Analog Integrated Circuits and Signal Processing
https://doi.org/10.1007/s10470-020-01598-w

A nature inspired optimization algorithm for VLSI fixed-outline floorplanning

M. Shunmugathammal[1] · C. Christopher Columbus[2] · S. Anand[3]

Received: 23 May 2019 / Revised: 23 May 2019 / Accepted: 29 January 2020

Abstract

VLSI floorplan optimization problem aim to minimize the following measures such as, area, wirelength and dead space (unused space) between modules. This paper proposed a method for solving floorplan optimization problem using Genetic Algorithm which is named as 'Lion Optimization Algorithm' (LOA). LOA is developed for non-slicing floorplans having soft modules with fixed-outline constraint. Although a number of GAs are developed for solving VLSI floorplan optimization problems, they are using weighted sum approach with single objective optimization and crossover between two B*tree structure is not yet attempted. This paper explains, power of B*tree crossover operator for multiobjective floorplanning problem. This operator introduces additional perturbations in initial B*tree structure to create two new different B*tree structures compared with classical GA approach. Simulation results on Microelectronics Center of North Carolina and Gigascale Systems Research Center benchmarks indicate that LOA floorplanner achieves significant savings in wirelength and area minimization also produces better results for dead space minimization compared to previous floorplanners.

Keywords B*tree crossover · Lion Optimization Algorithm · LOA · Genetic algorithm · VLSI · Multiobjective optimization

1 Introduction

VLSI is the process of fabricating million number of transistors into a single chip or IC. An Integrated Circuit (IC) is a well packaged electronic circuit on a small piece of single crystal silicon measuring few millimetres by comprising active devices, passive devices and their interconnections. The physical design process is nothing but identifying the physical location of active devices and interconnections between them within the boundary of an integrated circuit in which floorplanning is an important step which is nothing but arranging a set of rectangular modules on a rectangular chip area so that performance measures such as total layout area and wirelength between the modules have to be optimized [1]. The resulting layout is called a floorplan. Floorplanning problem could be solved with the constraints of non-overlapping between any two modules and fixed-outline which results in minimum Dead Space (DS). Dead Space or White space is the uncovered area existing within the floorplan during the optimisation process.

The floor planning problem has been proved to be NP hard [1, 10, 25, 32]. Evolutionary Algorithms (EA) are stochastic search methods that mimic the metaphor of natural biological evolution and the social behaviour of species. The first evolutionary-based technique introduced in the literature was the Genetic Algorithms (GA) [14].

✉ M. Shunmugathammal
shummyb@gmail.com

C. Christopher Columbus
christoccc@gmail.com

S. Anand
anandmay22@gmail.com

1 Cape Institute of Technology, Tirunelveli, Tamilnadu 627114, India

2 Department of Computer Science Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu 627 152, India

3 Department of Electronics and Communication, Rajagiri School of Engineering and Technology, Rajagiri Valley, Kakkanad, Kochi, Kerala 682039, India

Published online: 05 February 2020

Genetic Algorithm is part of metaheuristic algorithms then acceptable solution for floorplan optimization could be obtained by using this GA in a reduced computational time. In GA, search space grows exponentially with the problem size. In order to solve this large size problems, nature-inspired algorithms [26] a subset of bio-inspired optimization algorithms have been effectively developed by Rechenberg [20] and Holland [14]. Bongard [4] and Forbes [12] apply these bio-inspired algorithms in almost all the emerging fields. Souza and Costa [27] introduced bio-inspired algorithms for real life problems. The bio-inspired optimization algorithms can be broadly categorized into two namely, Evolutionary algorithms [4, 28] and swarm intelligence. Chen et al. [6] proposed particle swarm optimization (PSO) technique to obtain a feasible floorplanning in VLSI circuit physical placement. This PSO was applied based on module number with integer coding and a new recommended value of acceleration coefficients for optimal placement solution. Hoo et al. [13] proposed variable order ant system (VOAS) with a floorplan model namely corner list to optimize layout area and wirelength. Corner list is used to represent the floorplan layout in VOAS. Valenzuela and Wang [8] proposed GA for VLSI floorplanning problems. Rajakumar [19] described Lions mating and territorial defense and territorial takeover in his LOA. As per his LOA, mating derives new solutions from existing and in territorial defense and territorial takeover, bad solution is replaced by new best solution. In addition, with mating and territorial defense and takeover, Maziar and Fariborz [18] proposed a new LOA, in which they described the other characters of lions such as special style of prey capturing, territorial marking, migration, difference between life style of nomad and resident lions for solving complex optimization problems. In this LOA paper, characters of Lion such as, mating, territorial defense and takeover, territorial marking and difference between life style of nomad and resident lions are used for solving VLSI floorplanning problems. Using this LOA, objective function of any floorplanning problem could be computed by doing B*tree crossover.

For the past 3 decades, GA has been applied to the VLSI physical design problems. Genetic algorithms for VLSI standard cell placement problem was proposed by Shahookar and Mazumder [24], Cohoon and Paris [5], Mazumder and Rudrick [17] and Sait et al. [21]. Also, they have presented distributed genetic algorithms for VLSI physical design problems. Valenzuela and Wang [9] and Lin et al. [16] presented genetic algorithms using polish expression to solve the VLSI floorplanning problem. Polish expression is only for slicing floorplans which could not handle non-slicing floorplans. Simple B*tree based simulated Annealing approaches [2, 7, 33] cannot explore solution space like GA, PSO and other similar algorithms. Areibi [3] and Tang and Yao [29, 30] presented memetic algorithms for VLSI floorplan problems which has computationally expensive O-tree representation. Merging of two O-tree was not possible [33] so, in this work B*tree representation is used.

Most of the VLSI floor planning algorithms in the literature use the weighted sum approach to minimize the area and wirelength simultaneously. The difficulty with this approach arises while assigning weights, resulting in undesirable bias towards a particular objective. Alternatively, multiobjective methods are capable of producing different trade-off solutions, for the problem under consideration [11]. After the detailed literature survey, crossover between two B*tree structures have not yet been proposed in GA for VLSI floorplan optimization problems. In this LOA, GA is motivated by the concept of crossover between two B*tree structures.

Literature survey on evolutionary algorithms showed that, GA has been proved very efficient for its quick convergence and successively applied for VLSI non-slicing floorplans. So, in this work, an attempt is also made to solve the modern non-slicing VLSI design floorplanning problem with the multiobjective of minimizing the silicon area and interconnecting wirelength along with the fixed outline constraint, using a genetic algorithm-based methodology resulting in minimum area and wirelength in considerable amount of time. In this paper, A novel algorithm namely, "Lion Optimization algorithm" (LOA) has been proposed for non-slicing floorplans with the fixed outline constraint.

This paper is structured as follows; Lion's social behavior is under Sect. 2, proposed LOA described in Sect. 3. The results and discussions are in Sect. 4. Finally, conclusion is in Sect. 5.

2 Lion's social behaviour

The Lion's social structure is very interesting to observe. Lions are strong in both social behavior and appearance. This wild cat species has two types of social organization: resident lions and nomad lions. Resident Lions normally living in groups. These groups are called prides. Each pride consists of five or six females with their offsprings and one male or there can be up to two of them. In each pride only one will be the dominant male and the dominant one that has the right to mate with the females. Each and every member of the pride has a unique role to play. This role depends on their needs and the size of the pride. Male offsprings are excluded from their birth pride when they become sexually mature [19]. Nomads which are the second organizational behavior. Nomads are either in pairs or singularly. Male lion pairs are more seen among nomads because who have been excluded from their maternal pride. In general nomad lion may become resident and resident may become nomad [18, 19]. The lifestyle of lion switches between resident to nomad and vice versa. Both male and female lions in the same pride work as a team to protect each other and the young offsprings from other predators. There can be brutal battles between entire prides of lions over territory. When they have locations that overlap there may be a fight for who gets to keep it. Such battles are becoming more frequent. Lion's social behavior begins when a new nomad male takes over a pride. In this takeover, the male lion kills all the existing offsprings so that it can reproduce right away. In this work, mating and territorial takeover behavior of lions are used for optimization in VLSI floorplanning.

2.1 Objective function (fitness function) of proposed LOA

In this LOA, each individual admissible floorplan solution is represented by B*tree. This B*tree is taken as a single lion. Cost function of this lion (B*tree) is calculated using Eqs. (1) and (2). In general, VLSI floorplanning problems are cost minimization problems. The cost function/fitness function F for a floorplan solution x is given by,

1. *Area Objective* The area objective is represented by the following expression

$$F1(x) = \alpha 1\left(\frac{A}{norm\_A}\right) + \alpha c(\text{ARcost})^2 \tag{1}$$

2. *Wirelength Objective* The wirelength objective is represented by the following expression

$$F2(x) = \alpha 2\left(\frac{W.L}{norm\_W.L}\right) + \alpha c(\text{ARcost})^2 \tag{2}$$

where A is the area of the current solution x, norm_A is the average area, W.L is the wirelength of the current solution x, norm_W.L is the average wirelength, αc is the weight for aspect ratio cost (ARcost) which is assigned as one. ARcost is the aspect ratio cost given by the difference between the current floorplan aspect ratio and desired floorplan aspect ratio. The wirelength is calculated using half perimeter of minimal rectangle, enclosing the centers of the modules. Estimation of norm_A and norm_W.L could be performed by B*tree perturbations (k times; in this case k = 1000) and the average of area and wirelength are calculated. Dead Space (%) can be calculated by using Eq. (3)

$$\text{DS} = \frac{\text{Floorplan Area} - \text{Sum of all module Area (A)}}{\text{Floorplan Area}} * 100\% \tag{3}$$

3 LOA for floorplan optimization

This chapter presents a method to solve floorplan optimization problem using Genetic Algorithm based Lion Optimization Algorithm (LOA) which is developed for non-slicing floorplans having hard modules with fixed-outline constraint. Figure 1 shows the flowchart for Lion Optimization Algorithm.

The basic working model of proposed Lion Optimization Algorithm is grouped into 4 different operations. They are, (1) Initial population Generation, which is responsible for generating different B*tree structures using mutation (2) Mating is enough to derive new B*tree structures (3) Territorial Defense and (4) Territorial Takeover is for finding new best B*tree fitness(cost) function and replacing worst fitness function by new best solution. The above repeated procedure makes heuristic search for faster converging and to identify near to optimal solution for floorplanning problems.

The following steps explained about proposed Lion Optimization Algorithm.

Step 1 Pride Generation or Initial population Generation.
Step 1.1 Generate initial B*tree for given floorplanning problem.
Step 1.2 Mutation: subject initial B*tree to perturbations for different B*tree structures.
Step 2 Mating.
Step 2.1 Gender grouping.
Step 2.2 Crossover: B*tree crossover.
Step 2.3 Kill sick/weak off springs.
Step 2.4 Update Pride.
Step 3 Territorial Defense.
Step 3.1 Keep the record of offspring's fitness function.
Step 3.2 Do.
Step 3.2.1 Generate and Trespass.
Nomadic lion.
Until stronger Nomadic lion trespasses.
Until offsprings get matured.
Step 4 Territorial Takeover.
Step 4.1 Selection of best male lion (B*tree with least fitness function).
Step 4.2 Go to Step 2 until termination criteria is met.

3.1 Pride generation or initial population generation

In initial population, each individual floorplan is represented by its equivalent B*tree structure and each B*tree structure is treated as single lion. Generation of initial

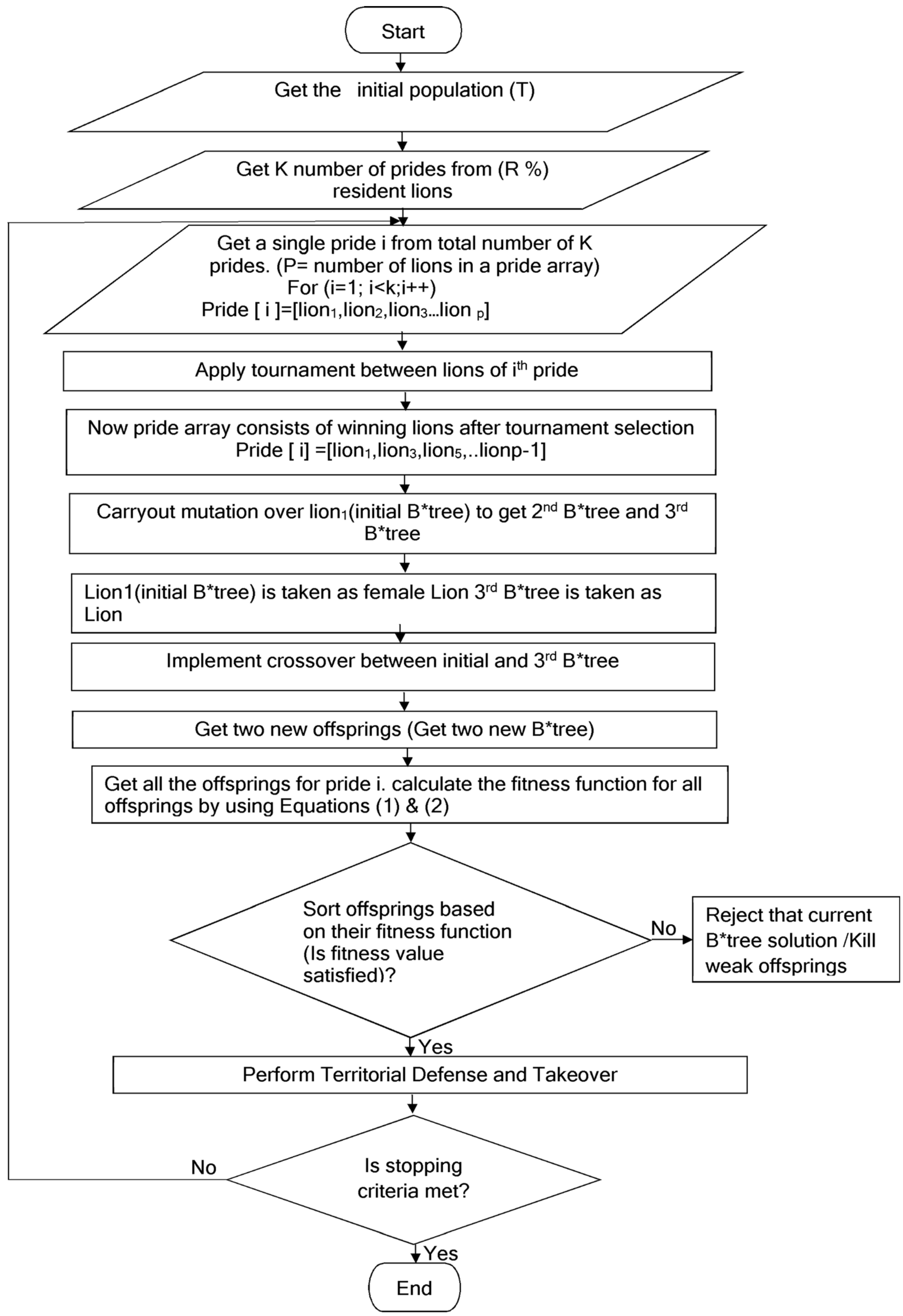


Fig. 1 Flow chart for proposed Lion Optimization Algorithm (LOA)

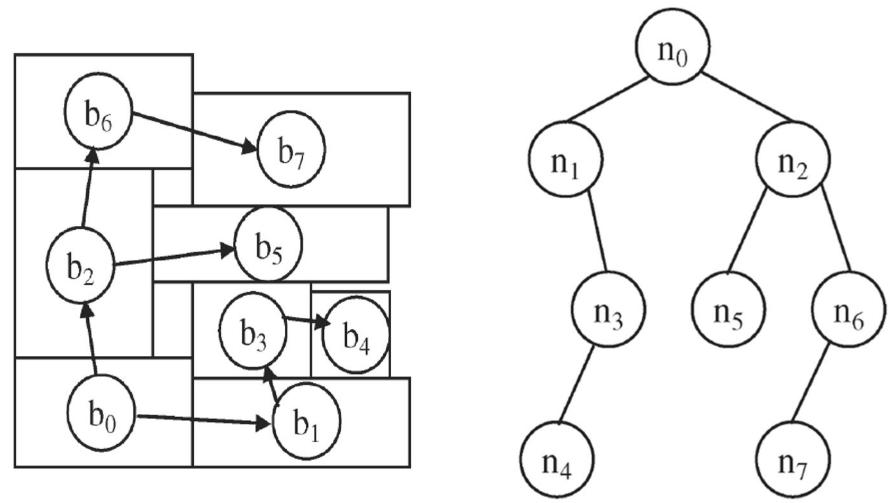
population starts from an admissible floorplan as shown in Fig. 2(a). An initial B*tree structure is generated from this admissible floorplan and this initial B*tree structure is the seed for population generation [shown in Fig. 2(b)].

3.1.1 Mutation

Perturbation is rearranging the modules which are present inside the layout to get another B*tree. Initial B*tree [shown in Fig. 2(b)] is the first solution of the given floorplan problem then this B*tree is subjected to random perturbations using following GA operators to produce large amount of B*tree solutions (initial Population).

1. Rotate a module.
2. Swap between two nodes.
3. Move a node to another place.

Fig. 2 An example floorplan and its B*tree representation



(a) Admissible floorplan *(F)* **(b)** Initial B*tree representing floorplan

Different B*tree structures (Initial population) are generated randomly using this B*tree perturbation. This LOA generates 10n number of B*trees from an initial B*tree during initial population generation, where, n is the number of modules in the initial floorplan. Flowchart for initial population generation is shown in Fig. 3. LOA objective function leads the search to identify the solution near to the optimal. Tournament selection process is applied between the lions on the same pride based on the fitness function. The lion which satisfies this cost function (fitness value) is taken for next process mutation and crossover.

3.2 Mating

Mating is a process of deriving new B*tree structures (new solutions) from the existing B*tree that includes process of B*tree crossover.

3.2.1 Gender grouping

In this LOA, initial population is taken as (T) in which some of the lions (%R) population are treated as resident lions, rest of the lions are treated as nomad lions and initial population T must be sum of nomad lions and resident lions. In LOA, female lions are generated during population generation; whereas male lions are generated during crossover operation and 50% of the pride's members are considered as female lion and the remaining 50% members are considered as a male lion. Tournament selection is applied between the lions in the same pride. Winners are taken into the next process mutation and cross over. From initial population T, k number of prides can be generated. Each pride consists of p number of lions. Winners in each pride after tournament selection are subjected to mating. Classification between female lion and male lion is used to find diversification among the solutions and killing sick/weak offsprings ensures the derived solutions to be best. Lion1 (initial B*tree) in T (initial population) are again partitioned into k subsets called prides. Each pride has p number of lions. In p number of lions, F% is taken as female and the rest M% are taken as male.

This pride is subjected to mutation (perturbation) process to get 2nd B*tree and 3rd B*tree. Initial B*tree in ith pride is taken as lion (female) and 3rd B*tree in that same pride is taken as lion (male). Crossover is done between initial B*tree (female lion) and 3rd B*tree (male lion). An example for B*tree mutation and crossover is shown by Fig. 4.

3.2.2 B*tree crossover (BX)

Figure 4(a) shows initial B*tree (female lion). The circled sub trees in the Fig. 4(a) are going to be swapped, i.e, Left sub tree rooted by node $\boldsymbol{n_9}$ and its leaf node $\boldsymbol{n_{10}}$, and right sub tree rooted by node $\boldsymbol{n_3}$, and its leaf node $\boldsymbol{n_4}$. This swapping produces the 2nd B*tree as shown in Fig. 4(b).

The 2nd B*tree will continue the swapping of for generating 3rd B*tree (male lion). Structural information between the initial (female lion) and 3rd B*tree (male) is interchanged by B*tree crossover. In both B*trees, the cross over point should be the right side of the root node. Recombination between initial and 3rd B*tree results offspring_1/new solution_1 and off spring_2/new solution_2. Figure 4(a), (c) show initial and 3rd B*tree with crossover point on right side below root node. Figure 4(d) is a new off spring_1 [new solution_1 (S1)]. Figure 4(e) shows off spring_2 [new solution 2 (S2)] Off spring_1 and Off spring_2 are produced due to BX. Crossover is an efficient

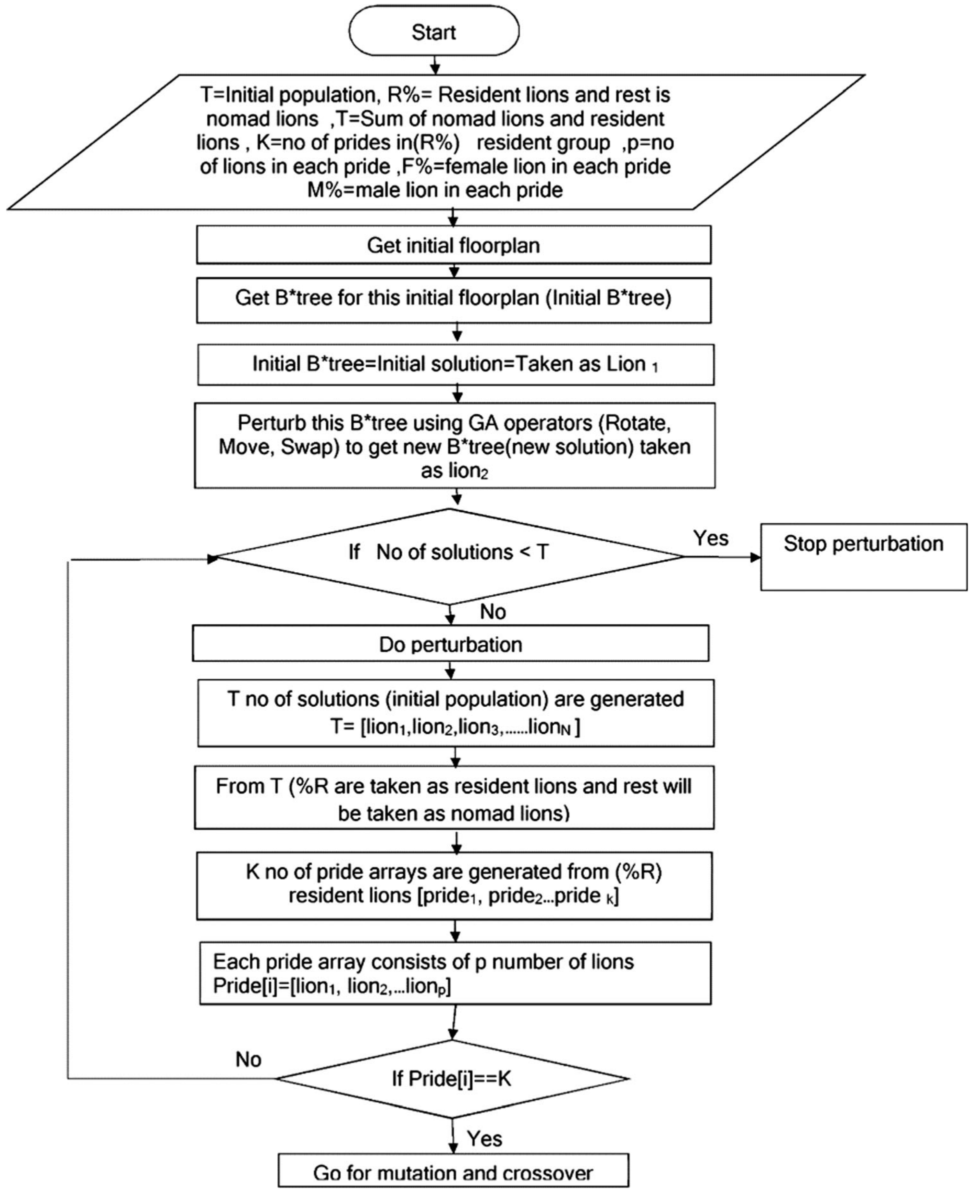


Fig. 3 Flow chart for initial population generation

and effective technique for optimization problems and it can combine two B*tree (parents) for producing two more B*tree (offsprings) but perturbation can introduce only one new B*tree. Perturbation leads to exploration whereas crossover leads to exploitation. The idea behind this crossover is that the new B*tree may be better than the initial B*tree (B*tree after perturbation) because many feasible combinations are produced due to this. Crossover operation leads to fast convergence, which is intended to pull the solution towards near to optimal solution. In this LOA, crossover is performed between the members in the same pride, since number of lions in a pride is taken as p, 2p number of offsprings are produced for a single pride.

3.2.3 Kill weak/sick off springs

Fitness of these offsprings are calculated using fitness function in Eqs. (1) and (2). Based on the fitness function new solutions (offsprings) are sorted for the next operation namely, territorial defense and takeover. p number of offsprings are sorted and beyond this p value, the remaining off springs undergo probability check and accepted as nomad lions and added with initial population (T). This sorting remembers battle between new mature lion offsprings. Then p number of new offsprings are compared with existing. If the new offspring (B*tree) dominates the existing members then replace parents with their new offspring. Weak offsprings (weak/sick offsprings) are removed from their pride based on the fitness value.

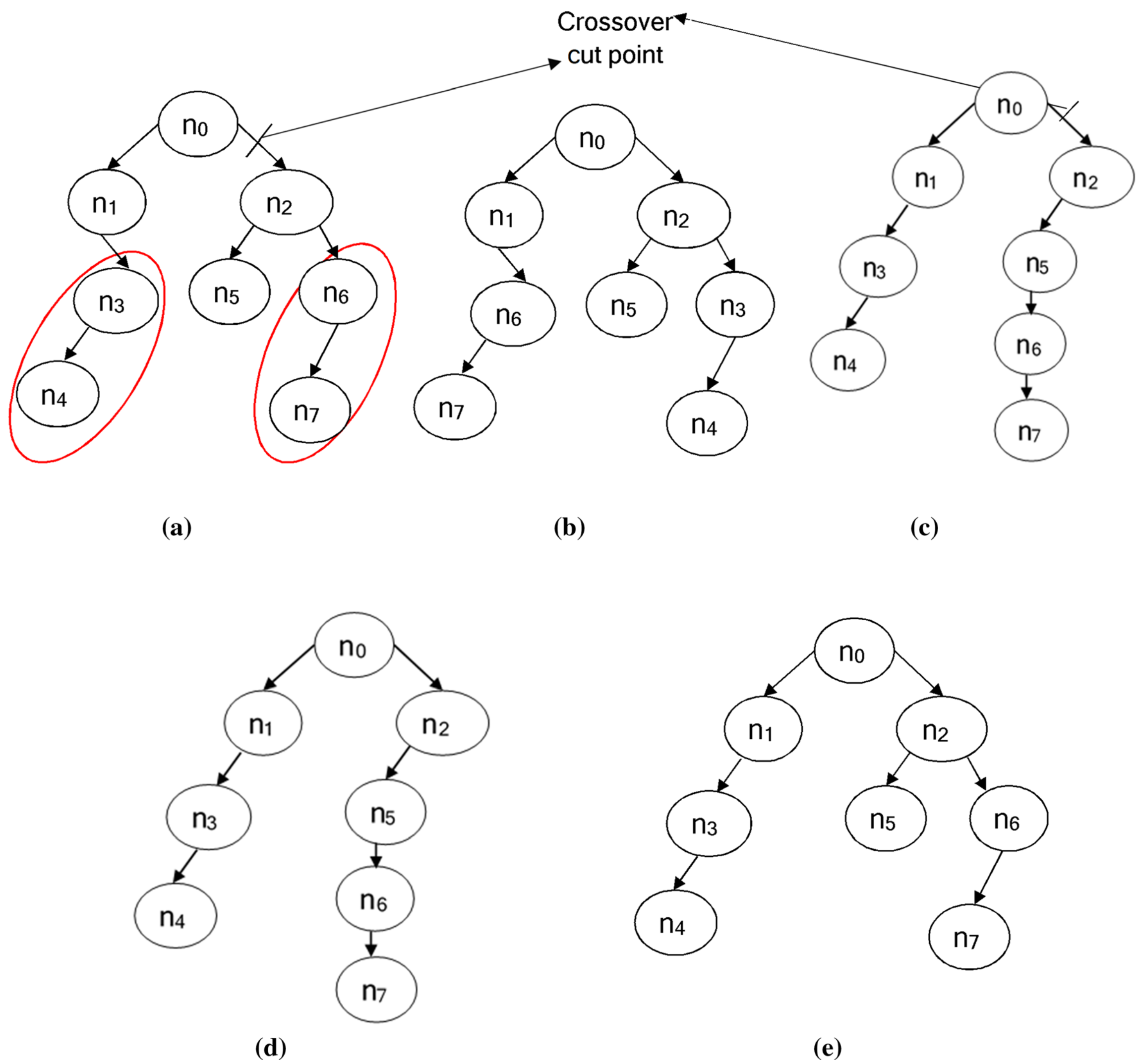


Fig. 4 B*tree crossover in LOA **a** Initial B*tree (Female lion). Encircled portion shows the subtrees need to be swapped. **b** 2nd B*tree—after swapping the circled sub trees. **c** 3rd B*tree (Male lion)—Swapping again to its original position (right/left). **d** offspring_1 and **e** offspring_2

3.2.4 Update pride

The above process computed in one iteration. Constant population of lions should be maintained in every pride for each and every iteration. Number of lions in each and every pride should be p. Equilibrium should be maintained at the end of each iteration in lion's each pride and also in initial population (T). This can be done by removing nomad lions with least fitness value. This process refers battle between nomads and resident male lions which is explained in territorial defense and takeover.

3.3 Territorial defense and takeover

Maziar and Fariborz [18] proposed that for each and every pride, battle starts when male offsprings become mature. New mature male offsprings fight with other male offsprings in the same pride. Defeated male cub is driven out from that pride and becomes a nomad and rest of the female lions and their female offsprings and male lions live together until a new nomad male lion made a battle with resident males. In this fight, nomad male wins, then it becomes a resident male and take over that pride and the defeated resident male is driven out as nomad. In this takeover, the male lion kill all the existing offsprings. There are two kinds of territorial defense and territorial takeover. First one is (1) between new mature male offsprings in the same pride and the other one is (2) between resident lions and nomad. This kind of territorial defense and takeover takes place after mutation and crossover. At the end of this territorial defense and territorial takeover, Lion Optimization Algorithm (LOA) identifies dominated B*tree (male lions) as optimal or acceptable solutions that play an important role in LOA. Figure 5 shows flow chart for territorial defense and takeover.

3.4 Equilibrium and stopping criteria

Always equilibrium should be maintained at each iteration of lion's initial population. This can be performed by removal of nomad lions with least fitness value. Stopping criteria could be based on the maximum number of iterations produced good results and also number of iterations produced bad results (based on fitness function/cost function). CPU time is also considered as stopping criteria for this Lion optimization algorithm.

Best male lion is selected in each and every iteration, If fitness value stagnates for designated number of iterations. In addition, with the above, fixed number of generations reached by this LOA is also considered as stopping criteria.

4 Experimental results and discussions

The proposed LOA algorithm is implemented in C++ programming language on Intel Pentium-D 3.20 GHz IBM machine. In order to verify the correctness and efficiency of this developed Lion Optimization Algorithm, it is tested with commonly used benchmark circuits namely, MCNC (Microelectronics Center of North Carolina) and Gigascale Systems Research Center (GSRC) benchmark to find the floorplan solution with the main objectives of minimizing the wirelength and area with the fixed-outline constraint. MCNC benchmark varies in size from 9 to 49 modules. GSRC bench mark (n100, n200, n300) contains modules 100, 200 and 300 respectively. Table 1(a), (b) show the characteristics of MCNC and GSRC. Using each benchmark circuitry of MCNC, LOA is executed more than 500 times to produce optimal solutions and 100 feasible solutions are tabulated for best/average solutions. Generally,

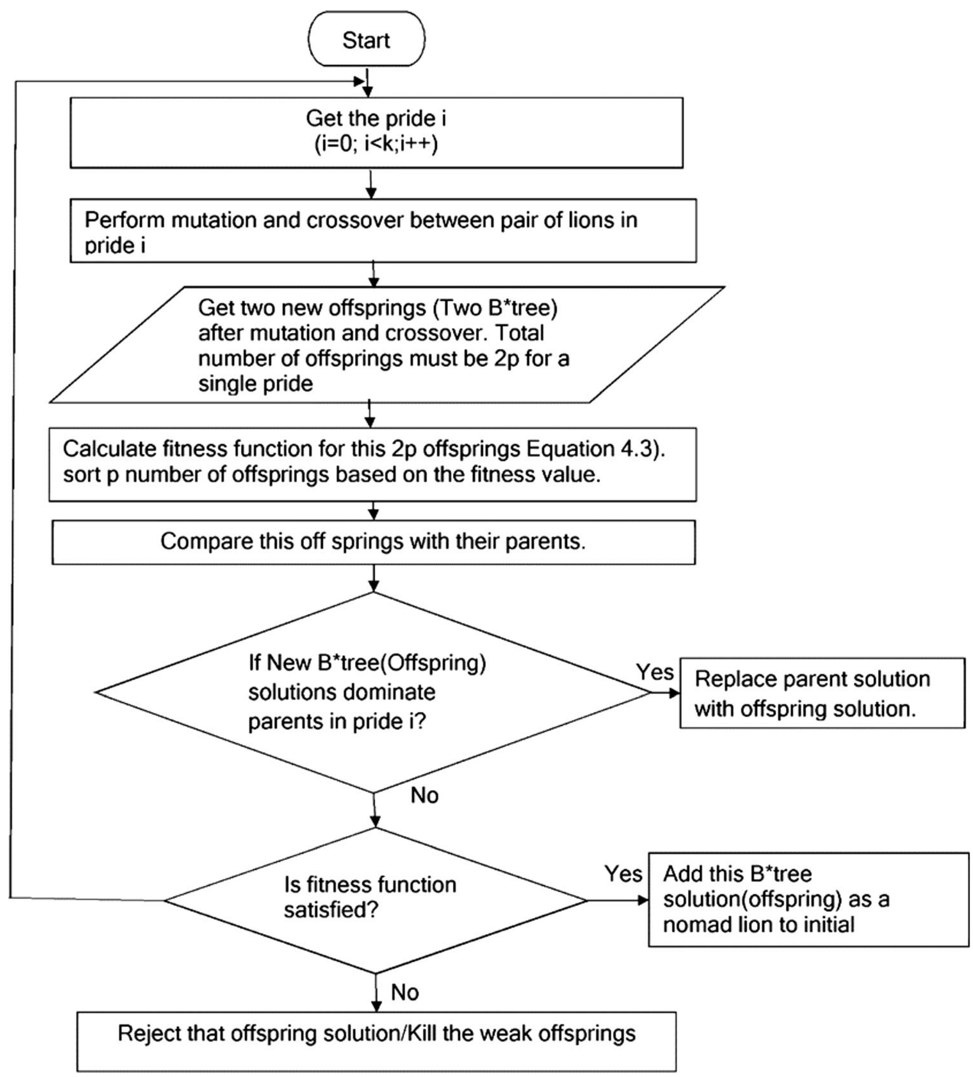


Fig. 5 Flow chart for territorial defense and takeover

Table 1 Details of MCNC and GSRC bench marks

MCNC bench mark	Modules	#Nets	#I/O pads	#Pins
apte	9	97	73	287
xerox	10	203	2	698
hp	11	83	45	309
ami33	33	123	42	522
ami49	49	408	22	953
GSRC benchmark			Number of modules	
n100			100	
n200			200	
n300			300	

GA population was set to 100 and mutation and crossover rate was 0.5 and 0.95 but for proposed LOA, population size was set to 10n and the probabilities for crossover and mutation were both 0.5 because, it was suitable combination found for LOA. Evidently LOA better performs, than any other optimization algorithms due to B*tree crossover.

It could be seen from Table 2; LOA results are tabulated for average and best/minimum solutions. Table 2 evaluates the quality of LOA for each MCNC benchmark problems. Performance of LOA is compared with following state-of-art algorithms, Thermal aware floorplan based Genetic Algorithm proposed by Hung et al. [15], B*tree based Fast-SA proposed by Chen and Chang [7], Genetic Algorithm proposed by Singha et al. [22], B*tree based Simulated spheroidizing annealing algorithm (SSAA) proposed by Anand et al. [2], and Improved simulated spheroidizing annealing algorithm by Anand et al. [2], B*tree-based Variable-Order Ant System (VOAS) proposed by Hoo et al. [13], Particle Swarm Optimization–Genetic Algorithm (PSO-GA) based Hybrid algorithm proposed by Sivaranjani and Senthil Kumar [23] in terms of area and wirelength. Acceptable solutions tabulated in Tables 2 and 3 are directly taken from respective published reports for these floorplanning algorithms mentioned above. Proposed LOA is applied on horizontally compacted floorplans. All the modules are considered as hard IP modules and aspect ratio is set as unity during this LOA simulation. Tables 2 and 3 clearly show that LOA has better solution searching ability for floorplan problems. LOA gives equal importance to both area and wire- length minimization objectives. So weightage factors α1 and α2 were set as 0.5 each (α1 = 0.5 and α2 = 0.5). Proposed LOA produces significant results in wirelength minimization and area minimization for MCNC bench marks.

Table 3 clearly explained that area minimization is achieved by LOA for benchmark apte, 2.27% reduced when compared to Fast-SA [7]; 0.44% reduced when compared to ISSAA [2]. For benchmark ami33, area minimization is achieved by 0.06% while compared to Fast-SA [7]; 0.03% reduced when compared with ISSAA [2]; 0.06% reduced when compared with GA and 0.01% reduced when compared with PSO-GA [23] based Hybrid algorithm. For benchmark ami49, area minimization is achieved by 1.91% compared with Fast-SA [7]; 1.05% reduced when compared with GA [22]; 0.15% reduced when compared with PSO-GA [23] based hybrid algorithm. LOA produces better results for ami33 and ami49 which are highest bench marks in MCNC. Table 4 clearly explained that wirelength minimization is achieved by LOA, for benchmark apte, 91% reduced when compared with ISSAA [2]; 24% reduced compared with Fast-SA [7]; 46% reduced when compared with PSO-GA [23] based hybrid algorithm. For the benchmark xerox, wire length minimization is achieved by 42.56% when compared with VOAS [13]; 50.6% reduced when compared with ISSAA [2]; 62.96% reduced when compared with GA [22]; 84.11% reduced when compared with Fast-SA [7]; 59.96% reduced when compared with PSO-GA [23] based hybrid algorithm. For benchmark hp, wire length minimization is achieved by 6.85% compared with VOAS [13]; 10.1% reduced when compared with ISSAA [2]; 71.4% reduced when compared with Fast-SA [7]. Benchmark ami33, wire length minimization is achieved by 7.28% for VOAS [13];

Table 2 Average area, average wire legth, minimum area and minimum wirelength of proposed LOA floorplanner for MCNC Benchmark Circuits when α1 = 0.5 and α2 = 0.5

MCNC benchmark circuits	Average area (mm^2)	Best/min area (mm^2)	Average wirelength (mm)	Best/min wirelength (mm)
apte	48.03	46.58	316.49	263.75
xerox	20.5	19.7	337.04	263.8
hp	9.89	9.44	142.95	126.74
ami33	1.23	1.16	52.22	38.48
ami49	38.45	35.64	822.77	641.84

Table 3 Average area comparison of proposed LOA for MCNC benchmark with various algorithms when α1 = 0.5 and α2 = 0.5

Algorithms	Proposed LOA floorplanner		B*tree based fast-SA	B*tree based ISSAA	B*tree based VOAS	Genetic algorithm	PSO-GA based hybrid algorithm
	Avg area (mm^2)	Min/best area (mm^2)	Area (mm^2)	Area (mm^2)	Area (mm^2)	Area (mm^2)	Area (mm$^{2)}$
apte	48.03	46.58	50.3	48.47	47.05	46.9	47.44
xerox	20.5	19.7	20.41	20.42	20.45	20.2	20.2
hp	9.89	9.44	9.6	9.40	9.40	9.85	NR
ami33	1.23	1.16	1.29	1.26	1.26	1.29	1.24
ami49	38.45	35.64	40.36	37.76	37.82	39.5	38.6

NR data not reported in literature

Table 4 Average wire length comparison of proposed LOA for MCNC benchmark with various algorithms when α1 = 0.5 and α2 = 0.5

Algorithms	ProposedLOA floorplanner		B*tree based VOAS	B*tree based ISSAA	Genetic algorithm	B*tree based fast-SA	PSO-ga based hybrid algorithm
	Avg WL (mm)	Min/best WL (mm)	WL (mm)	WL (mm)	WL (mm)	WL (mm)	WL (mm)
apte	316.49	263.75	246	408.47	191	541.43	463
xerox	337.04	263.8	379.6	387.64	500	421.15	497
hp	142.95	126.74	149.8	153.05	68.3	214.35	NR
ami33	52.22	38.48	59.5	49.28	46.2	59.96	48.4
ami49	822.77	641.84	667	824.49	912	816.51	673

WL wire length, *NR* data not reported in literature

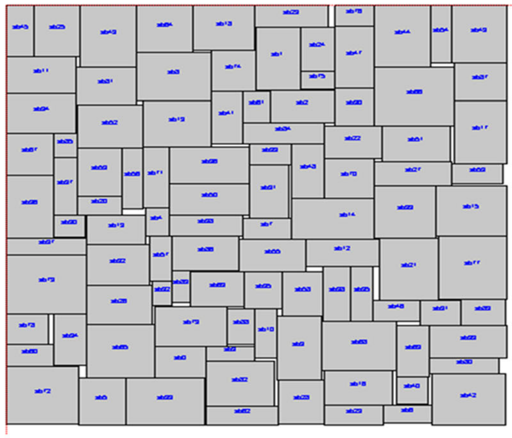
Table 5 Comparison of proposed LOA with other GA algorithms for improvement in area minimization from best to average solutions when $\alpha 1$ = 0.5 and $\alpha 2$ = 0.5

MCNC benchmark	Improvement in area minimization compared to genetic algorithm from LOA Best to avg (in %)	Improvement in area minimization compared to PSO-GA based hybrid algorithm from LOA Best to avg (in %)	Improvement in area minimization compared to thermal aware floorplan based GA from LOA Best to avg (in %)
ami33	0.13–0.06	0.08–0.01	0.11–0.04
ami49	3.86–1.05	2.96–0.15	3.52–0.71

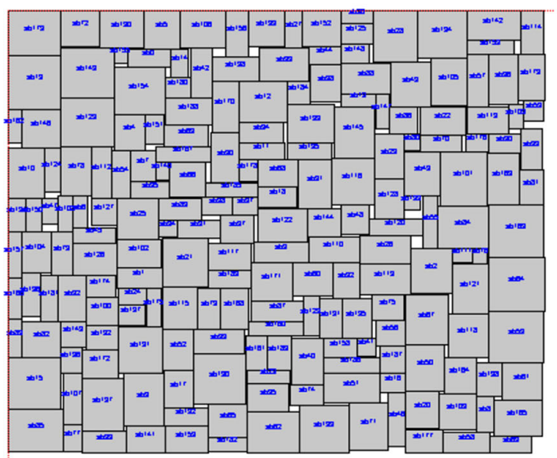
Table 6 Area and dead space comparison of proposed LOA floorplanner for GSRC bench mark when α1 = 1 and $\alpha 2$ = 0

GSRC benchmark circuits	Proposed LOA floorplanner				B*tree based improved simulated spheroidizing annealing algorithm (ISSAA)			
	Avg area (mm^2)	Best area (mm^2)	Avg DS (%)	Best DS (%)	Avg area (mm^2)	Best area (mm^2)	Avg DS (%)	Best DS (%)
n100	0.18	0.18	1.44	1.33	0.18	0.18	2.04	1.96
n200	0.20	0.18	1.33	1.24	0.18	0.18	3.03	2.47
n300	0.34	0.28	1.31	1.23	0.28	0.28	3.90	3.18

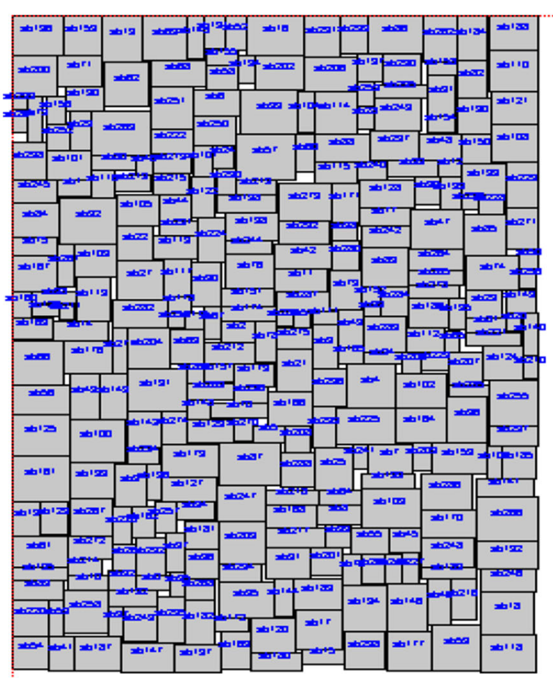
Avg average, *DS (%)* dead space reduction percentage in total area



(a) n100 with 1.44% Dead Space



(b) n200 with 1.33% dead space



(c) n300 with 1.31% dead space

◀**Fig. 6** Best layout structure for GSRC bench marks n100, n200 and n300

7.74% reduced when compared with Fast-SA [7]. For benchmark ami49, wire length minimization is achieved by 1.72% compared with ISSAA [2]; 89.23% reduced when compared with GA [22].

4.1 Performance of LOA with other GA algorithms

Proposed LOA is compared with other GA algorithms like Thermal aware floorplan-based GA proposed by Hung et al. [15], Genetic algorithm proposed by Singha et al. [22], PSO-GA Based Hybrid algorithm proposed by Sivaranjani and Senthil Kumar [23]. Performance results are tabulated in Table 5. From Table 5, Area minimization is improved for ami33 and ami49 compared to another GA [22]. Proposed LOA produces best results for highest bench mark circuits in MCNC. For apte, xerox and hp LOA produces equal results compared to SA but the algorithms based on B*tree using SA doesn't have property of B*tree crossover. B*tree crossover used in proposed LOA produces a balance between global exploration and local exploitation as compared to simulated annealing algorithm. Wirelength of LOA is improved by 89% compared to GA [22] and 31.16% is reduced when compared to PSO-GA Based Hybrid algorithm [23]. Wirelength for xerox is reduced by 162.96% compared to GA. MCNC benchmark xerox produces best optimal solution compared to other well-known algorithms.

4.2 Performance of LOA using GSRC

Table 6 compares the amount of dead space found in the floorplan created by LOA with improved SSAA proposed by Anand et al. [2]. It can be seen from Table 6; proposed LOA has 0.52–0.63% dead space minimization for n100; 1.14–1.23% for n200; 1.87–1.95% for n300 on GSRC benchmarks. Figure 6 shows the best layout structures of GSRC bench mark for n100, n200 and n300.

4.3 Performance analysis of LOA using Analyse-it

LOA experimental results are analyzed to prove the significance of proposed LOA using Analyse-it tool [31]. LOA results are compared with multistart Deterministic Algorithm (mDA), Memetic Algorithm (MA) [30] and Genetic Algorithm (GA). mDA, MA and GA results are directly taken from Tang and Yao [30]. A Kruskal–Wallis

Table 7 Kruskal–Wallis test results on area for the comparison between the mDA, GA, MA and LOA

MCNC benchmark	mDA			GA			MA			Proposed LOA floorplanner			*p*
	n	Rank sum	Mean rank	n	Rank sum	Mean rank	n	Rank sum	Mean rank	n	Rank sum	Mean rank	
ami33	30	1393.0	46.43	30	2237.0	74.57	30	465.0	15.50	30	406.0	14.5	< 0.0001
ami49	30	1103.5	36.78	30	2221.0	74.03	30	770.5	25.68	30	735.0	24.5	< 0.0001

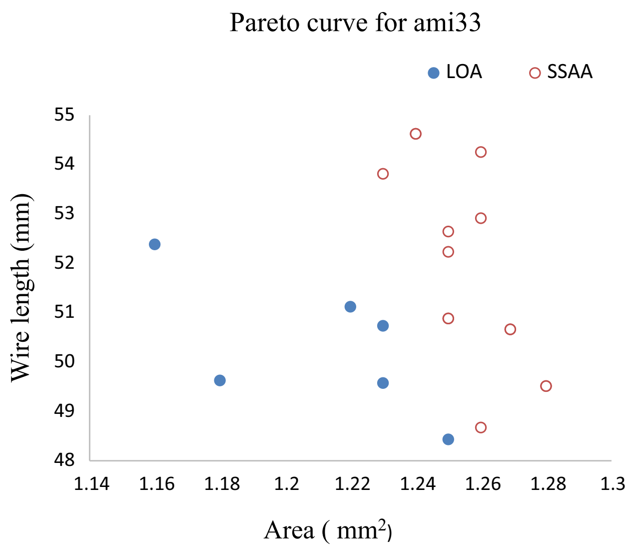


Fig. 7 Pareto-front for bench mark ami33

test on minimal area of 30 simulations has been performed on LOA. Simulation results are analyzed using a statistical software Analyse-it. The statistical test results are shown in Table 7.

4.4 Identification of pareto-front

Figure 7 shows the trade-off between the area and wire-length obtained using pareto-optimal front. A pareto-optimal set is defined as a set of solutions that are not dominated by any feasible member of the search space. The pareto-optimal solutions are optimal solutions of the multiobjective optimization problem. To evaluate the performance of proposed algorithms, a reference pareto-front is needed. This is generated using Simulated Spheroidizing Annealing Algorithm (SSAA). In SSAA, as single objective optimization, produces the solution in different runs for different weights. LOA has been applied to the benchmark ami33, and both the area and wirelength objectives are treated simultaneously as multiobjectives. Proposed LOA was tested for 10 trial runs with the parameters $\alpha 1 = 0.5$ and $\alpha 2= 0.5$. The best among the most repeated results of area and wire length for the given benchmark ami33 is given in Fig. 7. The pareto-front of LOA is then compared with SSAA. SSAA produced solutions without B*tree crossover, so it has less exploration capability, due to this the pareto-solutions of SSAA could be a scattered one. The proposed LOA method, due to its crossover operator, it produces more converged pareto-solutions than SSAA. The running time of the proposed method is comparatively lesser than that of well-known algorithms.

5 Conclusion

In this paper, a new optimization algorithm, Lion Optimization Algorithm has been presented for VLSI non-slicing floorplans with hard modules. The proposed LOA is developed by the combination of B*tree crossover with genetic algorithm. Furthermore, the algorithm is enhanced with improved stopping criteria. LOA proved that this B*tree crossover is an efficient and effective technique for floorplan optimization using GA. The reference pareto-front is compared with Simulated Spheroidizing Annealing Algorithm (SSAA) and the proposed method is found to perform better than SSAA. Multiobjective optimization enables evolution of trade-off solutions between different problem objectives through identification of the pareto-front. LOA provides very high computational efficiency and provides more exploration capabilities. This approach is quite flexible so that other formulations using different objectives and/or a larger number of objectives are possible. It is more flexible, easy to use, and easy to upgrade.

This LOA produces a balance between exploration and exploitation. LOA results are obtained using MCNC and GSRC benchmarks. Proposed LOA experimental results have been analysed using Analyse-it tool also. Statistical results show that proposed LOA produces significant results in wirelength minimization and area minimization for MCNC bench marks and also produces better results in dead space minimization for GSRC benchmarks while comparing the literature reports of well-known algorithms. There are several ways to extend this LOA in future. A detailed study of inserting this B*tree crossover into other bio inspired algorithms as, Particle Swam Optimization (PSO), Ant Colony Optimization etc. could be the future direction of this research. The authors are currently working in this direction.

References

1. Alpert, C. J., Mehta, D. P., & Sapatnekar, S. (2008). *Handbook of algorithms for physical design automation* (1st ed.). Auerbach, Boston: CRC publishers.
2. Anand, S., Saravanasankar, S., & Subbaraj, P. (2012). Customized simulated annealing-based decision algorithms for combinatorial optimization in VLSI floorplanning problem. *Computational Optimization and Applications, 52*(3), 667–678.
3. Areibi, S. (2001). Recursive and flat partitioning for VLSI circuit design. In *Proceedings of the 13th international conference on microelectronics* (pp. 237–240).
4. Bongard, J. (2009). Biologically inspired computing. *IEEE Computer Journal, 42*(4), 95–98.
5. Cohoon, J. P., & Paris, W. D. (1987). Genetic placement. *IEEE Transactions on CAD of Integrated Circuits and Systems, 6*(6), 956–964.
6. Chen, G., Guo, W., & Chen, Y. A. (2010). A PSO-based intelligent decision algorithm for VLSI floorplanning. *Soft Computing, Fusion, Foundation Methodologies and Applications, 14*(12), 1329–1337. https://doi.org/10.1007/s00500-009-0501-6.
7. Chen, T. C., & Chang, Y. W. (2006). Modern floorplanning based on fast simulated annealing. *IEEE Transactions on CAD, 25*(4), 510–522.
8. Christine Valenzuela, L., & Wang, P. Y. (2000). A genetic algorithm for VLSI floorplanning. In *Proceedings of the 6th international conference on parallel problem solving from nature* (pp. 671–680).
9. Christine Valenzuela, L., & Wang, P. Y. (2002). VLSI placement and area optimization using a genetic algorithm to breed normalized postfix expressions. *IEEE Transactions on Evolutionary Computation, 6*(4), 390–401.
10. David, E. G. (1989). *Genetic algorithms in search optimization and machine learning* (p. 41). Boston: Addison Wesley.
11. Deb, K. (2001). *A multi-objective optimization using evolutionary algorithms*. Hoboken: Wiley.
12. Forbes, N. (2000). Biologically inspired computing. *Computing in Science & Engineering, 2*(6), 83–87.
13. Hoo, C. S., Jeevan, K., Ganapathy, V., & Ramaiah, H. (2013). Variable-order ant system for VLSI multiobjective floorplanning. *Applied Soft Computing, 13*(7), 3285–3297.
14. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
15. Hung, W. L., Xie, Y., Vijaykrishnan, N., Addo-Quaye, C., Theocharides, T., & Irwin, M. J. (2005) Thermal-aware floorplanning using genetic algorithms. In *Sixth international symposium on quality electronic design (isqed'05)*. https://doi.org/10.1109/isqed.2005.122.
16. Lin, T., Chen, D. S., & Wang, Y. W. (2004). Robust fixed-outline floorplanning through evolutionary search. In *ASP-DAC'04: proceedings of the 2004 Asia and South Pacific design automation conference* (pp. 42–44). IEEE Press, Piscataway.
17. Mazumder, P., & Rudnick, E. M. (1999). *Genetic algorithms for VLSI design, layout and test automation*. Upper Saddle River, NJ: Prentice Hall.
18. Maziar, Y., & Fariborz, J. (2016). Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering, 3,* 24–36.
19. Rajakumar, B. R. (2012) The Lion's Algorithm: a new nature-inspired search algorithm. In *Proceedings of 2nd international conference on communication, computing & security (ICCCS-2012)* (Vol. 6, pp. 126–35). Procedia Technology.
20. Rechenberg, I. (1973). Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering.
21. Sait, S. M., El-Maleh, A. M., & Al-Abaji, R. H. (2003). General iterative heuristics for VLSI multiobjective partitioning. In *Proceedings of the international symposium on circuits and systems* (pp. 497–500).
22. Singha, T., Dutta, H. S., & De, M. (2012). Optimization of floorplanning using genetic algorithm. *Procedia Technology, 4,* 825–829.
23. Sivaranjani, P., & Senthil Kumar, A. (2012). Thermal-aware non-slicing VLSI floorplanning using a smart decision-making PSO-GA based hybrid algorithm. *Circuits, Systems, Signal Processing.*. https://doi.org/10.1007/s00034-015-0020-x.
24. Shahookar, K., & Mazumder, P. (1990). GASP: A genetic algorithm for standard cell placement. In *EURO-DAC'90: Proceedings of the conference on European design automation* (pp. 660–664). Los Alamitos, CA, USA: IEEE Computer Society Press.
25. Sherwani, N. A. (1999). *Algorithms for VLSI physical design automation*. Norwell, MA: Kluwer Academic Publishers.
26. Shadbolt, N. (2004). Nature inspired computing. *IEEE Journal on Intelligent Systems, 19*(1), 2–3.
27. Souza, J. G., & Costa, J. A. F. (2009). Unsupervised data clustering and image segmentation using natural computing techniques. In *IEEE international conference on systems, man and cybernetics* (pp. 5045–5050).
28. Storn, R. (1996) On the usage of differential evolution for function optimization. In *Biennial conference of the North American fuzzy information processing society (NAFIPS)* (pp. 519–523).
29. Tang, M., & Sebastian, M. (2005). *A genetic algorithm for VLSI floorplanning using O-tree representation, applications on evolutionary computing*. Lecture Notes in Computer Science (pp. 215–224). Berlin/Heidelberg: Springer.
30. Tang, M., & Yao, X. (2007). A memetic algorithm for VLSI floorplanning. *IEEE Transactions on Systems Man Cybernetics, 37*(1), 123–135.
31. The analyse-it general statistics software for Microsoft excel. Retrieved July 5, 2006, from http://www.analyse-it.com.
32. Wong, D., & Liu, C. (1989). Floorplan design of VLSI circuit. *Algorithmica, 4*(4), 263–291.
33. Wu, G. M., Wu, S. W., Chang, Y. W., & Chang, Y. C. (2000). B*-trees: A new representation for non-slicing floorplans. In *Proceedings of the 37th design automation.*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

M. Shunmugathammal was born in 1982, she received her Bachelor degree in Electrical and Electronics from Government College of Engineering, MS University Tirunelveli in 2003 and master degree in VLSI Design from Anna University Chennai in 2009. She works as an Assistant Professor at Cape Institute of Technology, Tirunelveli and doing her research under Anna University Chennai. Actually, her research interests are in Physical design of VLSI circuits, especially in floorplanning and placement also in meta-heuristic algorithms.

C. Christopher Columbus In 2011, Christopher Columbus. C received his Ph.D. degree in National Institute of Technology (NIT), Trichy. His research interests are in Parallel Computing, Distributed Computing, Optimization Techniques, Power System Operation. Now he is a professor in Department of Computer Science Engineering at PSN College of Engineering and Technology, Tirunelveli. He was in research project for AICTE under MODROBS scheme and completed the project in 2013.

S. Anand In 2012, Anand. S received his Ph.D. degree from National Centre for Advance Research in Discrete Mathematics (n-CARDMATH). He received his Master (M.E.) degree in VLSI Design, and Bachelor degree (B.E.) in Electronics and Communication from Anna University Chennai. His research focus is broadly in the area of VLSI physical design and evolutionary algorithms. Specifically, he has worked extensively in Floor planning problems, Multi-objective and Parallel Computing Methods. Now he is a professor at Rajagiri School of Engineering and Technology, Kerala, in the Department of Electronics and Communication.