

mrezapoorm@yahoo.com

()

NP-Complete

:

()

[]

[]

[] PC-Board

VLSI

()

[] []

G2 G1

G2 G1

G2

G2 S

G2 G1

G1

S G1

1

2

3



	[]						
	[]	$O(n!)$	n			Backtracking	
	[]				Hoffman	Luks	[]
	[]		$O(n^4)$				[]
NP-	[]						[]
[]	[]		[]	[]	[]	[]	Complete

$E \subset V \times V$ V (V, E, α)
 $f: V_1 \rightarrow V_2$ R_v R_v V $\alpha: V \rightarrow R_v$
 $M(G)$ $n \times n$ $G_2=(V_2, E_2)$ $G_1=(V_1, E_1)$
 $(f(u), f(w))$ G_1 (u, w)
 n G
 $M(G)$ H G
 σ $M(H)$ $M(G)$ $(M(H))$

$M(H) = P \cdot M(G) \cdot P^T$
 $J(\sigma) = \| M(H) - P \cdot M(G) \cdot P^T \|^2$
 $\theta(n^3)$ $J(\sigma)$ n $J(\sigma)$

$[P \cdot M(G) \cdot P^T]_{i,j} = [M(G)]_{\sigma(i), \sigma(j)}$
 $J(\sigma) = \| M(H) - P \cdot M(G) \cdot P^T \|^2$
 H G A j i $[A]_{i,j}$
 $(\| M \|^2 = \sum_i \sum_j |m_{ij}|^2)$ L1

$$J_k(\sigma) = \sum_{m=1}^n [M(H)]_{k,m} - [M(G)]_{\sigma(k), \sigma(m)} \quad | \quad + \sum_{m=1}^n [M(H)]_{m,k} - [M(G)]_{\sigma(m), \sigma(k)} |$$



$$J_k(\sigma) = 2 \times \sum_{m=1}^n [M(H)]_{k,m} - [M(G)]_{\sigma(k), \sigma(m)} \quad | \quad k$$

$$J(\sigma) = \sum_{k=1}^n J_k(\sigma) \quad \sigma$$

$$\theta(n^2) \quad J(\sigma)$$

[]

[] :

[] [] [] [] [] []

[] [] [] [] [] []

[]

n! n

n!

[]

[]

[] [] []

()

) K

m H u (

N $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_{KN}\}$ K

$\underline{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$

G M_m H u

- 4 Object Migrating Automata (OMA)
- 5 Oommen
- 6 Ma



$\{\Phi_{(j-1)N+1}, \dots, \Phi_{jN}\}$

H u

j

G M_j H u

$\Phi_{(j-1)N+1}$

Φ_{jN}

$\Phi_{(j-1)N+1}$

Φ_{jN}

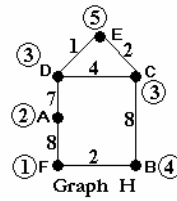
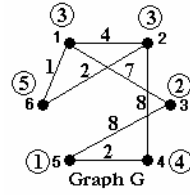
H G

{A,B,C,D,E,F}

H

{1,2,3,4,5,6}

G



{(1, D), (2, F), (3, A), (4, B), (5, E), (6, C)}

() Tsetline

{ }

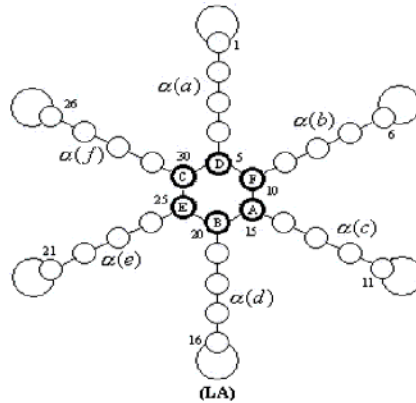
(G) $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$

{ }

H

() ()

()



{(1, D), (2, F), (3, A), (4, B), (5, E), (6, C)}

:

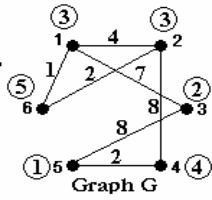
n-1

n

n-1

H G

$\{(1, E), (2, F), (3, B), (4, D), (5, A), (6, C)\}$ $\{(1, D), (2, E), (3, F), (4, B), (5, C), (6, A)\}$ $\{(1, B), (2, D), (3, E), (4, A), (5, F), (6, C)\}$
 $\{(1, B), (2, D), (3, C), (4, A), (5, E), (6, F)\}$ $\{(1, C), (2, F), (3, B), (4, E), (5, D), (6, A)\}$
 $\{(6, E), (5, F), (4, B), (1, D), (2, A), (3, C)\}$ H G
 H G



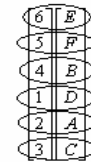
ماتریس مجاورت گراف G

$$\begin{bmatrix}
 1 & 3 & 4 & 7 & 0 & 0 & 1 \\
 2 & 4 & 3 & 0 & 8 & 0 & 2 \\
 3 & 7 & 0 & 2 & 0 & 8 & 0 \\
 4 & 0 & 8 & 0 & 4 & 2 & 0 \\
 5 & 0 & 0 & 8 & 2 & 1 & 0 \\
 6 & 1 & 2 & 0 & 0 & 0 & 5
 \end{bmatrix}$$

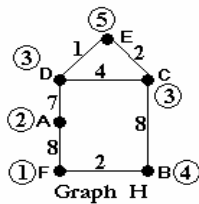
ماتریس درجه رؤوس گراف G

$$\begin{bmatrix}
 1 & 15 \\
 2 & 17 \\
 3 & 17 \\
 4 & 14 \\
 5 & 11 \\
 6 & 8
 \end{bmatrix}$$

ماتریس مرتب شده رؤوس گراف G برحسب درجه هر رأس

$$\begin{bmatrix}
 6 \\
 5 \\
 4 \\
 1 \\
 2 \\
 3
 \end{bmatrix}$$


نگاشت حاصل از رؤوس مرتب شده دو گراف H و G



ماتریس مجاورت گراف H

$$\begin{bmatrix}
 A & 2 & 0 & 0 & 7 & 0 & 8 \\
 B & 0 & 4 & 8 & 0 & 0 & 2 \\
 C & 0 & 8 & 3 & 4 & 2 & 0 \\
 D & 7 & 0 & 4 & 3 & 1 & 0 \\
 E & 0 & 0 & 2 & 1 & 5 & 0 \\
 F & 8 & 2 & 0 & 0 & 0 & 1
 \end{bmatrix}$$

ماتریس درجه رؤوس گراف H

$$\begin{bmatrix}
 A & 17 \\
 B & 14 \\
 C & 17 \\
 D & 15 \\
 E & 8 \\
 F & 11
 \end{bmatrix}$$

ماتریس مرتب شده رؤوس گراف H برحسب درجه هر رأس

$$\begin{bmatrix}
 E \\
 F \\
 B \\
 D \\
 A \\
 C
 \end{bmatrix}$$

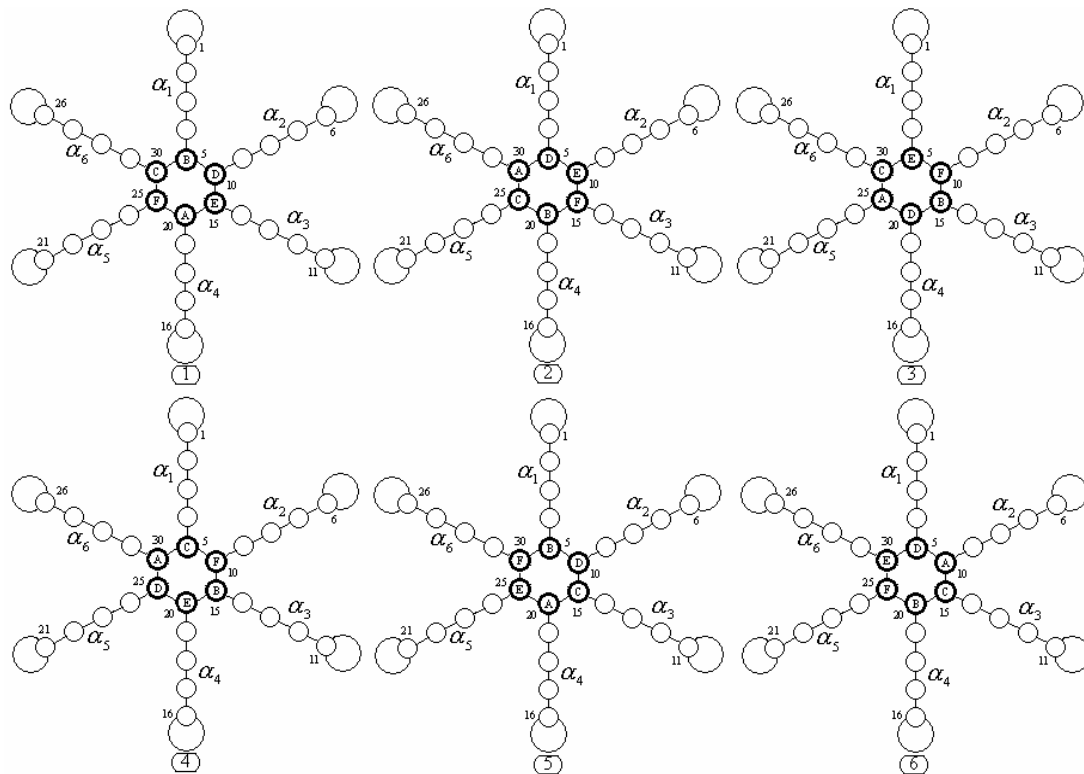
مرحله اول

مرحله دوم

مرحله سوم

مرحله چهارم

مرحله پنجم



H G

$$J(\sigma) \quad (\quad) \quad \sigma \quad :$$

$$P \quad H \quad G \quad M(H) \quad M(G)$$

$f(\sigma)$

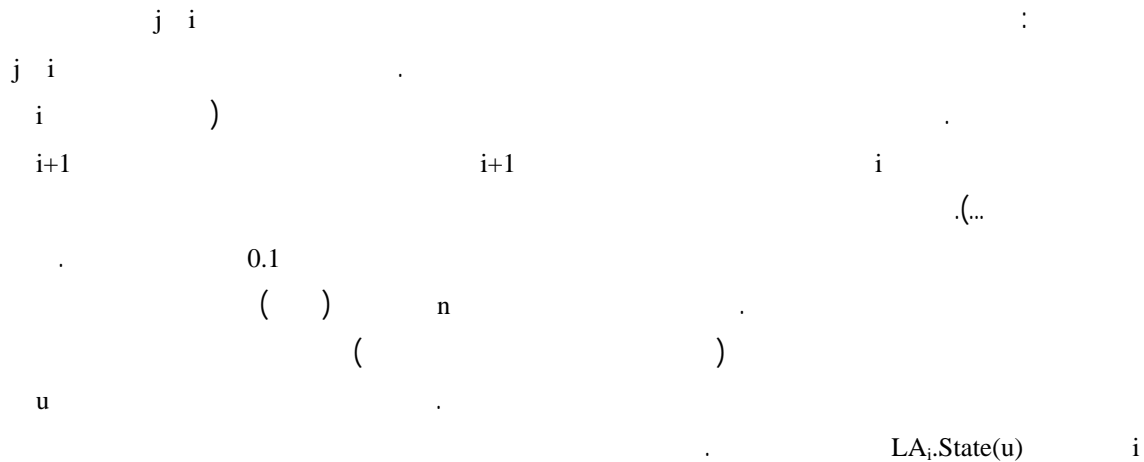
$$f(\sigma) = C \max_{P.M(G), P^T} - J(\sigma) = C \max_{P.M(G), P^T} - \| M(H) - P.M(G).P^T \|$$

 $O(n^2)$

$$[P.M(G).P^T]_{i,j} = [M(G)]_{\sigma(i),\sigma(j)}$$

$$\| M \| = \sum \sum | m_{ij} | \quad L1 \quad \| \|$$

$$f(\sigma) = C \max \left| \sum_{i=1}^n \sum_{j=1}^n m_{ij} - m_{\sigma(i),\sigma(j)} \right|$$


Procedure Crossover (LA_1, LA_2)

Generate two random numbers r_1 and r_2 between 1 to n

$r_1 = \text{Random} * n; r_2 = \text{Random} * n;$

$r_1 = \text{Min}(r_1, r_2), r_2 = \text{Max}(r_1, r_2)$

For $i = r_1$ **to** r_2 **do**

If ($J_i(LA_1) < J_i(LA_2)$) **then**

$j = \text{Action of } LA_2 \text{ where}$

$LA_2.Object(LA_2.Action(j)) = LA_1.Object(LA_1.Action(i));$

$\text{Swap}(LA_2.Object(LA_2.Action(i)), LA_2.Object(LA_2.Action(j)));$

Else

$j = \text{Action of } LA_1 \text{ where}$

$LA_1.Object(LA_1.Action(j)) = LA_2.Object(LA_2.Action(i));$

$\text{Swap}(LA_1.Object(LA_1.Action(i)), LA_1.Object(LA_1.Action(j)));$

Endif

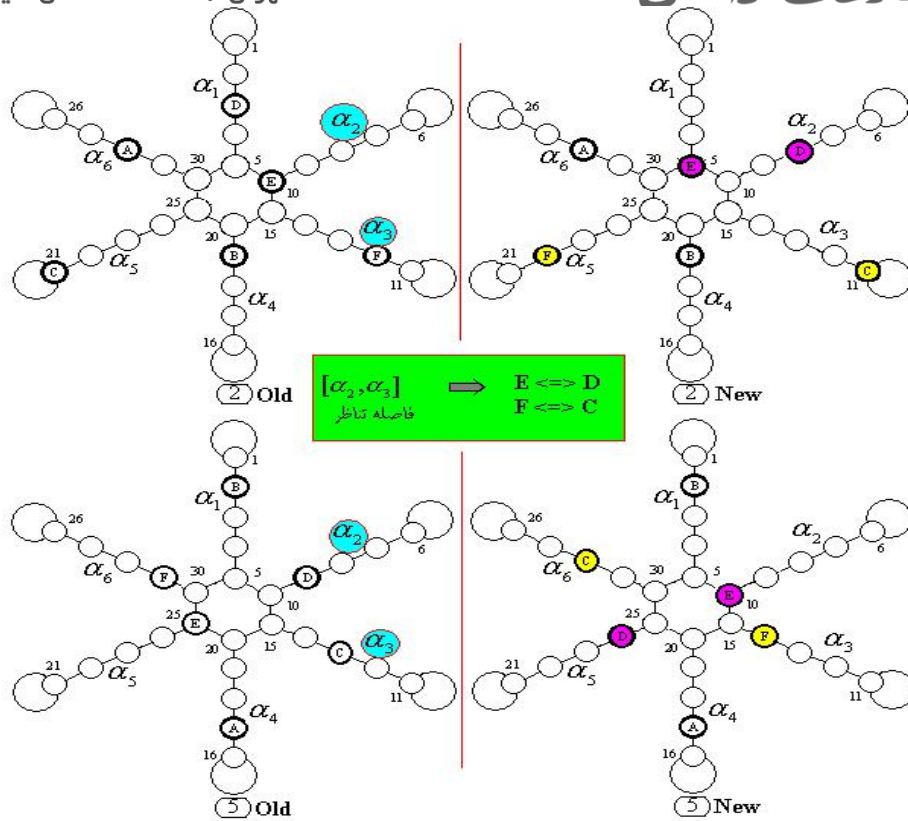
End For

End Crossover

$LA_2 \quad LA_5$

$\{\alpha_2, \alpha_3\}$

α_3, α_2



0.4

Procedure Mutation (LA)

```
i = Random *n; j = Random *n;
Swap( LA.Object( LA.Action(i)),LA.Object(LA.Action(j))) ;
End Mutation
```

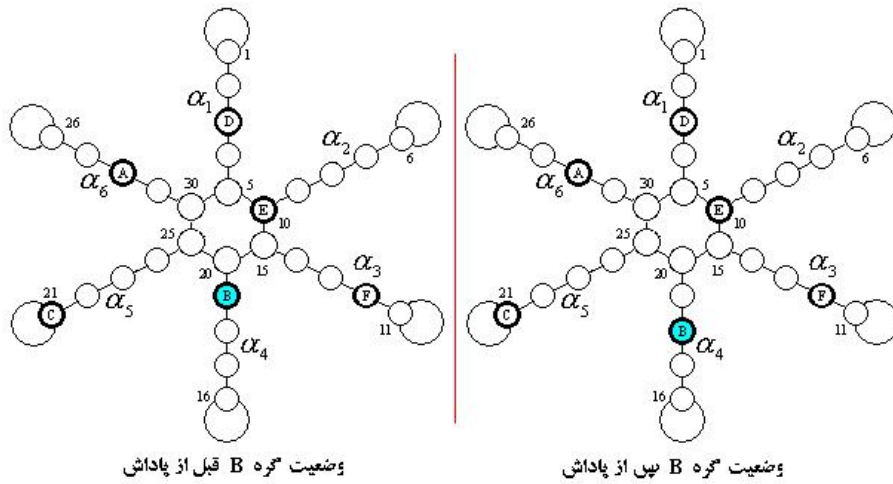
Procedure Reward (LA, u)

```
If (LA.State(u)-1) mod N <> 0 then
    Dec (LA.State(u));
End If
End Reward
```

{ } B Tsetline) G 4 H B (

B

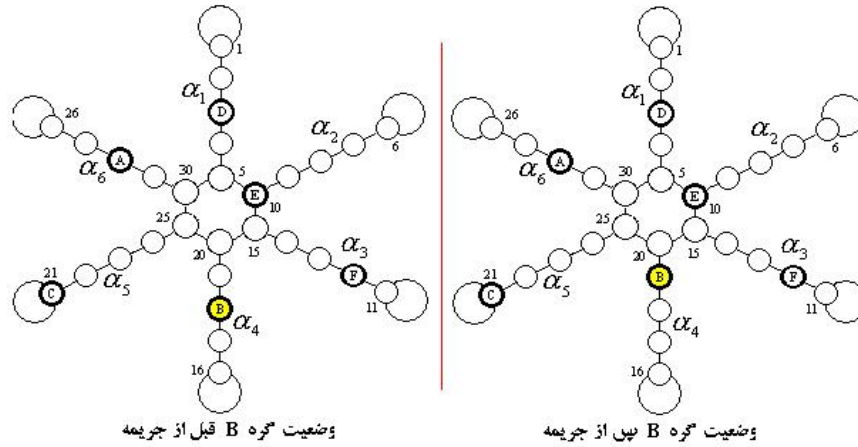
()



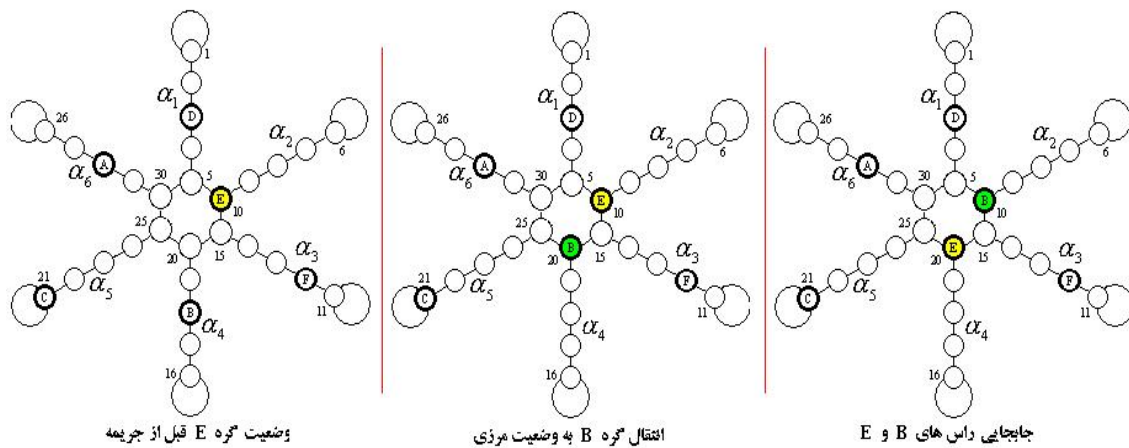
B

```

Procedure Penalize( LA, u )
  repeat
    For U = 1 to n do
      If (LA.State(U)) mod N <> 0 then
        Inc(LA.State(U));
      End If
    End for
    Until at least one node appears in the boundary state
    bestError = ∞ ;
    for U = 1 to n do
      Create mapping LA' from LA by swapping u and U
      If JU(LA') < bestError then
        bestError = JU(LA');
        bestNode = U;
      End If
    End for
    LA.State(bestNode) = LA.Action(bestNode)*N;
    LA.State(u) = LA.Action(u)*N;
    Swap(LA.State(u),LA.State(bestNode));
  End Penalize
  
```

H

**Function** GraphIsomorphism(G,H)n=|V_G|;Create the initial population LA₁ ... LA_n;

EvalFitness();

While(All $J(LA_1), \dots, J(LA_n) \neq 0$) **do**NewLA₁ = NewLA₂ = LA with minimum Value of J;**For** i = 2 to n **do**Select LA₁ ; Select LA₂ ;**If** (Random > 0.9) **then**Crossover (LA₁, LA₂);**End If****If** (Random > 0.6) **then**Mutation (LA₁); Mutation (LA₂);**End If**NewLA_{i+1} = LA₁;NewLA_{i+2} = LA₂ ;

i=i+2;

End For**For** i = 0 to n **do**LA_i = NewLA_i;

u = Random *n;

```

If ( $J_u(LA_i) < T$ ) then
  Reward( $LA_i, u$ );
Else
  Penalize( $LA_i, u$ );
End If
End For
EvalFitness();
End While
End GraphIsomorphism

```

[0,100]

(GA+LA)

(GA)

(LA)

(GA+LA)

(GA)

(LA)

(GA+LA)

(GA)

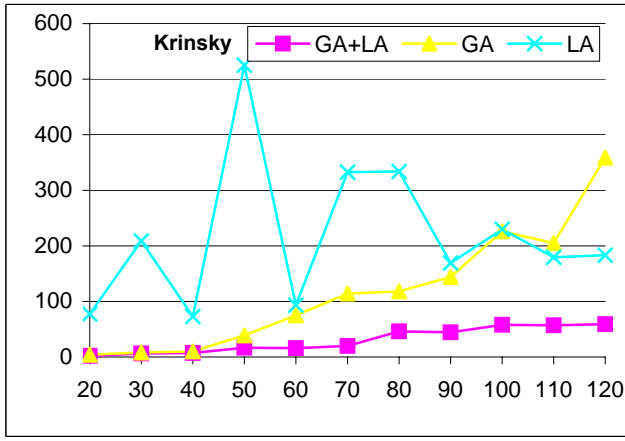
(LA)

(GA+LA)

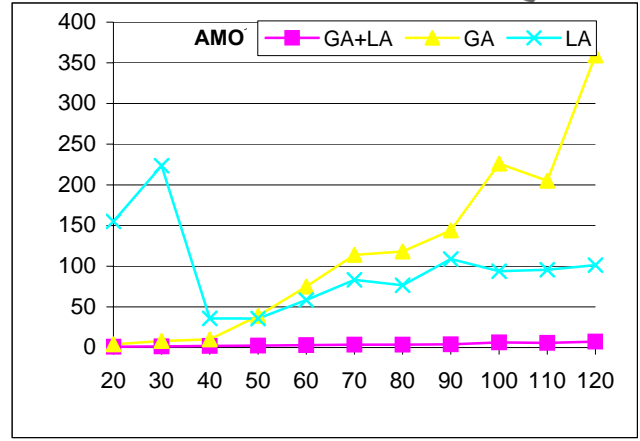
(GA)

(LA)

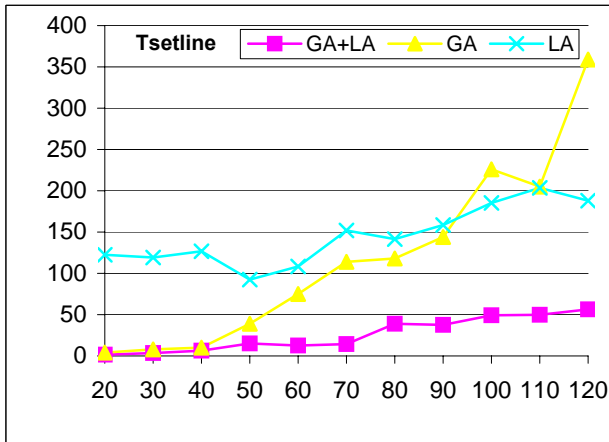
- [1] H. Beigy and M. R. Meybodi, "Randomized Las Vegas Algorithm for Graph Isomorphism", Proceedings of Third International Conference on Intelligent Data Engineering and Automated Learning, Manchester, UK, Aug. 12-14 2002.
- [2] Y. Wang and K. "Fan Genetic-Basic Search for Error-Correcting Graph Isomorphism", IEEE Transaction on Systems, Man. And Cybernetics-Par` t B: Cybernetics, Vol. 27, No. 4, August 1997.
- [3] B. T. Messmer, "Efficient Graph Matching Algorithms for Preprocessed Model Graphs", Ph.D. Thesis, Inst. of Comp. Sci. and Applied Mathematics, University of Bern, 1996.
- [4] H. Beigy and M. R. Meybodi, "Optimization of Topology of Neural Networks Using Learning Automata", Proc. Of 3th Annual Int. Computer Society of Iran Computer Conf. CSICC-98, Tehran, Iran, pp. 417-428, 1999.
- [5] C. M. Hoffman. Group-theoretic Algorithms and Graph Isomorphism. Springer Verlag, 1982.
- [6] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, MA: Addison-Wesley, 1989.
- [7] P. Mars, K. S. Narendra and M. Chrystall, "Learning Automata Control of Computer Communication Networks", Proc. Of Third Yale workshop on Application of Adaptive Systems Theory. Yale University, 1983.
- [8] A. Hashim, S. Amir and P. Mars, "Application of Learning Automata to Data Compression", In Adaptive and Learning Systems, K. S. Narendra (Ed), New York: Plenum Press, pp. 229-234, 1986.
- [9] M. A. L. Thathachar and P. S. Sastry, "Learning Optimal Discriminant Functions Through a Cooperative Game of Automata", IEEE Trans. Syst., Man and Cybern., Vol. 27, No. 4, pp. 588-597, 1997.
- [10] K. S. Narendra and M. A. L. Thathachar, Learning Automata: An Introduction, Prentice-hall, Englewood cliffs, 1989.
- [11] M. R. Meybodi and S. Lakshmi varhan, "A Learning Approach to Priority Assignment in a Two Class M/M/1 Queuing System with Unknown Parameters", Pproc. Of Third Yale Workshop on Applications of Adaptive System Theory, Yale University, pp. 106-109, 1983.
- [12] M. R. Meybodi and H. Beigy, "New Class of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters", Proc. Of EUFIT-98, Sep. 7-10, Achen, Germany, pp. 339-344, 1998.
- [13] B. J. Oommen and D. C. Y. Ma, "Deterministic Learning Automata Solution to the Keyboard Optimization Problem", IEEE Trans. On Computers, Vol. 37, No. 1, pp. 2-3, 1988.
- [14] H. Beigy and M. R. Meybodi, "Optimization of Topology of neural Networks Using Learning Automata", Proc. Of 3th Annual Int. Computer Society of Iran Computer Conf. CSICC-98, Tehran, Iran, pp. 417-428, 1999.
- [15] J. E. Hopcroft and J. K. Wong, "Linear time algorithm for isomorphism of planar graphs", In Annual ACM Symposium on Theory of Computing, pages 172--184, 1974.
- [16] J. R. Ullman, "An Algorithm for Subgraph Isomorphism", Journal of the Association for Computing Machinery, Vol. 23, No. 1, pp. 31--42, 1976.
- [17] B. Falkenhainer, K.D. Forbus, and D. Gentner. "The Structure-mapping Engine: Algorithms and Examples", Artificial Intelligence, No 41, pp. 1--63, 1989/90.
- [18] K. Bryant, Genetic Algorithms and the Traveling Salesman Problem , Thesis, Harvey Mudd College, Dept. of Mathematics, 2000.
- [19] E. Cantu-Paz, "A Survey of Parallel Gentic Algorithms", IlliGAL Report, No. 97003, May 1997.
- [20] B. J. Oommen, R. S. Valiveti J. R. and Zgierski, "An Adaptive Learning Solution to the Keyboard Optimization Problem", IEEE Trans. On Systems. Man. And Cybernetics, Vol. 21, No. 6, pp. 1608-1618, 1991.
- [21] B. J. Oommen and E. V. Croix, "Graph Partitioning Using Learning Automata", IEEE Trans. On Computers, Vol. 45, No. 2, pp. 195-208, 1996.



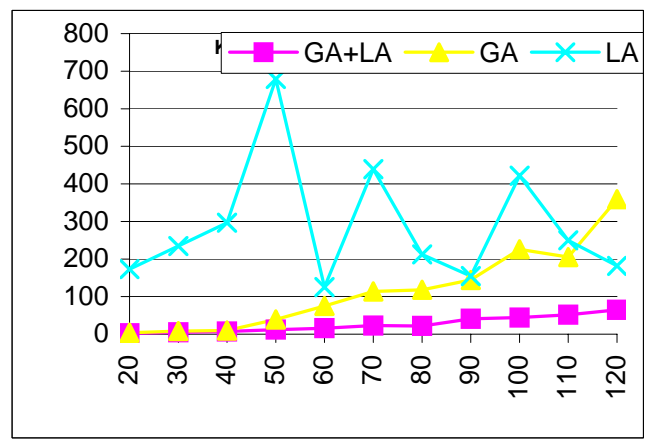
LA GA+LA :
GA



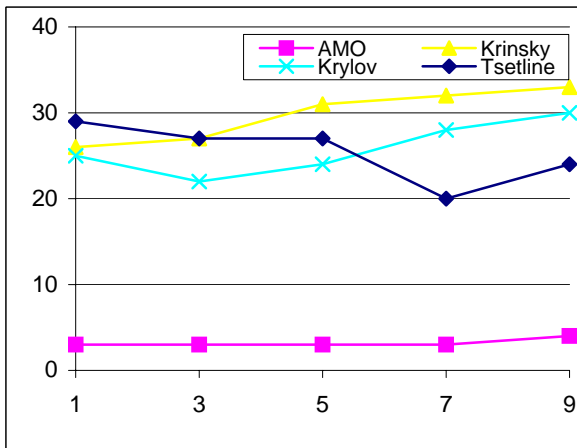
LA GA+LA :
GA



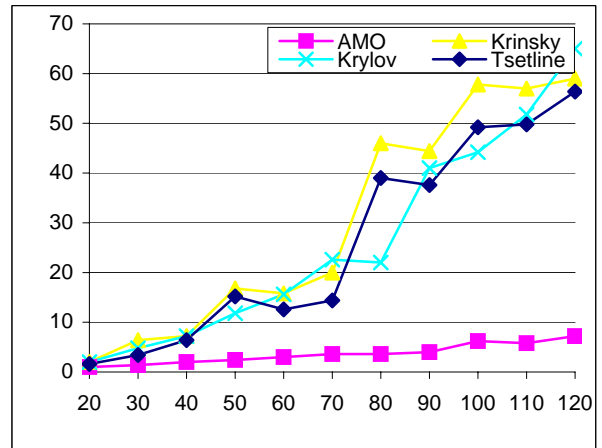
LA GA+LA :
GA



LA GA+LA :
GA



GA+LA



GA+LA :
: