Parallel Genetic Algorithm Based on Fuzzy Controller for Design Problems

Leonid Gladkov, Sergey Leyba, Nadezhda Gladkova and Andrey Lezhebokov

Abstract In this paper a method of joint solutions of placement and routing problems of digital equipment elements is offered. The authors suggested a new approach on the basis of evolutionary algorithm (EA) integration and a fuzzy control model of algorithm parameters. A fuzzy logical controller structure is described in the article. A model of parallel evolutionary algorithm is developed. To synchronize parallel computations, you proposed to use a modified migration operator. To confirm the method effectiveness a brief program description is reviewed.

Keywords Genetic algorithm • Fuzzy logic • Computer-aided design • Optimization • Parallel computing

1 Introduction

As a rule design problems are characterized by high computational complexity due to the search of huge number alternative solutions. Besides, to find the accurate solution it is necessary to carry out a exhaustive search, which is impossible. Design problems include partitioning problem, placement problem, routing problem etc. [1].

Placement and routing problems are the most important problems throughout the lifecycle duration. Traditionally, these problems are solved by different methods at different stages that lead to increase the time and computational complexity. So, a

L. Gladkov (✉) · S. Leyba · N. Gladkova · A. Lezhebokov
Southern Federal University, Rostov-on-Don, Russia
e-mail: leo_gladkov@mail.ru

N. Gladkova
e-mail: nadyusha.gladkova77@mail.ru

A. Lezhebokov
e-mail: legebokov@gmail.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_14

development of integrated methods for placement and routing problems seems appropriate at the present time. Such methods allow us to take into account constraints and current results during the problems solution [2].

To increase the problem solving effectiveness in terms of automated design of complex engineering systems, which contain a million components, it is useful to employ evolutionary and genetic algorithms [3].

A new development stage of genetic algorithm theory became hybrid systems. They are based on a combination of various scientific fields such as genetic algorithms, fuzzy systems and neural networks [4–9]. Currently mechanisms of parallelized evolutionary computations are widely used to effective computational resources management [10–12].

2 Problem Formulation

Let $E = \{e_i \mid i = 1, \ldots, N\}$ denote a set of elements, where $e_i = (l_i, h_i, T_i)$ is an element which should be placed and N is a number of elements. Here l_i is a length of the element, h_i is a height of the element and T_i is a list of pins which can be written as $\mathrm{Ti} = \{\mathrm{t_j} \mid \mathrm{j} = 1, \ldots, \mathrm{K}\}$.

Here t_j is a pin, K is a number of pins in the element. Each pin is described as $t_j = (x_j, y_j)$ where x_i, y_i are pin coordinates relative to the base point of the element.

The set of net that connected each element is defined as $\mathrm{U} = \{\mathrm{u_h} | \mathrm{h} = 1, \ldots, \mathrm{L}\}$, where u_h is a net, L is a number of nets.

The net is defined as $u_h = \{(Ne_k, Nc_k) | k = 1, \ldots, M\}$, where N_{ek} is a number of element, N_{ck} is a number of pin and M is number of pins connected by the net.

It is required to find such elements placement that $V = \{(x_i, y_i) | i = 1, \ldots, N\}$, where $(\mathrm{x_i}, \mathrm{y_i})$ are coordinates of upper left corner of *the* ith element.

For each net the contact list of connection field needs to be found.

$V = \{(x_i, y_i) | i = 1, \ldots, N\}, \mathrm{W_h} = \left\{\left(\mathrm{x_q}, \mathrm{y_q}\right) | \mathrm{i} = 1, \ldots, \mathrm{Q}\right\}$, where Q is a number positions through which passes the hth net.

3 Algorithm Description

For simultaneous solution placement and routing problems the parallelized genetic algorithm is used. It supposes a parallel implementation of evolutionary processes for several populations. The synchronization of asynchronous processes are performed in migration points. Migration points are defined by particular asynchronous events which may take place in each evolutionary process. If the event is occurs in the one of processes, the another random selected process is held. After that the migration operator is applied to both populations. The migration operator is transferred and copied individuals from one population to another.

The migration operator is applied to transfer chromosomes between populations. Individuals are selected from a number of chromosomes in populations with the best value of objective function. The selection is based on estimation of unrouting connection in the chromosome. For each placement which described by the chromosome the routing is implemented by the wave algorithm. Then chromosomes with the maximum total number of unrouting connections are copied from the population with the minimum total number of unrouting connections to other population. And the same numbers of chromosomes with the minimum value of the objective function are deleted from the second population.

Let the migration operator is applied to populations $P_i = \{h_1, h_2 \ldots h_s\}$ and $P_j = \{h_1, h_2 \ldots h_s\}$, where h_s is chromosome in population, $s = [1, N]$ (N is a number of chromosome in population). A number of unrouting connection in P_j population is less then in P_i population. As a result of selection we obtain a subpopulation $P_j' = \{h_1, h_2 \ldots h_t\}$ where h_i is a selected chromosome, $t = [1, M]$ (M is a number of selected from P_j population chromosome). Then in P_i population the t chromosomes of with the minimum value of objective function are replaced with chromosomes from P_j'. So the P_j population is not changed.

The right choice of migration frequency (time interval between migrations) and quantity of individuals is very important in the migration process. The frequent migration (migration of a large number of chromosomes) comes down to the combination of all populations and the preliminary convergence of the genetic algorithm. Also the rare migration can not prevent from the preliminary convergence of population. To define the probability of migration operator it is suggested fuzzy logic controller which is described below. The probability of migration operator along with crossover and mutation operators is determined on the basis of data on evolution effectiveness in each process.

In each evolutionary process the initial population is defined by the shotgun method. The selection is implemented by the roulette method. In the evolutionary process we apply the single-point crossover operator and multiple-point mutation operator in which the number of genes is proportional to the chromosome length. The probability of genetic operators is determined by fuzzy logic controller [4, 6, 13].

In Fig. 1 we show a block diagram of the developed algorithm for two population. In practice a number of population is considerably larger.

During the solution coding the set of position is represented by the regular structure (lattice). Each position $\mathrm{p_i}$ has coordinates $\mathrm{x_i}$, $\mathrm{y_i}$. Positions are enumerated in coordinate $\mathrm{x_i}$ ascending order within the string from left to right. But strings are enumerated in coordinate $\mathrm{y_i}$ ascending order from top to bottom. According with this rule position are enumerated as shown in Fig. 2.

Each solution is represented by chromosome Hi. The gene numerical number in chromosome corresponds the numerical number of the element which should be placed. The value of gene corresponds the number of position at connection field. A number of genes in the chromosome is equal a number of elements which should be placed.

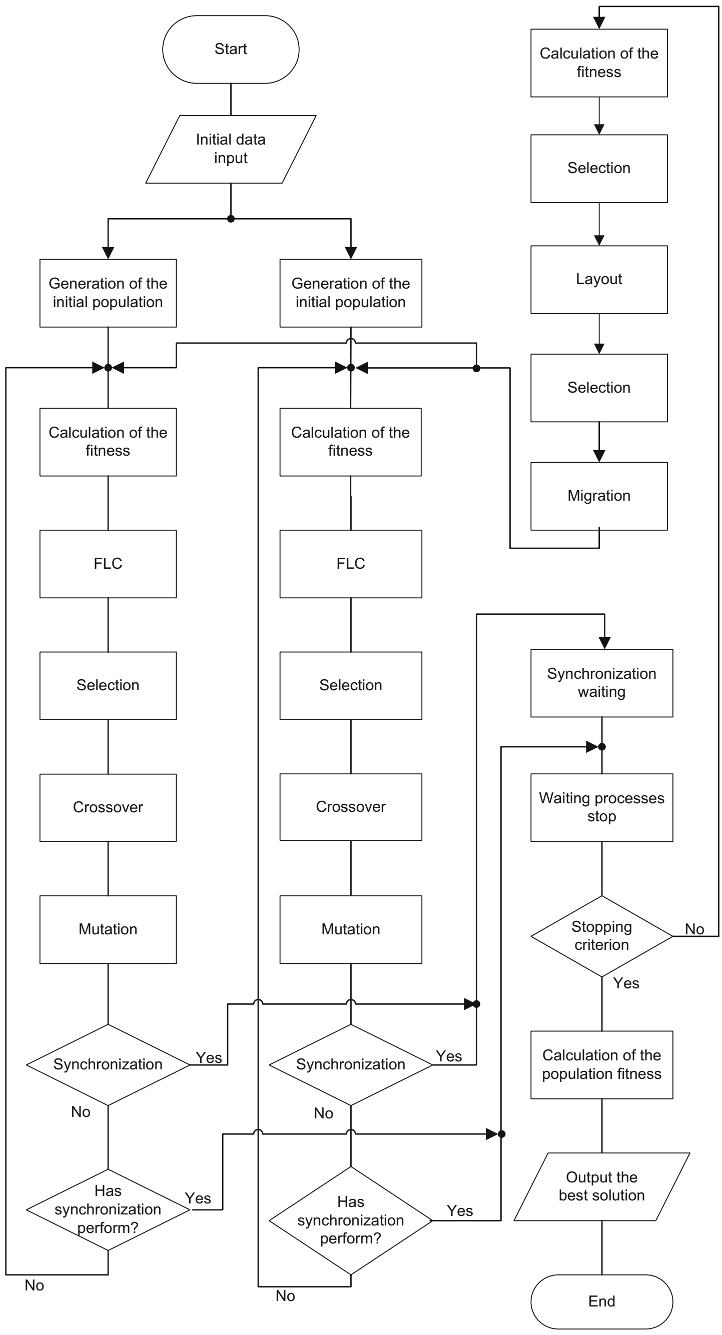

Fig. 1 The block diagram of the algorithm

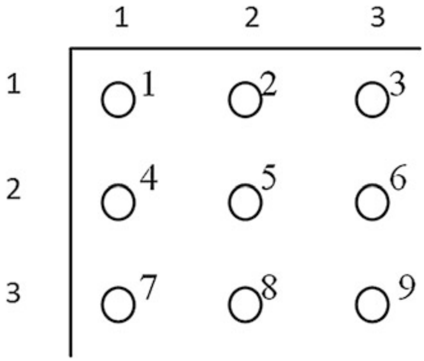

Fig. 2 The position enumeration at connection field

Each element has a basic point O_i^δ and basic coordinate axis $O_i^\delta X_i^\delta$, $O_i^\delta Y_i^\delta$. Each element has a square shape. Let element e_i is assigned in position p_j, if its basic point O_I^δ matches with the point in connection field which has coordinates x_j y_j [14].

For example the chromosome *<4> <5> <6> <9> <2> <3>* assign the placement shown in Fig. 3. Intersection points of black dashed lines correspond to position within connection field.

To calculate the objective function of placement it is needed to determined a normalized estimate of penalty for overlapping an area of placing elements, a total length of connection and a routing factor.

$$F_p = \alpha^* S + \beta^* L + \gamma^* T\ . \tag{1}$$

The penalty for overlapping an area of placing elements is a total intersection area of all elements.

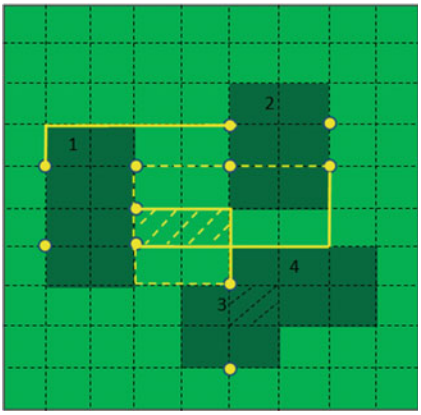

Fig. 3 Calculation of the placement fitness function

$$\sum_{i=0}^{n}\sum_{j=i+1}^{n} R_{ij} \tag{2}$$

where R_{ij} is an intersection area of *i*th and *j*th elements, *n* is a number of elements.

The interconnection length is estimated by calculation of half-perimeter of the net bounding rectangle.

$$L = \sum_{i=0}^{l} (x_i^{max} - x_i^{min}) + (y_i^{max} - y_i^{min}) \tag{3}$$

where x_i^{max}, x_i^{min}, y_i^{max}, y_i^{min} are maximum and minimum values of *x* and *y* in *i*th circuit, *l* is a number of nets.

The routing factor is determined by the total area of intersection of nets bounding rectangle.

$$T = \sum_{i=0}^{l}\sum_{j=i+1}^{l} Q_{ij} \tag{4}$$

where Q_{ij} is a an intersection area of bounding rectangles of *i*th and *j*th *nets*, *l* is a number of nets.

The objective function of routing is a percent of unrouting connections.

$$\mathrm{Fr} = {*}100, \tag{5}$$

where *Cr* is a number of unrouting connections, *C* is a total amount of connections.

4 Block of Fuzzy Control

The fuzzy control module is represented as follows:

$$\bar{y} = \frac{\sum_{k=1}^{N} \bar{y}^k \left(\prod_{i=1}^{n} \exp\left(- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right) \right)}{\sum_{k=1}^{N} \left(\prod_{i=1}^{n} \exp\left(- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right) \right)} \tag{6}$$

where x_i^{-k} is a centre, σ_i^k is a width of Gaussian curve (membership function of fuzzification block), y^k are centers of membership functions of defuzzification block fuzzy sets.

This expression is one of the most popular and frequently used approaches for fuzzy systems realization. Each element is defined by function block (sum, composition, Gaussian function) which allowed create a multilayer network. In this case a neural network contains four layer. At first layer input signals x_i __ arrive, and in output for this signals membership function values are formed. The second layer correspond a rule base and multipliers correspond an output block. The third and fourth blocks realize the defuzzification block [4–7, 13, 14].

To increase the quality of search results expert information includes with evolution process using fuzzy controller which regulate values of factors.

There are input parameters.

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)} \tag{7}$$

$$e_2(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{worst}(t) - f_{best}(t)} \tag{8}$$

$$e_3(t) = \frac{f_{best}(t) - f_{best}(t-1)}{f_{best}(t)} \tag{9}$$

$$e_4(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{ave}(t)} \tag{10}$$

where t is a time interval, $f_{best}(t)$ is the best value of the objective function at the iteration t; $f_{best}(t-1)$ is the best value of the objective function at the iteration $(t-1)$, $f_{worst}(t)$ is the worse value of the objective function at the iteration t, $f_{ave}(t)$ is an average value of the objective function at the t iteration, $f_{ave}(t-1)$ is an average value of the objective function at $(t-1)$ iteration [11] (Fig. 4).

As a result we obtain probabilities of crossover, mutation and migration operators. Consequently, to determine these parameters we use the fuzzy control block three times.

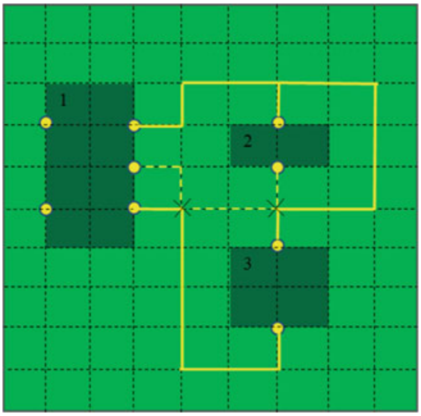

Fig. 4 Calculation of routing fitness function

5 Experimental Results

To estimate the algorithm effectiveness we placed and route 300 randomly generated elements and 150 nets with from 2 to 5 pins. Experiments result with different number of parallel algorithm streams is shown in Table 1.

To compare the effectiveness the test problems solved using the FLC and without it are investigated earlier [14]. Table 2 showed that the efficiency of the algorithm with use the controller is much higher than the efficiency of the algorithm without it.

Table 1 Experimental results

Number of streams	Experiment number, % unrouting connections					Average value of % unrouting connection
	1	2	3	4	5	
1	23	21	24	24	22	22.8
2	16	13	14	14	15	14.4
3	13	14	12	13	11	12.6
4	12	10	13	10	11	11.2
5	14	13	13	14	12	13.2

Table 2 Comparison

Number	Without FLC (N_{el} = 50)	With FLC (N_{el} = 50)	Without FLC (N_{el} = 100)	With FLC (N_{el} = 100)	Without FLC (N_{el} = 150)	With FLC (N_{el} = 150)
1	4585	3147	29,658	21,296	67,953	48,509
2	3870	3330	31,145	23,582	64,311	51,737
3	4245	2724	28,192	23,145	68,989	50,901
4	4056	3425	31,632	23,481	65,576	50,798
5	3774	2885	29,761	21,844	65,184	48,973
6	4896	2984	28,487	23,148	67,925	49,752
7	4129	2873	31,845	22,946	65,427	52,164
8	4812	3776	29,145	21,941	64,964	48,862
9	3981	3145	29,411	22,157	65,817	50,314
10	3876	3168	30,491	22,981	68,482	50,957
Average result	42,224	31,457	299,767	226,521	668,628	502,967
Increase quality of solution (%)	25.6		24.44		24.78	

6 Conclusion

Results of the experiments showed that the efficiency of the controller is increased after the introduction of the training unit on the basis of an artificial neural network model.

FLC parameters that were used in the study were obtained using a genetic algorithm learning. Training was carried out on the basis of statistical information on the dependence of the FLC parameters and the efficiency of the algorithm placement. This information is collected during the learning process.

We plan to further evaluate the effectiveness of the algorithm by simultaneously solving accommodation problems and tracing. Also, a comparative analysis of the obtained results with known analogues will be held.

Acknowledgments This research is supported by the Ministry of Education and Science of the Russian Federation, the project # 8.823.2014.

References

1. Shervani, N.: Algorithms for VLSI Physical Design Automation, 538 pp. Kluwer Academy Publisher, USA (1995)
2. Cohoon, J.P., Karro, J., Lienig, J.: Evolutionary algorithms for the physical design of VLSI circuits. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computing: Theory and Applications, pp. 683–712. Springer, London (2003)
3. Gladkov, L.A., Kureichik, V.V., Kureichik, V.M.: Genetic Algorithms. Fizmatlit, Moscow (2010)
4. Michael, A., Takagi, H.: Dynamic control of genetic algorithms using fuzzy logic techniques. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 76–83. Morgan Kaufmann (1993)
5. Lee, M.A., Takagi, H.: Integrating design stages of fuzzy systems using genetic algorithms. In: Proceedings of the 2nd IEEE International Conference on Fuzzy System, pp. 612–617 (1993)
6. Herrera, F., Lozano, M.: Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. J. Soft Comput. 545–562 (2003)
7. Liu, H., Xu, Z., Abraham, A.: Hybrid fuzzy-genetic algorithm approach for crew grouping. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), pp. 332–337 (2005)
8. King, R.T.F.A., Radha, B., Rughooputh, H.C.S.: A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In: Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, pp. 577–582 (2004)
9. Im, S.-M., Lee, J.-J.: Adaptive crossover, mutation and selection using fuzzy system for genetic algorithms. Artif. Life Robot. **13**(1), 129–133 (2008)
10. Rodriguez, M.A., Escalante, D.M., Peregrin, A.: Efficient distributed genetic algorithm for rule extraction. Appl. Soft Comput. **11**, 733–743 (2011)
11. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE T. Evol. Comput. **6**, 443–461 (2002)
12. Zhongyang, X., Zhang, Y., Zhang, L., Niu, S.: A parallel classification algorithm based on hybrid genetic algorithm. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 3237–3240 (2006)

13. Gladkov, L., Gladkova, N., Leiba, S.: Manufactoring scheduling problem based on fuzzy genetic algorithm. In: Proceeding of IEEE East-West Design and Test Symposium—(EWDTS'2014). Kiev, Ukraine, pp. 209–212 (2014)
14. Gladkov, L.A., Gladkova, N.V., Leiba, S.N.: Electronic computing equipment schemes elements placement based on hybrid intelligence approach. Advanced in Intelligent Systems and Computing. In:: Intelligent Systems in Cybernetics and Automation Theory, vol. 348, pp. 35–45. Springer International Publishing, Switzerland (2015)