Journal of Mathematics and Statistics 6 (2): 157-162, 2010
ISSN 1549-3644

A Modified Partially Mapped MultiCrossover Genetic Algorithm for Two-Dimensional Bin Packing Problem

[1]M. Sarabian and [1,2]L.S. Lee
[1]Department of Mathematics, Faculty of Science,
[2]Institute for Mathematical Research,
University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract: Problem statement: Non-oriented case of Two-Dimensional Rectangular Bin Packing Problem (2DRBPP) was studied in this study. The objective of this problem was to pack a given set of small rectangles, which may be rotated by 90°, without overlaps into a minimum numbers of identical large rectangles. Our aim was to improve the performance of the MultiCrossover Genetic Algorithm (MXGA) proposed from the literature for solving the problem. **Approach:** Four major components of the MXGA consisted of selection, crossover, mutation and replacement are considered in this study. Initial computational investigations were conducted independently on the named components using some benchmark problem instances. The new MXGA was constructed by combining the rank selection, modified Partially Mapped Crossover (PMXm), mutation with two mutation operators and elitism replacement scheme with filtration. **Results:** Extensive computational experiments of the new proposed algorithm, MXGA, Standard GA (SGA), Unified Tabu Search (UTS) and Randomized Descent Method (RDM) were performed using benchmark data sets. **Conclusion:** The computational results indicated that the new proposed algorithm was able to outperform MXGA, SGA, UTS and RDM.

Key words: Genetic algorithm, bin packing problem

INTRODUCTION

Bin packing problem is a branch of cutting and packing problems which has many applications in wood and metal industries. Non-oriented case of Two-Dimensional Rectangular Single Bin Size Bin Packing Problem (2DRSBSBPP) based on classification of Wascher *et al*. (2007) is studied in this study. Without loss of generality, the problem will be referred as Two-Dimensional Rectangular Bin Packing Problem (2DRBPP) henceforth. Lodi *et al*. (1999) defined the problem as follows:

> "Given are a set of n rectangles, which may be rotated by 90°. Each rectangle is defined by a height h_j and a width w_j, for j = 1, 2,...,n and an unlimited number of identical rectangular bins, each having height H and width W. The objective of this problem is to pack all the rectangles without overlaps, into the minimum number of bins"

The aims of this study is to improve the MultiCrossover Genetic Algorithm (MXGA) proposed by Lee (2008) for solving 2DRBPP and compare the effectiveness of the new proposed algorithm with MXGA, SGA and UTS and RDM. The multICrossover operator in the MXGA is able to repeat a standard 2-point crossover operator on the selected parents for t times in order to generate a list of temporary offspring with size of 2t. Then the fittest and a selected temporary offspring using the probabilistic binary tournament selection mechanism is chosen to be the offspring of the current generation. Swap operator is used instead of the reproduction strategy in MXGA when the multiCrossover operator is not applied to the selected parents. A random swap point is selected in a parent and the position of the substrings is swapped to form a new offspring. Two mutation operators are applied in the MXGA. First, a subset of individuals is selected from the new offspring population with a given individual mutation probability. Then each gene in the selected offspring will go through the gene mutation operator with the given probability. The replacement strategy used in MXGA is the elitism replacement scheme. In this strategy the fittest individuals are always selected from the combination of parents and offspring population before proceeding to the next

Corresponding Author: M. Sarabian, Department of Mathematics, Faculty of Science, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

generation. Filtration is used to remove the identical individuals and replaced them with randomly generated new individuals.

Our proposed algorithm is constructed by combining the most competitive techniques from each component of MXGA. The competitive techniques are obtained from the initial investigations. The new algorithm is referred as Improved MultiCrossover Genetic Algorithm (MXGAi) henceforth.

MATERIALS AND METHODS

Improved MultiCrossover Genetic Algorithm (MXGAi):

Representation: Bin permutation proposed by Lee (2008) is applied as the gene representation in the MXGAi. The length of the individual is equal to the number of rectangles. Each gene is represented by a uniform random permutation of the integer numbers of bins in the interval [1,LB], where LB is the lower bound proposed by Dell'Amico *et al*. (2002). After generating the initial population, the Lowest Gap Filled (LGF) developed by Lee (2008) is applied as a heuristic placement routine in the decoding stage of MXGAi. This placement routine provides a dynamic selection of the best fitting rectangle which can fill the existing gap in the partial layout. During the packing stage, any rectangle that cannot be feasibly packed in the bin will be kept as an unassigned rectangle in a list, which will be packed later using repack strategy. The repack strategy will try to pack any unassigned rectangles from the list into the used bin by unpacking the selected bin and repacking it after adding a rectangular from the unassigned rectangle list.

Selection mechanism: Rank selection scheme proposed by Baker (1985) is used as the selection mechanism in MXGAi. The initial investigations indicate that this selection strategy produced a better solution quality in comparison with other selection mechanisms namely: probabilistic binary tournament (Goldberg and Lingle, 1985), stochastic universal sampling (Baker, 1987) and sexual selection (Goh *et al*., 2003).

Crossover operator: Since maximizing the bin utilization of each bin is one of the objective for 2DRBPP, a specific crossover operator is proposed in this study to help the offspring to inherit the maximum bin utilization from their parents. The idea behind this crossover operator is derived from the Partially Mapped Crossover (PMX) (Goldberg and Lingle, 1985) and we will refer this crossover as modified Partially Mapped Crossover (PMXm). Similar to MXGA, the crossover operator is applied for t times in MXGAi. This crossover operator is applied as follows:

Step 1: Sort the bins in Parent 1 and Parent 2 (P1 and P2) in a non-increasing order of their bin utilization.

Step2: Group the rectangles with the same bin number in both P1 and P2.

Step 3: Let n_1 and n_2 be the number of applied bins in P1 and P2 respectively and $N = \min\{n_1, n_2\}$.

Step 4: Set $i \leftarrow 1$ and $C \leftarrow 0$. The variables $i(i = 1, 2,\ldots,N)$ and C are represent the bin number and the number of bins with different bin utilization in P1 and P2 respectively. Let U_{1i} and U_{2i} be the bin utilization of the bin number i in P1 and P2 respectively and let c_{1i} and c_{2i} be the number of rectangles which are allocated in the bin number i in P1 and P2 respectively.

Step 5: While $i \le N$, if $U_{1i} = U_{2i}$, set $i \leftarrow i+1$. Else Go to Step 6. If ($i>N$ and $C = 0$) then two offspring are generated by duplicating the parents and the procedure is stopped.

Step 6:

- (i) If $U_{1i}>U_{2i}$ and $c_{1i} = c_{2i}$, the bin number i in the offspring will inherit the corresponding rectangles from P1 and the other bins will inherit the ordering from P2, while the interchange mapping is applied for the rectangles which were allocated in the bin number i of P2. Go to Step 7.
- (ii) If $U_{1i}>U_{2i}$ and $c_{1i}>c_{2i}$, the bin number i in the offspring will inherit the corresponding rectangles from P1 and the other bins will inherit the ordering from P2, while the interchange mapping is applied and the extra rectangle(s) will be removed from the relevant bin(s) of P2 and they will be allocated in the bin number i of offspring. Go to Step 7.
- (iii) If $U_{1i}>U_{2i}$ and $c_{2i}>c_{1i}$, the interchange mapping is applied and the extra rectangles will be removed from bin number i of P2 and they will be allocated in the other bin which is randomly selected from the interval $[1,n_2]$. Note that if the generated random number is equal to i, another random number should be generated. Go to Step 7.
- (iv) Else if $U_{1i} < U_{2i,}$ the process in (i), (ii) or (iii) performs in apposite fashion. Go to Step 7.

Step 7: Set $i \leftarrow i+1$ and $C \leftarrow C+1$. If ($i>N$) or ($C \ge t$) ($t = 5$), stop the procedure. Else go to Step 5.

Note that generating large number of temporary offspring is very time consuming and in the case that the computational time is fixed, the algorithm will fail to explore the other region of the solution space. On the other hand a small number of temporary offspring may fail to use the advantage of MultiCrossover process. Depending on the variable N and C, parameter t has the ability to change during the procedure but it will not exceed 5.

Since PMXm is an effective but complicated crossover operator and applying this operator in the MultiCrossover process will be time consuming, it can be mixed with other simple crossover operators (e.g., 1-point Crossover, 2-point Crossover and Uniform Crossover (UX)). This can be done by generating 2 temporary offspring from applying each crossover operator on the selected parents in order to generate a pool of temporary offspring. We refer to this crossover operator as Mixed Crossover (MX).

Initial computational experiments which are conducted independently for MXGA with different crossover operators show that PMXm is able to outperform other crossover operators. Other crossover operators which were applied in the initial investigations are namely: 2-point Crossover, Standard UX (Syswerda, 1989), 1X (Davis, 1985), Order Crossover (Davis, 1985), Maximal Preservative Crossover (Muhlenbein *et al*., 1988), Matrix Crossover (Homaifar *et al*., 1993) and MX.

It is worth noting that except for the standard UX which can be applied to any type of gene representation, the other crossover operators are only suitable for item permutation, while the matrix crossover is only suitable for matrix representation. Since the gene representation which is defined in the MXGAi is bin permutation we considered some modifications in order to apply the above crossover operators in the MXGAi. Note that in all cases the MultiCrossover operator is applied to the selected parents via a given crossover probability ($p_c = 0.75$).

Mutation operator: Initial investigations indicate that the proposed mutation operator with two mutation operators by Lee (2008) in MXGA will generate a better solution quality in comparison with the Exchange Mutation (Banzhaf, 1990), Scramble Mutation (Syswerda, 1991) and Displacement Mutation (Michalewiz, 1992). Hence the applied mutation operator for MXGAi is the mutation with two mutation operators.

Replacement strategy: Initial computational results show that elitism replacement scheme produces a better solution quality in comparison with the steady state strategy. Thus the proposed MXGAi applies elitism replacement scheme. The advantage of this scheme is that the fit individuals are never lost unless fitter individuals are generated. During the elitism replacement stage, the combination of parent and offspring population with the size of $2P_{pop}$ is sorted in a non-increasing order of their related fitness value. Then the first half of the combined population forms the new population for the next generation. After the elitism replacement scheme, the filtration process is applied in the MXGAi to identify the identical individuals from population and replace them by uniform randomly generated new individuals.

Unified Tabu Search (UTS): The UTS algorithm applied in the computational experiments is introduced by Lodi *et al*. (1999). According to them, this algorithm has the ability to generate high quality solutions, regardless of the inner heuristic placement routine applied in the search. The main feature of this algorithm is a unified parametric neighborhood which is independent of the specific packing problem to be solved and whose size is dynamically changed during the search.

Randomized Descent Method (RDM): The RDM used in the computational experiments is developed by Lee (2008). The main characteristics of the RDM are similar to the UTS, while the main differences between RDM and UTS are an acceptance rule which allows the natural moves solution up to R consecutive iterations before terminating the algorithm and the randomization process which selects a random move from the list of identical moves in a single iteration.

RESULTS

The performance of the proposed MXGAi is compared with the MXGA, SGA, UTS and RDM. All the algorithms are coded in ANSI-C using Microsoft Visual C++ 6.0 as the compiler and run on a Pentium 4, 2.0 GHz processor with 1.0 GB memory. Ten different classes of problem instances are considered. The first six classes (I-VI) are proposed by Berkey and Wang (1987) and the next four classes (VII-X) are proposed by Martello and Vigo (1998) (Table 1). In each of the first six classes, all the rectangles are generated in the same interval while in the other four classes a more realistic situation is considered and the rectangles are classified into four types:

Type 1: w_j uniformly random in $\left[\frac{2}{3}W, W\right]$, h_j uniformly random in $\left[1, \frac{1}{2}H\right]$

Type 2: w_j uniformly random in $\left[1, \frac{1}{2}W\right]$, h_j uniformly random in $\left[\frac{2}{3}H, H\right]$

Type 3: w_j uniformly random in $\left[\frac{1}{2}W, W\right]$, h_j uniformly random in $\left[\frac{1}{2}H, H\right]$

Type 4: w_j uniformly random in $\left[1, \frac{1}{2}W\right]$, h_j uniformly random in $\left[1, \frac{1}{2}H\right]$

For each class, we considered five values of n: 20, 40, 60, 80 and 100 where n represents the number of rectangles which are going to be packed into the bins. For each combination of class and value of n, ten problem instances are generated. The performance of the local search algorithms is compared on the basis of the Average Ratio and the Overall Bin Utilization:

$$\text{Average ratio} = \frac{\sum_{i=1}^{10} \frac{UB_i}{LB_i}}{10} \quad (1)$$

where, UB_i and LB_i represent the heuristic solution and the lower bound for the problem instance i respectively:

$$\text{Overall bin utilization} = \frac{\sum_{i=1}^{10} \frac{\sum_{j=1}^{UB_i} \left(\frac{A_j}{A}\right)}{UB_i}}{10} \quad (2)$$

Where:
A_j = The total area of all the rectangles in bin j (j = 1, 2, …, UB_i)
A = The area of the bin

In order to have a fair comparison between MXGAi, MXGA and SGA, all these algorithms will start their implementation with the same initial population and a stopping criteria of 120 CPU seconds is employed for each problem instances in all algorithms.

Table 1: Classes for the problem instances

Class	Bin (W×H)	Rectangle (w_j and h_j)
I	10×10	Uniformly random in [1,10]
II	30×30	Uniformly random in [1,10]
III	40×40	Uniformly random in [1,35]
IV	100×100	Uniformly random in [1,35]
V	100×100	Uniformly random in [1,100]
VI	300×300	Uniformly random in [1,100]
VII	100×100	Type 1 with probability 70%, Type 2, 3 and with probability 10% each
VIII	100×100	Type 2 with probability 70%, Type 1, 3 and 4 with probability 10% each
IX	100×100	Type 3 with probability 70%, Type 1, 2 and 4 with probability 10% each
X	100×100	Type 4 with probabilty 70%, Type 1, 2 and 3 with probability 10% each

Note that the crossover operator is applied for t (t = 5) and t'(t'≤5) times in both MXGA and MXGAi respectively. The MultiCrossover operator in MXGA has the ability to generate 2t temporary offspring, while in MXGAi the operator generates t' temporary offspring. Unlike MXGA and MXGAi, SGA applies the standard 2-point crossover operator in order to generate exactly two offspring from two selected parents. The reproduction strategy is applied in SGA instead of swap operator. The gene mutation operator in SGA applies for the entire population, while in MXGA and MXGAi only 25% (p_M = 0.25) of the individuals have the chance to be mutated. Steady state replacement strategy is employed as the replacement strategy in SGA.

The extensive computational results are presented in Table 2. The first pair of columns in Table 2 indicates the class and the value of n. The following pairs of columns give the result of MXGA, MXGAi, SGA and UTS respectively. For each algorithm, entries in the first and second columns of the table report the average ratio (Eq. 1) and the average overall bin utilization (Eq. 2) respectively which are computed for 15 times over all problem instances. The final row of each class gives the overall average over that class for all values of n and the final row of the table gives the overall average over all classes.

DISCUSSION

The computational results of MXGAi and MXGA indicate that combining the PMXm crossover operator with rank selection mechanism in MXGAi can improve the average ratio over all classes about 0.4%. Improvement of 1.5% in the MXGA as compared to SGA, shows that applying the crossover operator for t times in MXGA can improved the solution quality.

Table 2: A Comparison of MXGAi with MXGA, SGA, UTS and RDM

		MXGA		MXGAi		SGA		UTS		RDM	
Class	n	Ratio	OBU	Ratio	OBU	Ratio	OBU	Ratio	OBU	Ratio	OBU
	20	1.027	81.46	1.027	81.46	1.027	81.04	1.027	80.26	1.052	76.77
	40	1.033	86.45	1.031	86.72	1.049	84.42	1.032	85.48	1.064	80.75
I	60	1.040	88.30	1.040	88.41	1.040	87.85	1.040	87.33	1.076	81.90
	80	1.059	87.30	1.059	87.49	1.059	86.93	1.059	86.43	1.046	82.13
	100	1.025	92.79	1.023	93.29	1.025	92.59	1.035	90.75	1.063	86.31
	Average	1.037	87.26	1.036	87.47	1.040	86.57	1.039	86.05	1.060	81.57
	20	1.000	42.40	1.000	42.40	1.000	42.40	1.000	42.40	1.000	42.40
	40	1.100	56.02	1.100	55.74	1.100	55.03	1.100	53.76	1.100	53.58
II	60	1.000	76.42	1.000	76.34	1.167	70.38	1.000	75.80	1.007	75.12
	80	1.011	82.60	1.000	83.48	1.067	78.00	1.000	83.30	1.024	81.25
	100	1.000	81.12	1.000	81.40	1.033	78.05	1.000	81.20	1.004	80.68
	Average	1.022	67.71	1.020	67.87	1.073	64.77	1.020	67.29	1.027	66.61
	20	1.037	68.91	1.037	68.94	1.057	66.21	1.037	67.04	1.053	65.79
	40	1.087	74.47	1.084	74.81	1.106	71.15	1.089	72.32	1.118	68.61
III	60	1.086	80.94	1.078	82.07	1.095	78.35	1.079	80.21	1.125	74.34
	80	1.080	81.37	1.072	82.46	1.092	79.17	1.081	80.66	1.125	74.41
	100	1.067	83.65	1.061	84.36	1.087	80.39	1.062	83.41	1.138	75.79
	Average	1.071	77.87	1.067	77.92	1.087	75.06	1.070	76.73	1.112	71.79
	20	1.000	38.36	1.000	38.36	1.000	38.36	1.000	38.36	1.000	38.36
	40	1.000	56.51	1.000	56.16	1.000	55.01	1.000	54.94	1.020	52.98
IV	60	1.100	71.08	1.100	71.21	1.100	69.64	1.100	69.85	1.100	69.76
	80	1.091	74.54	1.058	77.13	1.100	72.86	1.033	78.44	1.071	76.99
	100	1.038	78.21	1.033	78.94	1.067	75.37	1.037	78.82	1.038	78.36
	Average	1.046	63.74	1.038	64.36	1.053	62.25	1.034	64.08	1.046	63.29
	20	1.042	70.46	1.042	70.50	1.058	68.06	1.042	68.38	1.060	66.83
	40	1.067	75.95	1.063	76.39	1.080	73.36	1.072	72.95	1.109	69.76
V	60	1.067	78.23	1.062	78.96	1.075	76.33	1.074	76.52	1.110	71.55
	80	1.067	78.59	1.065	79.20	1.080	76.40	1.071	77.34	1.109	72.18
	100	1.065	82.22	1.054	83.62	1.081	79.81	1.072	81.03	1.106	75.86
	Average	1.061	77.09	1.057	77.74	1.075	74.79	1.066	75.24	1.099	71.23
	20	1.000	29.23	1.000	29.23	1.000	29.23	1.000	29.23	1.000	29.23
	40	1.400	48.92	1.400	48.63	1.400	47.42	1.400	46.73	1.400	47.44
VI	60	1.023	68.37	1.017	68.79	1.050	65.96	1.050	65.47	1.030	66.72
	80	1.000	67.55	1.000	67.93	1.000	67.00	1.000	67.02	1.000	67.05
	100	1.078	74.46	1.067	75.57	1.100	72.40	1.067	74.61	1.069	74.86
	Average	1.100	57.70	1.097	58.03	1.110	56.40	1.103	56.61	1.100	57.06
	20	1.110	71.76	1.110	71.86	1.130	68.79	1.130	68.50	1.126	67.92
	40	1.083	79.52	1.076	80.46	1.116	75.28	1.083	78.07	1.123	73.41
VII	60	1.047	85.16	1.042	85.72	1.085	80.68	1.057	84.18	1.100	77.93
	80	1.084	83.95	1.073	85.34	1.091	82.22	1.092	83.18	1.125	77.70
	100	1.071	85.37	1.064	86.46	1.092	82.20	1.079	84.52	1.119	78.35
	Average	1.079	81.15	1.073	81.97	1.103	77.83	1.088	79.69	1.119	75.06
	20	1.100	72.41	1.100	71.85	1.120	69.01	1.130	69.40	1.112	69.77
	40	1.093	80.06	1.092	80.47	1.093	78.41	1.105	78.27	1.122	75.18
VIII	60	1.060	84.65	1.056	85.40	1.099	79.67	1.070	83.31	1.106	78.22
	80	1.075	84.59	1.066	85.95	1.106	80.62	1.081	83.86	1.116	78.37
	100	1.070	85.43	1.061	86.68	1.086	82.79	1.083	84.23	1.113	78.99
	Average	1.080	81.43	1.075	82.07	1.101	78.10	1.094	79.81	1.114	76.11
	20	1.000	43.57	1.000	43.57	1.007	43.03	1.000	42.97	1.004	42.31
	40	1.011	45.74	1.011	45.75	1.011	45.56	1.011	45.10	1.011	44.27
IX	60	1.007	43.56	1.007	43.56	1.007	43.49	1.007	42.75	1.007	42.12
	80	1.009	45.11	1.009	45.11	1.009	45.01	1.009	43.68	1.009	43.50
	100	1.007	45.69	1.007	46.06	1.007	46.01	1.007	43.96	1.008	43.87
	Average	1.007	44.73	1.007	44.81	1.008	44.62	1.007	43.69	1.008	43.21
	20	1.125	68.33	1.125	68.33	1.125	67.01	1.125	64.12	1.157	62.95
	40	1.061	79.58	1.061	79.78	1.061	77.87	1.061	77.55	1.077	75.45
X	60	1.069	83.74	1.061	84.72	1.085	80.59	1.076	81.90	1.094	79.20
	80	1.060	85.19	1.049	86.59	1.064	83.17	1.049	85.92	1.088	80.21
	100	1.050	86.18	1.043	87.01	1.066	83.36	1.045	86.76	1.081	81.32
	Average	1.073	80.60	1.068	81.29	1.080	78.40	1.071	79.25	1.099	75.82
AVERAGE		1.058	71.93	1.054	72.35	1.073	69.88	1.059	70.84	1.078	68.18

Generally, RDM produces the least impressive results and this indicates that the algorithm is not a suitable choice for solving 2DRBPP. According to the computational results, UTS is performing better than RDM. Average computed values, for each class indicates that MXGAi outperformed the other algorithms with only one exception occurs for class IV. In this class UTS performs the best only in term of overall average ratio. The overall results over all the classes in Table 2 indicate that MXGAi is the more preferred choice followed by MXGA, UTS, SGA and RDM for solving 2DRBPP.

CONCLUSION

This study presents a new MXGA for solving 2DRBPP by combining the rank selection, PMXm, mutation with two mutation operators and elitism replacement scheme with filtration. Extensive computational experiments were conducted and the results indicated that the new proposed algorithm is able to outperform the MXGA, SGA, UTS and RDM.

REFERENCES

Baker, J.E., 1985. Adaptive Selection Methods for Genetic Algorithms. Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, July 1985, Lawrence Erlbaum Associates, Carnegie Mellon University, Pittsburgh, Pa, pp: 101-111. http://www.questia.com/PM.qst?a=o&docId=97565147

Baker, J.E., 1987. Reducing bias and inefficiency in the selection algorithm. Proceedings of the 2nd International Conference on Genetic Algorithms and Their Applications, July 1987, Lawrence Erlbaum Associates Cambridge, Massachusetts, United States, pp: 14-21. http://www.questia.com/PM.qst?a=o&docId=97597314

Banzhaf, W., 1990. The "molecular" travelling salesman. J. Biol. Cybernet., 64: 7-14. DOI: 10.1007/BF00203625

Berkey, J.O. and P.Y. Wang, 1987. Two-dimensional finite bin-packing algorithms. J. Operat. Res. Soc., 38: 423-429. DOI: 10.1057/jors.1987.70

Davis, L., 1985. Job shop scheduling with genetic algorithms. Proceeding of the 1st International Conference on Genetic Algorithms and Their Applications, July 1985, Lawrence Erlbaum Associates, Carnegie Mellon University, Pittsburgh, Pa, pp: 136-140. http://www.questia.com/PM.qst?a=o&docId=97565182

Dell'Amico, M., S. Martello and D. Vigo, 2002. A lower bound for the non-oriented two-dimensional bin packing problem. J. Discr. Applied Math., 118: 13-24. DOI: 10.1016/S0166-218X(01)00253-0

Goh, K.S., A. Lim and B. Rodrigues, 2003. Sexual selection for genetic algorithms. J. Artif. Intel. Rev., 19: 123-152. DOI: 10.1023/A:1022692631328

Goldberg, D.E. and R. Lingle, 1985. Alleles, loci and the travelling salesman problem. Proceeding of the 1st International Conference on Genetic Algorithms and their Applications, July 1985, Lawrence Erlbaum Associates, Carnegie Mellon University, Pittsburgh, PA., pp: 154-159. http://www.questia.com/PM.qst?a=o&docId=97565200

Homaifar, A., S. Guan and G.E. Leipins, 1993. A new approach on the travelling salesman problem by genetic algorithm. Proceeding of the 5th International Conferences on Genetic Algorithms, June 1993, Morgan Kaufmann, San Francisco, CA., USA., pp: 460-466. http://portal.acm.org/citation.cfm?id=604335

Lee, L.S., 2008. A genetic algorithm for two-dimensional bin packing problem. MathDigest. Res. Bull. Inst. Math. Res., 2: 34-39.

Lodi, A., S. Martello and D. Vigo, 1999. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. INFORMS J. Comput., 11: 345-357. DOI: 10.1287/ijoc.11.4.345

Martello, S. and D. Vigo, 1998. Exact solution of the two-dimensional finite bin packing problem. J. Manage. Sci., 44: 388-399. DOI: 10.1287/mnsc.44.3.388

Michalewiz, Z., 1992. Genetic Algorithms + Data Structures = Evolution Programs. 3rd Rev. Edn., Berlin, Springer Publishers, ISBN: 978-3-540-60676-5, pp: 101-105.

Muhlenbein, H., M. Gorges-Schleuter and O. Kramer, 1988. Evolution algorithms in combinatorial optimization. J. Parall. Comput., 7: 65-85. DOI: 10.1016/0167-8191(88)90098-1

Syswerda, G., 1989. Uniform crossover in genetic algorithms. Proceeding of the 3rd International Conferences on Genetic Algorithms, June 1989, Morgan Kaufmann, San Francisco, CA., USA., pp: 2-9. http://portal.acm.org/citation.cfm?id=657265

Syswerda, G., 1991. Schedule Optimization Using Genetic Algorithm. In: Hand Book of Genetic Algorithm, Davis, L. (Ed.). Van Nostrand Reinhold, New York, USA., ISBN: 978-0442001735, pp: 332-349.

Wascher, G., H. Haubner and H. Schumann, 2007. An improved typology of cutting and packing problems. Eur. J. Operat. Res., 183: 1109-1130. DOI: 10.1016/j.ejor.2005.12.047