influences of crossover techniques in hybrid GA: In the second part of GA solution approach, the influences of different crossover techniques on the solutions of the packing problems were examined using the best GA parameters obtained from the first part of GA solution approach. In order to clearly see the influences of crossover techniques, a more difficult 29 pieces test problem was employed.

The simulations in this part use the following GA parameters: Population size is 80 and mutation rate is 0.7. The length of a chromosome is 29. Selection operator is roulette wheel. Six different crossover operators are used: OBX, CX, OX, PMX, UX and SJX. Mutation operator is swap mutation.

Fig. 5 illustrates the best and worst results obtained in the second part of the GA solution approach. Fig. 5(a) and (b) show the maximum, average and minimum fitness values obtained with the OBX and SJX techniques, respectively. As can be seen in Fig. 5, the OBX technique produces better results than the other technique. The individual results obtained by using the OBX technique are changed extensively as compared to other crossover techniques. Thus there exists enough diversity in population when OBX crossover technique is used. Based on results obtained here, crossover techniques can be enumerated from the best to the worst trim loss value as: OBX, CX, OX, PMX, UX, and SJX.

6.2. SA approach

SA is combined with improved BL algorithm to solve the two dimensional non-guillotine rectangular packing problems. This solution approach is known as hybrid SA. In this approach, influences of various parameters are studied below in two parts.
The influences of parameters in hybrid SA: In the first part of SA solution approach, many different SA parameters were used to study influences of the parameters on the solutions of the packing problems. The hybrid SA was applied to solve the 17 pieces five different test problems.

The simulations in this part use the following SA parameters: Possible solution is represented using the permutation representation and the length of a solution is 17. Initial temperatures vary between 0.1 and 0.8. Final temperature is 0.01. Two neighborhood moves are applied: swapping move (sw) and shifting move (sh). Proportional decrement schedule (a) and Lundy and Mees schedule (b) are used as cooling schedule. Outer loop is equal to maximum number of iteration 1000. Three inner loop criterions are used: 3, 5 and 10.
The influences of neighborhood moves in hybrid SA:
The influences of neighborhood moves were examined using two different cooling schedules and neighborhood moves for the various initial temperature values (from 0.1 to 0.8) on the solutions of test problems. The inner loop criterion is fixed at 5. Fig. 6 shows the average normalized fitness values obtained by using two different neighborhood moves for 17 pieces five different test problems. As can be seen in Fig. 6, the swapping move works better than shifting move in the different temperature values.

The influences of cooling schedules and inner loop criterions in hybrid SA:
It was examined influences of cooling schedules and three inner loop criterion using two different cooling schedules and the swapping move for the different initial temperature values on the solutions of test problems. Fig. 7 illustrates the average normalized fitness values obtained using the swapping move, the two different cooling schedules and three inner loop criterions for the same test problems. This study was carried out to compare the cooling schedules and to investigate influences of inner loop criterions in the different initial temperature values. The inner loop criterion was chosen as 3, 5, and 10. As can be seen in the figure, the Lundy and Mees schedule works better than the proportional decrement schedule and the best results were obtained when inner loop criterion is fixed at 3.

The solution of 29 pieces a test problem for obtained best SA parameters:
In the second part of SA solution approach, it was examined on the solution of 29 pieces a test problem using the best SA parameters obtained from the first part of SA solution.

The simulations in this part use the following SA parameters: The length of a solution is 29. Initial temperatures vary between 0.1 and 0.8. Final temperature is 0.01. The swapping move and Lundy and Mees schedule is used. Outer loop is 1000 and inner loop criterion is 3.

Fig. 8 shows normalized fitness values using best parameters obtained from previous studies, which are the swapping move, the Lundy and Mees schedule, with 3 of inner loop criterion, in the different temperature values for 29 pieces test problem. Normalized fitness values are the biggest at initial temperatures (between 0.2 and 0.3). Then normalized fitness values decrease after 0.4 initial temperatures.

In Leung et al., 2001, the inner loop criterion was fixed at 5 and the influences of all other parameters on the test problems were examined using this fixed value. The test problems have identical and non-identical pieces and the number of pieces varied between 10 and 20. Two different placement algorithms were employed: the DP algorithm and the improved BL algorithm. The results of this paper show that the fixed inner loop criterion and variety of pieces cause poor searching of parameter influences. It was not determined which of them more effective: number of pieces, placement algorithms or changing SA parameters. Therefore, it was used three inner loop criterion (3, 5, and 10) to examine influences of SA parameters in this study. To examine influences of SA parameters, the number of pieces was initially taken as fixed. It was studied by five different test problems consisting of 17 individual rectangular pieces. The influences of pieces number were later examined by a test problem consisting of 29 individual rectangular pieces. The difference process algorithm is
better than the improved BL algorithm, but computationally more expensive. Therefore, the improved BL algorithm is used in this study.

To see all results taken from hybrid GA and hybrid SA, the results are given in Table 1. This table summarizes average values of the worst and the best trim loss obtained from 17 and 29 pieces test problems. The worst trim loss is changed between 7% and 21% for six different rectangular packing problems. Similarly, the best trim loss is changed between 2% and 10% for same six different rectangular packing problems. As can be seen in Table 1, better results are obtained when hybrid GA is used.

Fig. 9 shows the best fitness values obtained with using of hybrid GA and hybrid SA for 17 pieces test problems. As compared these methods in terms of iteration number, hybrid GA is much better than hybrid SA. Hybrid GA has the best fitness values after approximately 450th iterations. When hybrid SA was used for same problem, there exists a barren search until approximately 900th iterations. As can be seen in Fig. 9, results obtained from hybrid GA are better than hybrid SA. Fig. 10 shows the best fitness values obtained with using hybrid GA and hybrid SA for 29
As seen in Fig. 10, hybrid GA reached the best fitness value after approximately 200th iterations. The best fitness value by using hybrid SA was obtained after approximately 900th iterations for 29 pieces test problem. The results in Fig. 10 are very similar to the results obtained for 17 pieces test problems.

7. Conclusions

The results obtained in this study can be summarized below:

Increasing the population size influences the solution of the cutting and packing problems. However, there is no or little effect on the solution as the mutation rate increases. The best results are obtained when the population sizes vary between 60 and 80. There is little variation in mutation rates when the normalized fitness values are between 0.3 and 0.9.

The influences of different crossover techniques on the solutions of the 29 pieces a test problem were studied. The best trim loss value was obtained from OBX technique. Based on their solution performance, crossover techniques used here can be enumerated from the best to the worst trim loss value as: CX, OX, PMX, UX, and SJX.

The influences of different cooling schedules, neighborhood moves and number of inner loop on the solution of the test problems were examined for the different temperature values in SA approach. According to this examination, swapping neighborhood move works better than shifting move in the different temperature values and Lundy and Mees schedule works much better than proportional decrement schedule for different number of inner loop (3, 5, and 10). When number of inner loop is 3, the best results are obtained. Lundy and Mees schedule, swapping moves and 3 for number of inner loop should be chosen to solve the cutting problems with SA approach.

As hybrid GA is used to solve the 17 pieces five different test problems, the trim loss values changed between 2% and 11%. As hybrid SA is used to solve the 17 pieces test problems, the trim loss values varied between 4% and 21%. For the 29 pieces test problems, the trim loss values obtained using the hybrid GA varied between 5% and 9% and the trim loss values obtained with using hybrid SA varied between 10% and 17%. Based on these results, hybrid GA was much better than the hybrid SA for the two dimensional non-guillotine cutting and packing problems.

References


