

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Performance Comparison of PSO and Its New Variants in the Context of VLSI Global Routing

---

Subhprattim Nath, Jamuna Kanta Sing and  
Subir Kumar Sarkar

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.72811>

---

## Abstract

Substantial reduction of gate delay occurred in recent times owing to radical decrement of transistor size. The interconnect length and delay are accordingly increased owing to the exponential escalation of packaging density with additional transistors being fabricated on the same chip area. The function of VLSI routing that seems to be more defying to the scholars, is categorized in global routing and detailed routing phase. In global routing phase, the prevalent method to lessen the wire length for reducing interconnect delay is to adjust the cost of the Steiner tree, devised by the terminal nodes to be interconnected. Nevertheless, Steiner tree problem is a NP-complete problem in classical graph theory where meta-heuristics might impart beneficial elucidations. Particle swarm optimization (PSO) is a robust algorithm concerning VLSI routing field. This chapter is regarding the proposal of a self-adaptive mechanism for monitoring acceleration coefficient of PSO and evaluating its functionalities with the existing acceleration coefficient controlled PSO in numerous allocation topologies of terminal nodes within definite VLSI layout. The outcomes of PSO variant with constriction factor in context to VLSI route reduction ability and robustness are also inspected. Additionally, a new effort in adapting the PSO with embracement of genetic algorithm is established.

**Keywords:** VLSI, global routing, Steiner tree, meta heuristics, PSO

---

## 1. Introduction

Stating optimization as a usual phenomenon is a bit of exaggeration, which includes economic development to engineering strategy as well as job scheduling to resource allocation. The intention of optimization is certainly to produce comparable outcomes under specified conditions bearing some parametric minimization or maximization. In VLSI physical design context

numerous parameters are needed to be optimized like transistor count, power delay product, interconnect delay.

Conferring to Moore's Law, the transistor count doubles up in every 18 months [1]. Since the dimensions of the transistors are radically diminishing by the contemporary ages as a result of massive development in technologies, additional transistors are getting assimilated in that single chip region than before by means of cutting-edge assembling techniques. Therefore the length of interconnect has also been considerably amplified. Previously it was sufficient to overhaul the gate delay but interconnect delay has been more noticeable after the 130 nm technological node was pioneered.

The objective of VLSI physical design is to optimize the devices arrangements and interconnection schemes among these devices for desired performance.

Wire-length approximation of interconnects is considered in the routing phase of VLSI physical design process, which is largely categorized into Global Routing and Detailed Routing. In Global routing phase the circuit interconnections in shortest possible wavelength and minimum interconnect delay is required to be achieved. The complexities of global routing problem is solved to some extent with sequential approach where VLSI nets are sequenced according to their criticality and practical routers employs improvement phase. Technique of rerouting after ripping interfering wires [2] and 'shove-aside' technique [3] and also introducing concurrent approach where parallel integer programming concept are tried for enhancement of global routing but with limitations.

The Routing problem of VLSI physical design can also be mapped in classical Graph Theory where wire-length minimization of interconnected nodes rests in solving the Rectilinear Minimal Steiner Tree Problem (RMST) [4], a renowned NP Complete problem of Graph Theory. Such NP complete problems can be aimed to solve by a division of Artificial Intelligence known as Swarm Intelligence. Swarms interact among themselves to persist in any situation. It has been observed that these social agents have restricted competences of their own as an individual, however as an assemblage they are capable of accomplishing an assignment, somewhat perceptively for their existence. Scientists and engineers were ardent to mimic these activities of these natural swarm systems. Swarm intelligence was maidenly commenced in 1989 by G Beni and J Wang [5] to crack some practical problems associated to global optimization. These algorithms are heuristics and meta-heuristics in character. Heuristic infers "to ascertain by trial and error". These approaches are fairly beneficial in obtaining optimal solution or near optimal solution aimed at a complex optimization problem (like NP-complete problems) within a judicious time frame. Meta-heuristics ("meta" means "beyond" or "higher level") conversely execute even better than heuristic algorithms herein because they comprise of precise compromises related to randomization & local search. One such prevalent meta-heuristics algorithm is PSO (Particle Swarm Optimization) [6].

In VLSI system for solving the Routing problem the researcher Dong et al. [7] used PSO in 2009. The proposal by the authors was mainly novel encoding and updating schemes for a discrete or binary version of PSO where Interconnect delay is one of the most important disadvantage. A routing scheme based on PSO combined with buffer insertion at suitable intervals, was taken into consideration for overcoming the disadvantage, as proposed by Ayob

et al. [8]. Again a PSO algorithm was proposed by Liu et al. [9] to reduce binding during routing where this version of PSO algorithm was successful in reducing the binding problem but in turn it increased the cost of the wire lengths. Among many other proposed algorithm, the one proposed by Shen et al. [10] has some important significance which dealt with the self-adaptive technique of inertia weight update. Many other technique based on SI has been used for escalation of VLSI routing, among them the one proposed by Arora and Moses is important. Both Manhattan as well as a non- Manhattan routing scheme based on Ant Colony Optimization (ACO) [11] were proposed by the duo. A proposal was introduced by Ayob et al to evade VLSI routing scheme by using Firefly optimization [12].

Two algorithms centered on inertia weighted PSO (PSO-W) [13] are presented in here. In the prior one, named as Self-Adaptive acceleration coefficient PSO (PSO-SAAC), the acceleration coefficients are adjusted by the means of an adaptive procedure of local search and global search to heighten the property of the searching technique composed with efficient converging rate. Additional new tactic is implemented where the perception of the genetic algorithm is hybridized with PSO-W by comprising a component related with a breeding factor in the position update characteristic equation of PSO-W. In supplement to the above two alternatives of PSO, PSO with constriction factor (PSO-C) [14] are evaluated with these algorithms, most lately familiarized, on a mutual platform of optimization of VLSI global routing. Further all the algorithms are verified in several distribution topologies of VLSI terminal nodes in a definite search area.

The remaining chapter is arranged accordingly. Section 2 portrays the elementary theory of VLSI Global routing, RMST and PSO. In Section 3 algorithms variants on PSO-W are illustrated in specifics shadowed by the implementation of modified PSO algorithms in Section 4. Section 5 confers the experimental results acquired in context to VSLI global routing and lastly the chapter gets concluded with Section 6.

## 2. Preliminaries

### 2.1. Global routing in VLSI physical design

VLSI routing in Physical Design context initiates with the procedure of interconnections amongst the circuit blocks and pins, specified according to the net list, generate results in the phase of placement. The inputs to the general routing problem are as follows:

- Net list.
- Timing budget for the critical nets.
- RC delay of per unit length of metal layers and vias.

Conventionally, the routing fashion can be broadly cleft into two main stages: the initial stage, also called as the “Global Routing” allocates a list of routing areas for individual net, putting aside the absolute geometrical blueprint of wires; whereas, the secondary stage, called as “Detailed Routing”, finds the absolute geometrical blueprint of any net within the allocated routing areas.

The purpose of the routing problem is to curtail the wire length, at the same time accommodating the timing budget for the critical nets. Global routing is the initial phase of routing

where a catalogue of routing regions are essentially allocated for a net, in default of stipulating the tangible geometric layout of interconnects, acknowledging that no net could be crisscrossing each other.

Now, the complications related to global routing can be overcome using two basic approaches—the sequential approach and the concurrent approach.

The term ‘sequential’ in sequential approach refers to something which occurs in a sequence of steps; and here as the name indicates, nets are connected in succession. This approach of routing is very susceptible to the order in which the nets are considered for routing, because a net once routed may hinder other nets. The parameters based on which the nets are ordered includes half of the total area of the confining polygon, their criticality, and the number of terminals. High criticality number is assigned to the net on the critical path as the performance of the circuit is dependent on them to a great extent. But nonetheless these techniques of sequencing are not impeccable. As a blunt aftermath of this, a factual router engages a development or improvement phase besides the sequencing phase to do away with the jam when no more routing is possible. But still this may not conquer the frailty of the sequential approach. ‘Rip-up and reroute’ [2] and ‘shove-aside’ techniques [3] are examples of improvement phase like this. As the name indicates, in case of the ‘rip-up and reroute’, the intrusive cables are ripped up and rerouted; but in case of the ‘shove aside’ technique, cables which make is viable to outright the failed connections are put aside without curbing the extant connections in any way.

In Concurrent approach, by simultaneously considering all the nets, the complications related to ordering can be evaded. There is no compelling polynomial algorithm (not even for nets with only two terminals) and as a consequence this approach is computationally tougher. Ergo, it was proposed to use integer programming; nonetheless as a consequence of the very large size of the resulting program, it can’t be utilized efficiently. So, to overcome this problem, the program is fragmented into smaller sub programs using hierarchical method (which follows top down approach), which can be then readily dealt with by integer programming. A sequence of routing channels is accredited to each net by global routing without contravening the capacity of channels. At the same time, the total length of the wire is also occasionally revamped.

The intention of the routing problem is reliant on the attributes of IC which is to be fabricated. Nowadays, VLSI chips contain up to billions of transistors which makes it possible to complete a layout to route millions of nets. In turn there may be several thousands of routes in each net. It comprises the trade-off between routability of all nets and minimization of the wire length in interconnects. The objective function of this routing puzzle is an eminent NP-complete problem. All these attributes results the routing problem to be a computationally tough one which furnishes the scope for metaheuristic algorithm, like for resolving the problem, SI based algorithm are to be used.

## 2.2. Rectilinear Steiner tree problem

Rectilinear Steiner Tree (RST) is the resulting of Minimum Spanning Tree (MST) for a specified grid graph. For a certain set of vertices of a graph, minimum spanning tree is shaped by interconnecting them where MST is the maximal sub graph and the MST cost is the sum of the weights of all edges in the tree. Certain additional intermediate vertices are appended with MST



to create Steiner tree in turn to lessen the entire length of the MST where intermediate vertices are Steiner points. The challenges remain in selecting the number and position of Steiner vertices in the Grid layout. Steiner tree with restricted edge to be in rectilinear in rectangular grid graph originates RST. This tool is utilized to acquire the least possible length of the inter connections for a specified set of nodes in the grid graph. RST, although NP complete problem [15], is an efficient method in improving the length of interconnects in VLSI circuits.

Nonetheless some other constraints for instance noise, power, electro-migration, signal integrity, packaging density, skew, inductance, reliability etc., frequently have vast impact on the objective function in deep-submicron VLSI design, the length of the non-critical nets however preserves its significance in wire length minimization.

### 2.3. Particle swarm optimization

PSO is a multi-agent equivalent search technique which engages incorporates an iterative method to obtain the ideal solution in a multi-dimensional search space. Assume that there exists a  $d$  dimensional search space, where the number of agents arbitrarily allocated are  $n$ . The agents are primed with certain position and velocity vectors as  $X_i = \{X_1, X_2, \dots, X_n\}$  and  $V_i = \{V_1, V_2, \dots, V_n\}$  respectively. These vectors are periodically updated rendering to the characteristic equations of PSO as given in Eqs. (1) and (2).

$$V_{i,t+1} = V_{i,t} + c_1 * r_1 * (p_{best} - X_{i,t}) + c_2 * r_2 * (g_{best} - X_{i,t}) \quad (1)$$

$$X_{i,t+1} = X_{i,t} + V_{i,t+1} \quad (2)$$

Where the constants  $c_1$  and  $c_2$  are accountable for the impact of the distinctive particle's individual information and so as of the group information correspondingly. The variables  $r_1$  and  $r_2$  are unvaryingly distributed random numbers [16]. All particles are adjusted randomly and keep on promoting the fitness value influenced by the  $p_{best}$  value (best position value of the individual) and  $g_{best}$  value (best position value of the entire swarm) correspondingly in anticipation of the optimal solution to be accomplished.

### 2.4. PSO parameters

#### 2.4.1. Swarm size

Swarm size infers the number of particles existing in the swarm. A huge number of particles can exploit a huge extent of a search space; therefore fewer iterations are required so as to achieve the optimal solution. Contrariwise an enormous swarm size upsurges the computational complexity and time complexity likewise.

#### 2.4.2. Iteration number

Number of iterations is a problem contingent parameter related to swarm size. An inadequate number of iterations can terminate the program precipitately earlier to the conjunction whereas huge number of iterations generates a redundant computational and time complexity.

### 2.4.3. Velocity component

The velocity update factor in Eq. (1) comprises of three terms. The first term is the preceding velocity vector i.e. the former direction & the magnitude of the velocity of a particle. This component averts a particle from a radical alteration in velocity in current iteration. The following component is the cognitive one. It is centered on the memory of an agent in agreement with its proficiency. The cognitive component continuously inspire a particle to reappear to its position, suitable for it in local. The subsequent component is the social component. This is the knowledge to an individual by social communication, which constantly encourage the particle to travel in the direction of the best position, knowledgeable by its vicinity.

### 2.4.4. Acceleration coefficients

The variables  $c_1$  and  $c_2$  are known as acceleration coefficients, which attempt to generate an equilibrium amid the cognitive component and social component of the velocity.

- If  $c_1 = c_2 = 0$ , Eq. (1) will be  $V_{i,t+1} = V_{i,t}$ . This implies that all the particles retain to hover with their initial velocity, ensuing no search condition.
- If  $c_1 > 0$  and  $c_2 = 0$ , Eq. (1) resolves to  $V_{i,t+1} = V_t + c_1 * r_1 * (p_{best} - X_{i,t}) + 0$ . This implies that all the particles revolve around their searching space autonomously. Since they are not interacting with the neighbours, they are incapable to obtain the global optimal solution whatsoever.
- If  $c_1 = 0$  and  $c_2 > 0$ , Eq. (1) resolves to  $V_{i,t+1} = V_t + 0 + c_2 * r_2 * (g_{best} - X_{i,t})$ . It infers that all particles are fascinated to a single point, which is not revised in each time step.
- If  $c_1 = c_2$ , all particles will travel towards average  $p_{best}$  and  $g_{best}$  values.
- If  $c_1 \gg c_2$ , it results in manipulating the particles in the direction of  $p_{best}$  position and  $c_2 \gg c_1$ , resulting the particles to be enticed towards the  $g_{best}$  position and in both circumstances the particles sprint precipitately to the optimum solution.

Usually  $c_1$  and  $c_2$  are considered as equal, constant values and various intellectual articles propose their values as 2 for getting decent optimal results [14].

## 3. Analysis of PSO characteristics & modification

### 3.1. Velocity clamping

Particle's velocity, a significant parameter of PSO algorithm, is the step size of swarm in every iteration. With all time step, the particles alter their velocity & travel in all direction in the problem space. If the velocity is extreme, the assessment attribute of the particle turn out to be high and simultaneously the particle might hastily vacant the periphery of the search space and swerve. On a converse if velocity is low, the movement of particles is limited upon a small boundary and it happens confined in a local optima. Therefore, it is required to preserve an equilibrium amidst exploration & exploitation by situating a parameter  $V_{max}$ , assumed as  $V_{max} = \frac{(X_{max} - X_{min})}{k}$ . The empirical value of  $k$  is set as 2 [14].

### 3.2. Inertia weight

Inertia Weight ( $w$ ) was additionally familiarized [13] progressing PSO-W to substitute  $V_{max}$ , to regulate the momentum of the particle in assessing the updated velocity. It is presented to regulate the exploration and exploitation aptitudes of the swarm with the intention of the algorithm to converge more efficiently upon time. Therefore Eq. (1) is adapted as Eq. (3).

$$V_{i,t+1} = w*V_{i,t} + c_1*r_1*(p_{best} - X_{i,t}) + c_2*r_2(g_{best} - X_{i,t}) \quad (3)$$

- If  $w = 1$ : then Eq. (3) is similar to the original Eq. (1).
- If  $w > 1$ : then the velocity will increase over time and the particles will barely be capable of altering their direction.
- If  $w < 1$ : particles can rapidly alter their route subjective to pbest and gbest values.
- If  $w = 0$ : particles travel lack of any acquaintance of the preceding velocity.

Typically the inertia weight  $w$  is selected dependent to the size of the search space. A high value of  $w$  is essential for complex high dimensional problem space and trivial value for small dimensional search space.

The inertia weight can be differed by Eq. (4), where  $s$  is the population size,  $D$  is the Dimension size and  $R$  is relative quality of corresponding solution standardized to  $[0,1]$ .

$$w = \left[ 3 - \exp\left(\frac{-s}{200}\right) + \left(\frac{R}{8} * D\right)^2 \right]^{-1} \quad (4)$$

### 3.3. Constriction factor

The PSO algorithm is reorganized to substitute the inertia weight  $w$  & max velocity  $V_{max}$  by a fresh parameter  $\chi$ , known as constriction factor given in Eq. (6). Clerc [17] pioneered this factor, which proved to be exceptionally significant in regulating the exploration & exploitation trade-off, thus guaranteeing an efficient conjunction of algorithm. Eq. (1) gets amended as Eq. (5).

$$V_{i,t+1} = \chi * [V_{i,t} + \Phi_1 * (p_{best} - X_{i,t}) + \Phi_2 * (g_{best} - X_{i,t})] \quad (5)$$

$$\chi = \frac{2}{[2 - \phi - \sqrt{\phi^2 - 4\phi}]} \quad (6)$$

Here,  $\phi = \phi_1 + \phi_2$ ,  $\phi_1 = c_1 * r_1$  and  $\phi_2 = c_2 * r_2$ . Characteristically applying the value of  $\phi$  as 4.1 the value of  $\chi$  results to 0.729. Therefore,  $\chi * w = 0.729 * w < w$ , infers that the particles rapidly alter their course manipulated by  $p_{best}$  and  $g_{best}$  with assured convergence. Both  $(p_{best} - X_{i,t})$  and  $(g_{best} - X_{i,t})$  are multiplied by  $2 * 0.729 = 1.458$  [18]. Generally these values are preferred for improved stability and convergence.

### 3.4. Acceleration coefficient

Acceleration coefficient as a PSO parameter has previously been explained in the preceding segment. Usually both the values of  $c_1$  and  $c_2$  are applied to be 2 [19]. The equilibrium concerning



these parameters can be monitored in two distinct ways, mentioned underneath, to accomplish superior result in perspective of path minimization of VLSI global routing.

#### 3.4.1. Self-tuned

In this algorithm PSO-ST [20], the acceleration constants both  $c_1$  and  $c_2$  are reduced linearly throughout the time steps in the range of 2 to 1.49. At the beginning, the algorithm is primed with  $c_1=c_2=2$ . By this changing of linear decrement, both exploration and exploitation abilities of the swarm can be preserved efficiently for velocities updating and can deliver a swift convergence to the algorithm. This algorithm turns out to be competent to obtain optimal result with lofty convergence rate.

#### 3.4.2. Self-adaptation

An algorithm PSO-SAAC is introduced where the two acceleration constant parameters  $c_1$  and  $c_2$  have been assorted in such a style that they got enhanced influence over the trade-off in between global exploration and local exploitation. The algorithm commences with highest exploration and lowest exploitation aptitudes of swarm, which have eventually been altered in every time step over the entire iteration process. Therefore the particles of the swarm are capable of dispersing all over the search space consistently, motivated by the social component of the velocity vector at the first phase of experiment. Since the cognitive component outpace the social component in the subsequent phase of the experiment, the swarm accomplish the local search process entered on the assessed results of the Global search process with the intention of obtaining the finest local optima. Throughout the whole searching process this self-adaptive procedure can be effectual in producing most significant  $g_{best}$  value and thus in this manner heightening the optimization rate.

### 3.5. PSO with mutation

A fresh algorithm is presented where the principle of Genetic Algorithm is featured in PSO [21]. The algorithm after utilizing some time steps initiates with selection of swarms from existing generation in the first phase. The swarms with high fitness probability get selected where the probability of selection factor is  $\frac{f_j}{\sum_{j=1}^N f_j}$ , where N is the population size. The high fitness factor is extracted from the selected pool generating a mutant in the second phase. This enhanced knowledge of high fitness property is induced in the position vector Eq. (2) to evolve a new generation of swarms causing mutation in PSO [22]. The proposed position vector in Eq. (7) is given below.

$$X_{i,t+1} = (\psi * X_{i,t} + \xi) + V_{i,t+1} \quad (7)$$

where,  $\psi$  is the randomization factor and  $\xi$  is the mutant fitness factor.

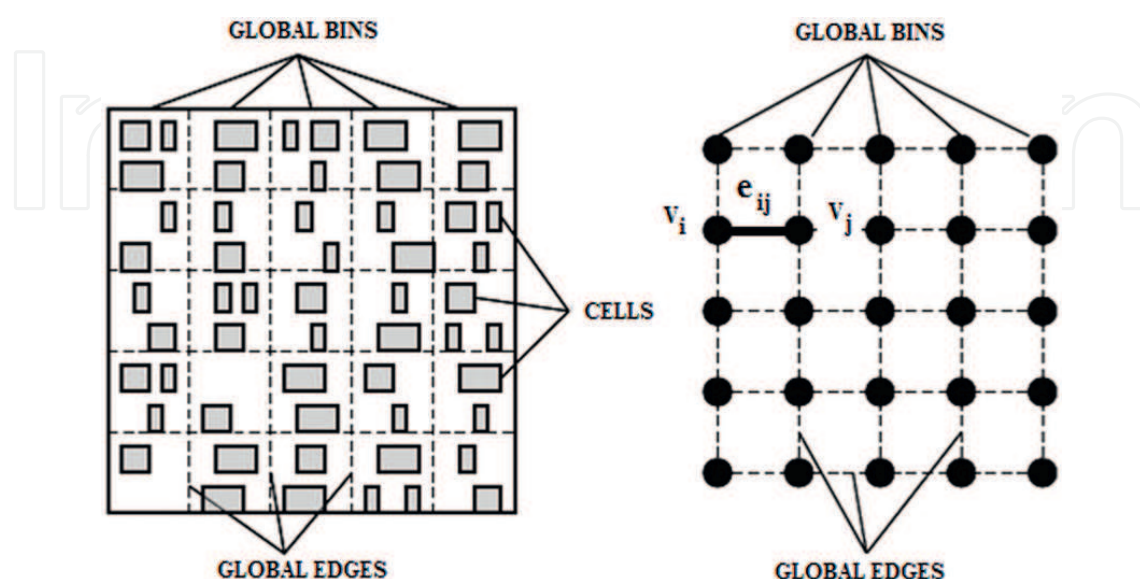
## 4. Problem formulation and implementation

The challenge of global routing problem of a VLSI Physical design rests in two objectives. First in minimizing power dissipation and secondly increase the speed of signaling among the

partitions or blocks in VLSI layout which can be achieved by reducing the complete wire length of interconnected terminals or blocks. The Global routing problem can be formally stated where  $N=\{N_1, N_2, N_3 \dots N_m\}$  be the set of nets denoting interconnections among blocks in the VLSI layout and the estimated wirelength of net  $N_i$ ,  $1 < i < m$  is denoted by  $D_i$ . The problem function can be expressed such that overall total wirelength  $\sum_{i=1}^m D_i$  is minimized. Global routing formulation is done by mapping the required VLSI layout in classical graph theory as Grid Graph model. Here the grid graph model is regarded as to execute the above proposed algorithms. The grid graph,  $G=(V, E)$ , is an exemplification of a routing region layout where region is carved into a number of unit square cells as shown in **Figure 1**. Each cell representing routing areas between blocks as empty area is signified by vertex a  $v_i$  and the edge  $e_{ij}$ , linking the two neighboring vertices  $v_i$  and  $v_j$ . The vertices resemble to the nodes and edges resemble to the routing paths between blocks in a VLSI layout.

To obtain the solution of the VLSI routing problem for a multi-terminal net, the primary assignment is to articulate it as the problem of obtaining an RMST (Rectilinear Minimum Spanning Tree) from a Graph. The Minimum spanning tree of the interconnected terminal nodes is generated using graph algorithms results in measuring the minimum cost of interconnected length. With introduction of random Steiner nodes along with the terminal nodes of multi-terminal VLSI layout the cost or the overall wirelength is further reduced generating the minimum Steiner tree cost (length) in the graph. Depending on the position and the number of Steiner nodes the cost or overall length can be further minimized. With large number of terminal nodes the probability of determining the number of Steiner nodes and desired positioning of these Steiner node become computationally hard and hence the PSO algorithm is used to select probable number of Steiner nodes and generate these random position in order to optimize the Steiner cost.

The algorithm commenced with random generation of swarms size of  $z$  particles and they are placed in required graph of  $n \times n$  dimension. Each of the swarm consists at most  $(p-2)$  randomly generated Steiner points drawn from Steiner set  $S$  with  $(n||2-p)$  points where  $p$  is



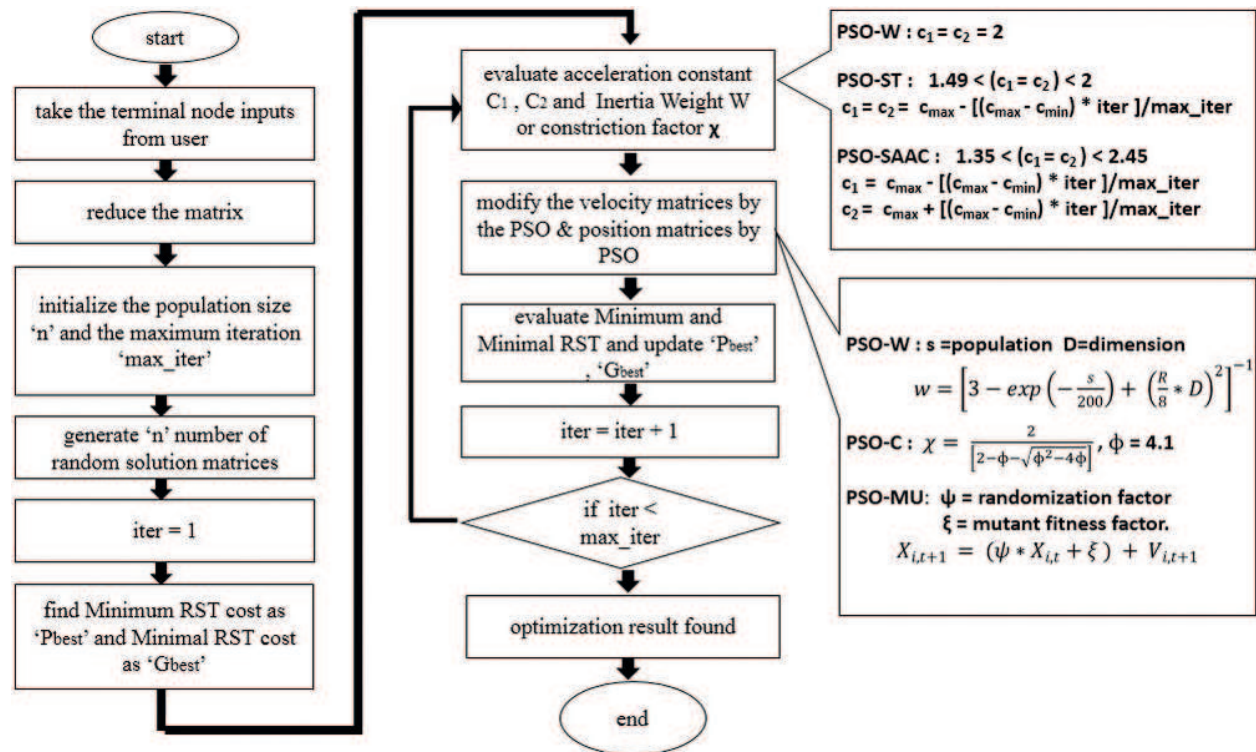
**Figure 1.** Routing region layout in grid graph.

the number of terminal nodes with designated vertex  $V_i$ ,  $i = \{1, 2, 3, \dots, r\}$  and thereby Steiner subset  $Q^j \subseteq S$ , where  $j = \{1, 2, 3, \dots, z\}$  is formed.  $100 \times 100$  search space is used. The defined destination nodes or terminal nodes are represented by 1 to generate the problem matrix. Rows & Columns without the destination nodes are eliminated to reduce computational complexity generating the reduced matrix Steiner points are introduced in a randomly in the problem space, is denoted by 1. The reduced matrix and the corresponding Steiner matrix, are shown in **Figure 2**.

For implementation of PSO, mapping is done with the creation of swarm particles where each these Steiner matrix is considered as a particle. One such particle with the destination nodes is shown is **Figure 3**. Fitness  $F_i$  for each particle seed is calculated by evaluating Minimum Rectilinear Steiner Tree (MRST) cost using objective function  $MST(G_i)$  and also  $MIN(MST(G_i))$  which

$$x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

**Figure 2.** Matrix generated from reduced graph and corresponding Steiner matrix.



**Figure 3.** Flow chart of PSO algorithm and its variants.

is minimum among all  $MST(G_i)$  is calculated. PSO parameter are initialized and maximum iterations are fixed. Evaluation of  $p_{best}$  and  $g_{best}$  values are made using PSO velocity equations with acceleration coefficient tuned either in linear decreasing or in self-adaptive mode as described earlier and the corresponding position equation either in classical mode or with mutation factor is evaluated. When maximum iteration is reached the corresponding  $g_{best}$  value generated or the best swarm particle, is the optimized RMST cost. The optimized RMST cost on termination of the PSO algorithm is the minimum overall length of the interconnected terminal nodes in the VLSI system and thereby minimum wire length routing path of VLSI layout is achieved. The flow chart of the PSO algorithms and its variants are shown in **Figure 3**.

The pseudo code of the PSO algorithm and modifications for implementation is given below.

Input:

*Search space and terminal nodes are defined*

*Swarm size and Max-iterations are defined*

Initialization:

*Generate an initial population of particles  $X_i = \{X_1, X_3, \dots, X_n\}$*

*Calculation of  $f(X|i)$  and  $\text{MIN}(f(X|i))$*

Output:

*Optimized result of  $\text{MIN}(f(X|i))$*

Begin:

*While ( $t < \text{max iter}$ )*

*Evaluate  $c_1$  and  $c_2$  according to any of the variants mentioned in this chapter.*

*Evaluate Inertia Weight ( $w$ ) as in Eq. (4) or evaluate Constriction Factor ( $\chi$ ) as in Eq. (6)*

*Set  $p_{best} = f(X|i)$  and  $g_{best} = \text{MIN}(f(X|i))$*

*for  $i=1: n$  (for particles)*

*Calculate particle velocity  $V_{i,t+1}$  according to the velocity equation as in Eq. (3) or Eq. (5)*

*Update the particle position  $X_{i,t+1}$  in accordance to position equation as in Eq. (2) or*

*Update the particle position as in Eq. (7)*

*Evaluate  $f(X|i)$  and  $\text{MIN}(f(X|i))$*

*Update  $p_{best}$  and  $g_{best}$*

*end for  $n$*

*$t = t + 1$*

*end while*

*Post processing the results and visualization*



5. Experimental results and discussion

Two coordinate sets of 15 terminal nodes are randomly created based on varied distribution topology of terminal nodes in VLSI system on a defined two dimensional  $100 \times 100$  search space. Coordinate sets for nearly Uniform distribution and Bivariate distribution are graphically represented in **Figures 4** and **5** respectively.

Experiments on all the algorithms are performed 25 times for each of these coordinate sets of varied distribution topologies in VLSI system. The population size of the swarms has been set as 100 and maximum iteration of 75 is used for all the algorithms.

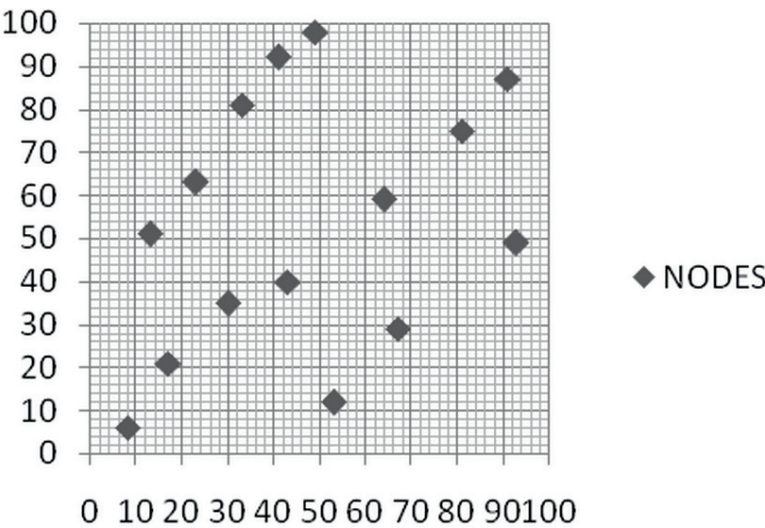


Figure 4. Nearly uniform distribution of terminal nodes on  $100 \times 100$  search space.

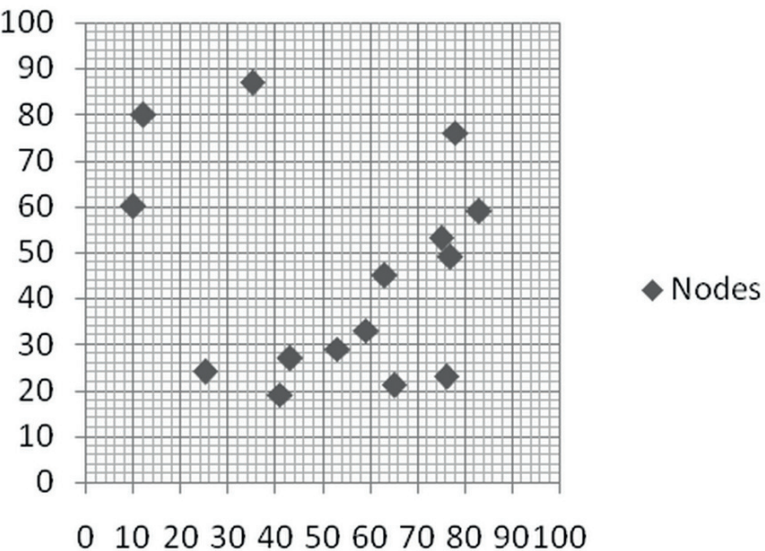


Figure 5. Bivariate distribution of terminal nodes on  $100 \times 100$  search space.



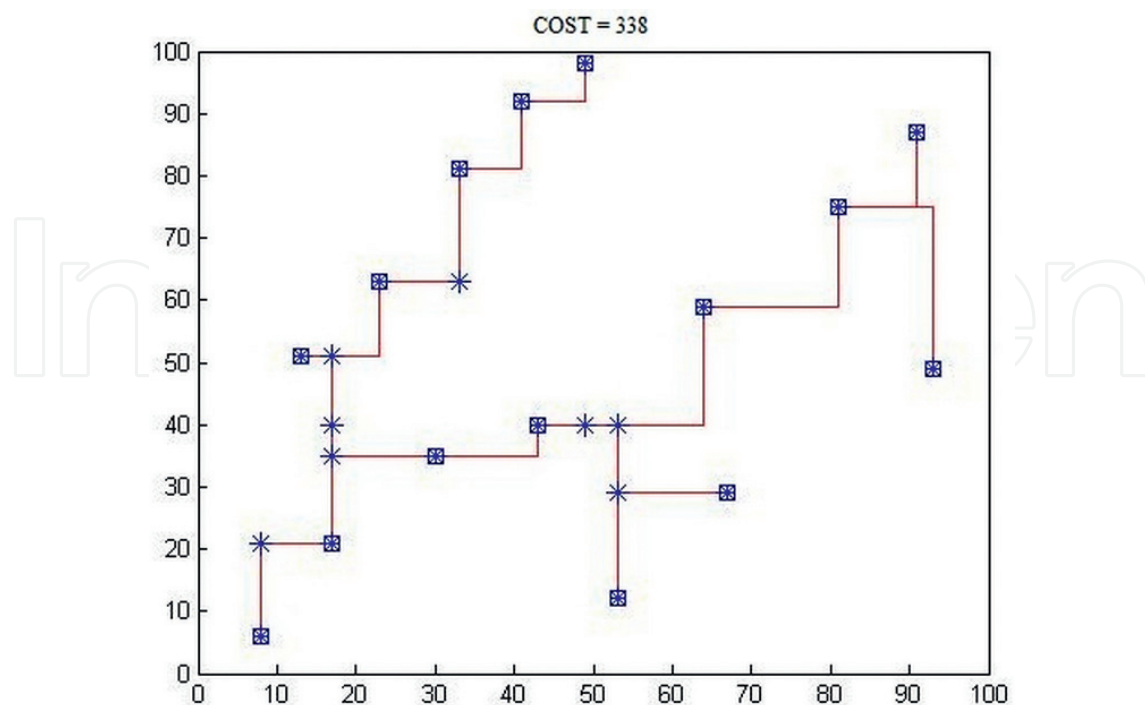
## 5.1. Experiment A

Experiments on PSO-W as well as on the modified algorithms PSO-ST [20] and PSO-SAAC are performed separately for two said coordinate sets to interconnect the terminal points for each set and to return the minimum cost of interconnection correspondingly. The minimum interconnection VLSI global routing cost, the average interconnection VLSI global routing cost over the 25 simulations of the algorithms and the corresponding standard deviations are recorded. The results of average  $g_{best}$  and minimum  $g_{best}$  are summarized in **Table 1**.

For nearly uniform distribution of terminal nodes in VLSI layout, PSO-SAAC works best in compared to the other two algorithms. In SET 1 '338' is achieved as the minimum interconnection global cost value for PSO-SAAC, given in **Figure 6**. From **Table 1** it can be analyzed that for bivariate distribution of terminal nodes in VLSI layout, self-tuned acceleration constant controlling mechanism for PSO-ST outruns the other two algorithms. In random uniform distribution environment, PSO-ST generates lowest minimum interconnection cost as 253,

| Test case | $g_{best}$ value | PSO-W | PSO-ST | PSO-SAAC |
|-----------|------------------|-------|--------|----------|
| SET 1     | Average          | 354.5 | 348.4  | 341.7    |
|           | Minimum          | 350   | 343    | 338      |
| SET 2     | Average          | 256.7 | 254.9  | 257.3    |
|           | Minimum          | 253   | 253    | 255      |

**Table 1.** Comparison of PSO-W with PSO-ST and PSO-SAAC.



**Figure 6.** Minimum 'cost' Steiner tree obtained for PSO-SACC in SET 1.



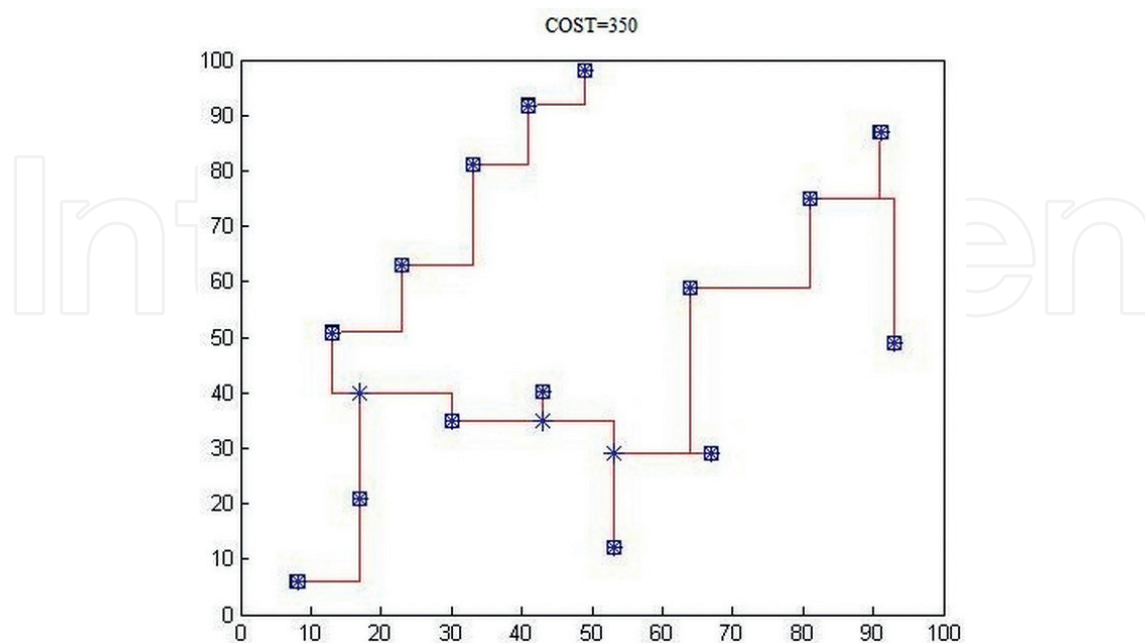
coordinate set which have been considered. The results of minimum interconnect cost, average cost and average execution time of all algorithms are recorded on **Table 3**.

The Minimum Rectilinear Spanning Tree (RMST) for minimum interconnect cost generated for the said VLSI topologies in case of all algorithms are shown in **Figures 8, 9, and 10**. It reveals that the algorithm PSO-MU generate lowest minimum global best value as well as minimum mean value in all two coordinate sets. This indicates that this algorithm PSO-MU, in comparison to PSO-W and PSO-C, ensure efficient VLSI global routing cost minimization and better convergence.

It is observed from **Table 3** that for Coordinate Set 1,  $g_{best}$  value of PSO –MU is found to be 329 where execution time of this algorithm is much greater than PSO-W. Runtime of PSO-C is found to be 101.51 compared to 85.48 for PSO-MU algorithm. This implies that PSO-MU

| Test case | $g_{best}$ value | PSO-W  | PSO-C  | PSO-MU |
|-----------|------------------|--------|--------|--------|
| SET 1     | Average          | 354.5  | 350.4  | 336.8  |
|           | Minimum          | 350    | 345    | 329    |
|           | System time      | 52.825 | 101.51 | 85.48  |
| SET 2     | Average          | 256.7  | 256    | 250.4  |
|           | Minimum          | 253    | 254    | 248    |
|           | System time      | 49.05  | 86.01  | 66.96  |

**Table 3.** Comparison of PSO variants over average, minimum  $g_{best}$  value and system time.



**Figure 8.** Minimum 'interconnection cost' Steiner Tree obtained for PSO-W in SET 1.

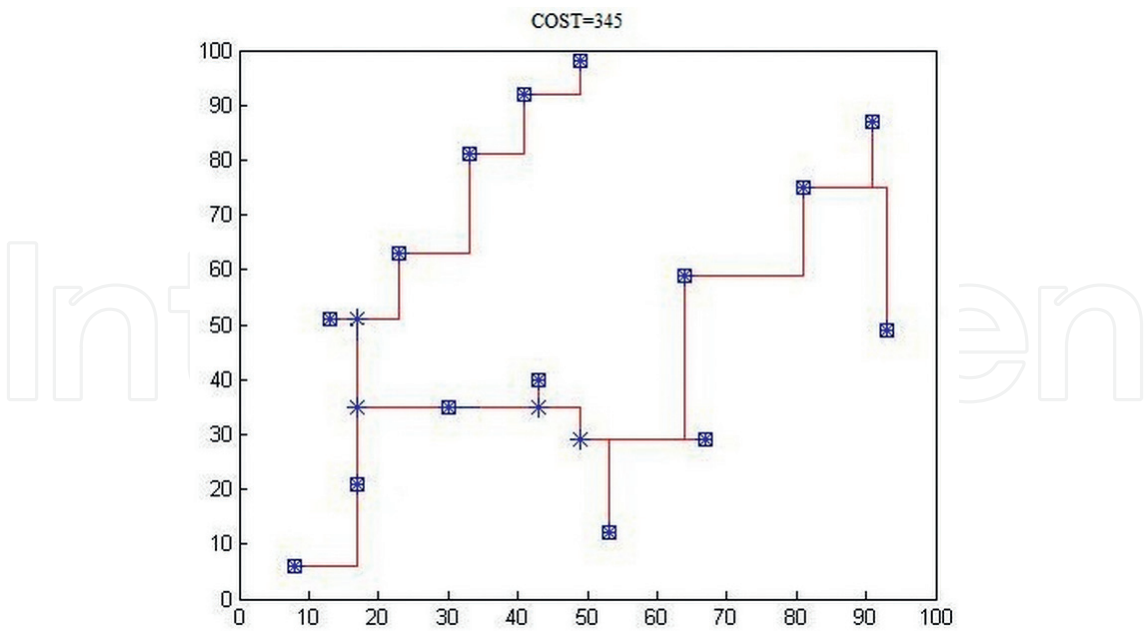


Figure 9. Minimum ‘interconnection cost’ Steiner Tree obtained for PSO-C in SET 1.

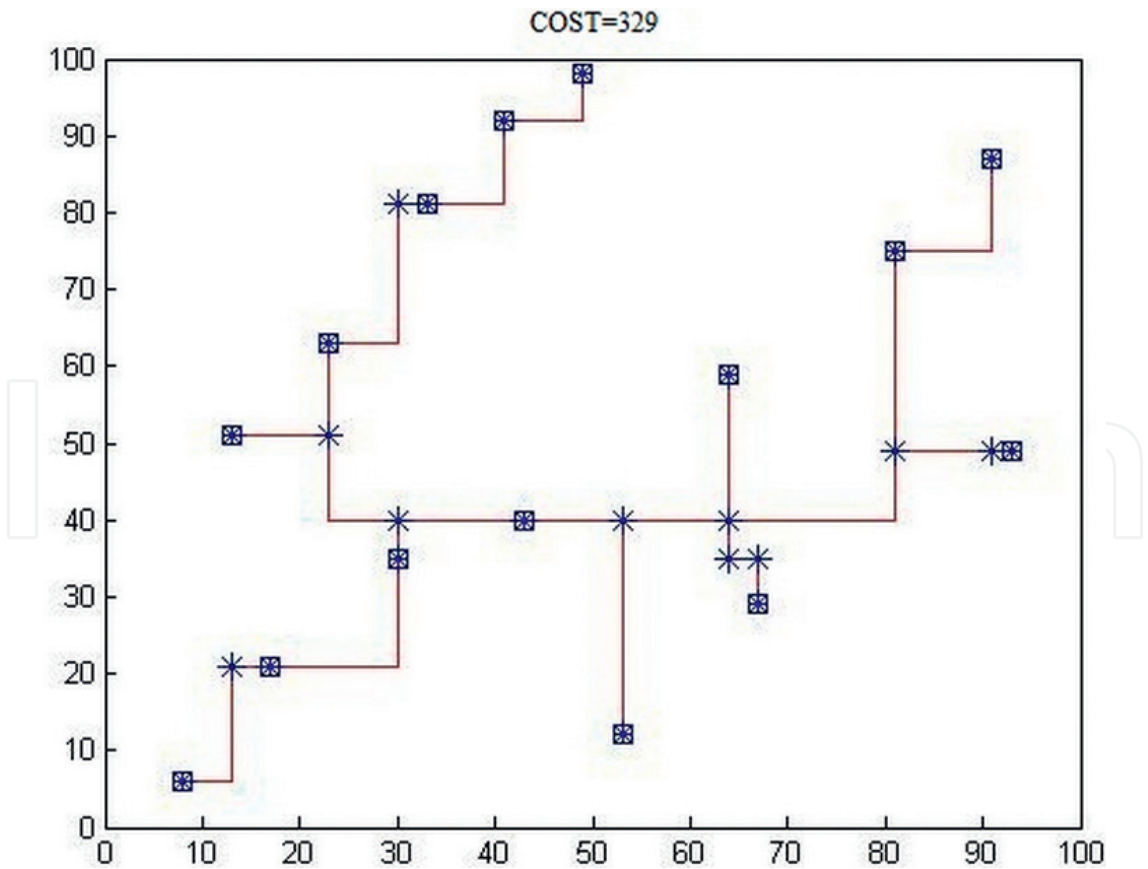


Figure 10. Minimum ‘interconnection cost’ Steiner Tree obtained for PSO-MU in SET 1.

| Test case | PSO-W | PSO-C | PSO-MU |
|-----------|-------|-------|--------|
| SET 1     | 7.77  | 0.71  | 5.65   |
| SET 2     | 1.94  | 1.88  | 3.83   |

**Table 4.** Standard deviation of  $g_{best}$  values.

algorithm outperforms the performance of conventional PSO-W and PSO-C algorithm while reducing the Timing budget with respect to PSO-C algorithm in context to VLSI global routing.

In order to analyze the consistency of these algorithms, the standard deviation (SD) values are calculated for all algorithms on each of the two coordinate sets and are recorded in the **Table 4**. SD value of PSO-C is found to be 0.71 and 2.25 for the two coordinate sets. These values are much lower compared to all other SD values of PSO-W and PSO-MU ensuring robustness although sacrificing system execution time of the algorithm independent of the distribution complexities of the search space in VLSI layout. This implies that PSO-C although generates higher value of global routing interconnection cost as well as system execution time compared to PSO-W and PSO-MU, it exhibits robustness of the algorithm throughout all varied distribution topologies of the terminal nodes in the said VLSI layout.

## 6. Conclusion

This chapter intends variants developed on Particle Swarm Optimization algorithm to resolve the global routing problem in VLSI domain. Simultaneously the controlling of acceleration constant in PSO has been verified for the VLSI routing problem. Lastly, a proportional analysis is done amongst the pre mentioned algorithms beside three variants of PSO, which have been recognized as decent routing algorithms in VLSI design. Researches are piloted to inspect the optimization property, rate of convergence, computational time and robustness of the algorithms including the ways by which algorithms work proficiently in problem space with dissimilar distributive topologies of VLSI layout.

The outcomes demonstrates that from the standpoint of topologically dissimilar problem spaces of VLSI domain, the general performance of PSO-ST [20] is very agreeable, however PSO-SAAC executes finest in an approximately uniform distributed problem space. It has also been observed that the performance of PSO-C and PSO-MU are unhampered of the diverse distribution of VLSI global routing problem space. The performance of the algorithm PSO-MU preserves a balance between the optimization and convergence rate. Although PSO-MU is realized to be steady in random problem space [22], PSO-C is appeared to be the best algorithm in the perspective of robustness.

Therefore the chapter indicated the exclusive merits and demerits of the PSO algorithm and its variants, well-matched for solving the wire-length minimization problem of global routing in VLSI physical design. It is projected that in the situation of VLSI global routing optimization, the paradigm of hybridization with essence of genetics can contest with the functioning of PSO conventional ones and can exhibit enhanced performance. Hence the global routing problem



in VLSI can be competently managed by contemporary PSO meta-heuristics and by hybridization of distinct swarm intelligence.

## Author details

Subhrapratim Nath<sup>1\*</sup>, Jamuna Kanta Sing<sup>1</sup> and Subir Kumar Sarkar<sup>2</sup>

\*Address all correspondence to: suvro.n@gmail.com

1 Department of CSE, Jadavpur University, India

2 Department of ETCE, Jadavpur University, India

## References

- [1] Moore GE. Cramming more components onto integrated circuits. *Proceedings of the IEEE*. 1998;**86**:82-85. DOI: S 0018-9219(98)00753-1
- [2] Bollinger H. A mature DA system for PC layout. In: *Proceedings of First International Printed Circuit Conference*. Lo Alamitos: IEEE Computer Society Press; 1979. pp. 85-99
- [3] Dees WA, Karger PG. Automated rip-up and reroute techniques. In: *Proceedings of Design Automation Conference*. Piscataway: IEEE Press; 1982. pp. 432-439. DOI: 10.1109/DAC.1982.1585535
- [4] Ho J-M, Vijayan G, Wong CK. New algorithms for the rectilinear Steiner tree problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1990;**9**(2):185-193. DOI: 10.1109/43.46785
- [5] Beni G, Wang J. Swarm intelligence in cellular robotic systems. In: Dario P, Sandini G, Aebischer P, editors. *Robots and Biological Systems: Towards a New Bionics? NATO ASI Series (Series F: Computer and Systems Sciences)*. Berlin/Heidelberg: Springer; 1993. pp. 703-712. DOI: 10.1007/978-3-642-58069-7\_38
- [6] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: *Sixth International Symposium on Micro Machine and Human Science*, 4-6 October 1995, IEEE: Nagoya; 2002. pp. 39-43. 10.1109/MHS.1995.494215
- [7] Dong C, Wang G, Chen Z, Sun S, Wang D. A VLSI routing algorithm based on improved DPSO. In: *IEEE International Conference on Intelligent Computing and Intelligent Systems ICIS*. Vol. 1. IEEE; 2009. pp. 802-805. DOI: 10.1109/MHS.1995.494215
- [8] Ayob MN, Yusof ZM, Adam A, Abidin AFZ, Ibrahim I, Ibrahim Z, Hani MK. A particle swarm optimization approach for routing in VLSI. In: *Second International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*. IEEE; 2010. pp. 49-53. DOI: 10.1109/CICSyN.2010.42

- [9] Liu G, Chen G, Guo W, Chen Z. DPSO-based rectilinear Steiner minimal tree construction considering bend reduction. In: Proceedings of Seventh International Conference on Natural Computation (ICNC). Vol. 2. IEEE; 2011. pp. 1161-1165. DOI: 10.1109/ICNC.2011.6022221
- [10] Shen Y, Liu Q, Guo W. Obstacle-avoiding rectilinear Steiner minimum tree construction based on discrete particle swarm optimization. In: Proceedings of Seventh International Conference on Natural Computation (ICNC). Vol. 4. IEEE; 2011. pp. 2179-2183. DOI: 10.1109/ICNC.2011.6022440
- [11] Arora T, Moses ME Ant colony optimization for power efficient routing in Manhattan and non-Manhattan VLSI architectures. In: Proceedings of Swarm Intelligence Symposium, SIS '09, March-April 2 2009. IEEE, pp.137-144. DOI: 10.1109/SIS.2009.4937856
- [12] Ayob MN, Hassan F, Ismail AH, Basri HH, Azmi MS, Abidin AFZ A firefly algorithm approach for routing in VLSI. In: Proceedings of IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), 3-4 December 2012 pp.43-47. DOI: 10.1109/ISCAIE.2012.6482066
- [13] Shi Y, Eberhart RC. Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation; 6-9 July 1999; IEEE: Washington, DC; 2002. p. 1945-1950. DOI: 10.1109/CEC.1999.785511
- [14] Clerc M, Kennedy J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation. 2002;6(1):58-73. DOI: 10.1109/4235.985692
- [15] Garey MR, Johnson DS. The rectilinear Steiner tree problem is NP-complete. SIAM Journal on Applied Mathematics. 1977;32(4):826-834. DOI: 10.1137/0132071
- [16] Carlisle A, Dozier G. An off-the-shelf PSO. In: Proceedings of the Workshop on Particle Swarm Optimization. Indianapolis: Purdue School of Engineering and Technology, IUPUI; 2001. pp. 1-6. DOI: 10.1.1.589.485
- [17] Valle YD, Venayagamoorthy GK, Mohagheghi S, Hernandez J-C, Harley RG. Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Transactions on Evolutionary Computation. 2008;12(2):171-195. DOI: 10.1109/TEVC.2007.896686
- [18] Eberhart RC, Shi Y. Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation; 27-30 May 2001 IEEE: Seoul; 2002. pp. 81-86. DOI: 10.1109/CEC.2001.934374
- [19] Jordehi AR, Jasni J. Parameter selection in particle swarm optimisation: A survey. Journal of Experimental and Theoretical Artificial Intelligence. 2013;25(4):527-542. DOI: 10.1080/0952813X.2013.782348
- [20] Ghosh S, Nath S, Sarkar SK. PSO algorithm with self tuned parameter for efficient routing in VLSI design. In: International Conference on Futuristic Trends in Computing and Communication; 29th April 2015; Chennai; 2015. SSRG International Journal of Electronics and Communication Engineering. pp. 60-63. DOI: ISSN: 2348-8549

- [21] Hassan R, Cohanin B, Weck OD, Venter G. A comparison of particle swarm optimization and the genetic algorithm. In: 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference; 18-21 April 2015. American Institute of Aeronautics and Astronautics: Austin; 2015. pp. 1-13. DOI: 10.2514/6.2005-1897
- [22] Nath S, Ghosh S, Sarkar SKA. Novel approach to discrete particle swarm optimization for efficient routing in VLSI design. In: 4th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions); 2-4 September 2015. IEEE: Noida; 2015. DOI: 10.1109/ICRITO.2015.7359375