

Analyzing Directed Acyclic Graph Recombination

Carlos Cotta and José M. Troya

Dept. Lenguajes y Ciencias de la Computación,
ETSI Informática (Office 3.2.49), University of Málaga,
Campus de Teatinos, 29071 - Málaga - SPAIN

`ccottap@lcc.uma.es`

Abstract. This work studies the edge-based representation of directed acyclic graphs, as well as the properties of recombination operators working on it. It is shown that this representation is not separable, and the structure of the basic information units that must be processed in order to maintain feasibility of the solutions is described. As an experimental analysis indicates, a recombination operator using these units has sub-quadratic complexity in the graph size. It is also shown that a standard gene-transmission recombination operator is biased to produce solutions of lower edge-density than the parents' average. An unbiased allelic recombination operator provides better results on an ad-hoc test problem.

1 Introduction

Given a digraph $G(V, E)$, where $V = \{v_1, \dots, v_n\}$ is a set of vertices and $E \subseteq V \times V$, a vertex v_j is said to be a descendant of v_i if, and only if, $(v_i, v_j) \in E$, or $(v_k, v_j) \in E$ and v_k is a descendant of v_i . A Directed Acyclic Graph (DAG) is a digraph $DAG(V, E)$, where E verifies that no $v_i, v_j \in V$ exist such that v_i is a descendant of v_j and vice versa.

DAGs are important data structures since they can be used to represent many relevant entities such as *field programmable gate arrays* (FPGAs) [1], Bayesian networks [11, 7], task graphs [2], scene graphs [15], dependence graphs [8, 6] etc. The synthesis of such entities is generally complex, being often approached via evolutionary algorithms (e.g., [5, 9, 12]). For this reason so it is worth analyzing application issues of evolutionary heuristics to this task.

In this work, we focus on DAG recombination from the perspective of an edge-based representation of DAGs. Both this representation and the innards of information transmission during recombination are theoretically analyzed within the framework of Forma Analysis [13]. For this purpose, some background information on Forma Analysis is given (Section 2) prior to the mentioned analysis (Section 3). Subsequently, several variants of recombination are empirically studied in terms of time complexity, edge-transmission ratios and algorithmic performance (Section 4). Finally, some conclusions and future work are outlined (Section 5).

2 Background

Let $\Phi = \{\phi_1, \dots, \phi_n\}$ be a set of n equivalence relations defined over a discrete search space \mathcal{S} . Let Ξ_ϕ be the set of equivalence classes induced by ϕ , and let Ξ_Ψ ($\Psi \subseteq \Phi$) be the set of vectors of equivalence classes induced by equivalence relations in Ψ . Let $[x]_\phi$ be the equivalence class to which x belong under ϕ . If it holds for Φ that for any two different solutions $x, y \in \mathcal{S}$, there exists $\phi \in \Phi$ such that $[x]_\phi \neq [y]_\phi$, then Φ covers the search space \mathcal{S} . If Φ is independent and covers the search space \mathcal{S} , then each solution $x \in \mathcal{S}$ can be represented as a string $\langle [x]_\phi \mid \phi \in \Phi \rangle$, i.e., $x = \langle \eta_1, \eta_2, \dots, \eta_n \rangle \Leftrightarrow \{x\} = \bigcap_{i=1}^n \eta_i$. Being any ambiguity clear from the context, the same notation is used for equivalence classes and their labels. Each $\phi \in \Phi$ is analogous to a gene, and each $\eta \in \Xi_\phi$ is analogous to an allele for this gene.

The *Dynastic Potential* $\Gamma(\{x, y\})$ of x and y is defined as

$$\Gamma(\{x, y\}) = \bigcap_{\phi \in \Phi} ([x]_\phi \cup [y]_\phi), \quad (1)$$

i.e., the set of solutions that only carry information present in x or y . On the other hand, the *Similarity Set* $\Sigma(\{x, y\})$ is defined as

$$\Sigma(\{x, y\}) = \bigcap_{\phi \in \Phi, [x]_\phi = [y]_\phi} [x]_\phi, \quad (2)$$

i.e., the largest forma that contains x and y [14]. Notice that $\Gamma(\{x, y\}) \subseteq \Sigma(\{x, y\})$. If for any alleles η, ζ such that $\eta \cap \zeta \neq \emptyset$, and $x \in \eta$, $y \in \zeta$, it holds that $\eta \cap \zeta \cap \Sigma(\{x, y\}) \neq \emptyset$, then Φ is *separable*. The notation $\eta \triangleright \Psi$ denotes that Ψ is the intersection of several alleles, including η . Similarly, the notation $\Theta \triangleright \Psi$ denotes that Ψ is the intersection of several alleles, including all $\theta \triangleright \Theta$.

3 DAG Recombination

Let $V = \{v_1, \dots, v_n\}$ be the set of available vertices, and let \mathcal{S}_V be the search space composed of all DAGs defined over V ¹. Given $z \in \mathcal{S}_V$, let $E(z)$ be the set of edges in z . Now let us consider a family of equivalence relations ϕ_{ij} defined over \mathcal{S}_V as

$$\phi_{ij}(x, y) = \text{TRUE} \Leftrightarrow [(v_i, v_j) \in E(x) \Leftrightarrow (v_i, v_j) \in E(y)] . \quad (3)$$

Each equivalence relation ϕ_{ij} induces two formae ϕ_{ij}^1 and ϕ_{ij}^0 , respectively comprising the solutions that include/exclude edge (v_i, v_j) . Let $\Phi = \{\phi_{ij} \mid 1 \leq i, j \leq n, i \neq j\}$. This set of equivalence relations can be used to induce a representation of solutions in \mathcal{S}_V . To show this, it is necessary to prove that (a) Φ is independent, and (b) Φ covers \mathcal{S}_V . This is done below.

¹ This search space has an enormous size. There is no closed form for $|\mathcal{S}_V|$, but it grows super-exponentially with $|V|$. For instance, $|\mathcal{S}_V| = 543$ for $|V| = 4$, and $|\mathcal{S}_V| \approx 10^{18}$ for $|V| = 10$ (cf. [10])

Lemma 1. *Let $\Phi' \subset \Phi$, and let $\phi_{ij} \in \Phi - \Phi'$. There always exists an allele $\eta \in \Xi_{\phi_{ij}}$ such that for some $x, y \in \eta$, no allele set $\Omega \in \Xi_{\Phi'}$ can be found for which $\{x, y\} \subseteq \bigcap_{\xi \in \Omega} \xi$.*

Proof. The proof is done by induction on the cardinality of Φ' . Initially, let $\Phi' = \{\phi_{kl}\}$. Let $\eta \equiv \phi_{ij}^0$. Let $x, y \in \eta$ be such that $(k, l) \in E(x)$, and $(k, l) \notin E(y)$. These two solutions exists (for instance, consider $E(x) = \{(k, l)\}$, and $E(y) = \emptyset$). Now, Ω can take two values: $\{\phi_{kl}^0\}$, or $\{\phi_{kl}^1\}$. Clearly, x does not belong to ϕ_{kl}^0 in the first case, and the same holds for y with respect to ϕ_{kl}^1 in the second case. Hence the base case is established. We now assume that the lemma holds whenever $|\Phi'| = k$. Subsequently, we consider the case $|\Phi'| = k + 1$. Let $\Phi' = \Psi \cup \{\phi_{kl}\}$. Then, any $\Omega \in \Xi_{\Phi'}$ can be expressed as $\Theta \cup \{\zeta\}$, where $\Theta \in \Xi_{\Psi}$ and $\zeta \in \Xi_{\phi_{kl}}$. Assume that the lemma is false in this situation. Thus, for all $\eta \in \Xi_{\phi_{ij}}$ and $x, y \in \eta$, there exists $\Omega \in \Xi_{\Phi'}$ such that $\{x, y\} \subseteq \bigcap_{\xi \in \Omega} \xi$. However, $\{x, y\} \subseteq \bigcap_{\xi \in \Omega} \xi$ implies that $\{x, y\} \subseteq \bigcap_{\xi \in \Theta} \xi$ (recall $\Theta \subset \Omega$). But this is false according to the induction hypothesis. Since we arrive at a contradiction, the lemma must also hold in this case. \square

Proposition 1. *The set of equivalence relations Φ is independent and covers the search space \mathcal{S}_V .*

Proof. (Independence) It must be shown that given $\phi_{ij} \in \Phi$, no $\Phi' \subseteq (\Phi - \{\phi_{ij}\})$ exists such that $\phi_{ij} \equiv \bigcap_{\psi \in \Phi'} \psi$, i.e., ϕ_{ij} cannot be expressed as the intersection of other equivalence relations in Φ . The proof is done by absurdity. Let $\phi_{ij} \equiv \bigcap_{\psi \in \Phi'} \psi$. This implies that for any $\eta \in \Xi_{\phi_{ij}}$, an allele set $\Omega \in \Xi_{\Phi'}$ exists for which $\eta \equiv \bigcap_{\xi \in \Omega} \xi$. Clearly, this means that for all $x \in \eta$, it holds that $x \in \bigcap_{\xi \in \Omega} \xi$. But this contradicts Lemma 1, so the initial assumption ($\phi_{ij} \equiv \bigcap_{\psi \in \Phi'} \psi$) must be false.

(Coverage) It must be shown that for all $x, y \in \mathcal{S}_V$ ($x \neq y$), there exists $\phi_{ij} \in \Phi$ such that $\phi_{ij}(x, y) = \text{FALSE}$. This is easy to prove since –without loss of generality– $x \neq y$ implies that $v_i, v_j \in V$ exist, such that $(v_i, v_j) \in E(x)$ but $(v_i, v_j) \notin E(y)$. Hence, $x \in \phi_{ij}^1$ and $y \in \phi_{ij}^0$, i.e., $\phi_{ij}(x, y) = \text{FALSE}$. \square

Proposition 1 shows that any DAG $x \in \mathcal{S}_V$ can be univocally and compactly represented as a string $\langle \phi^k \mid \phi \in \Phi, k \in \{0, 1\} \rangle$. As shown below, this representation induced by Φ is not separable.

Proposition 2. *The set of equivalence relations Φ is not separable.*

Proof. The proof is done by example. Let $v_i, v_j, v_k \in V$ be three different vertices. Clearly, it holds that

$$\phi_{ij}^1 \cap \phi_{jk}^1 \neq \emptyset, \quad \phi_{ij}^1 \cap \phi_{ki}^1 \neq \emptyset, \quad \phi_{jk}^1 \cap \phi_{ki}^1 \neq \emptyset, \quad \text{and} \quad (4)$$

$$\phi_{ij}^1 \cap \phi_{jk}^1 \cap \phi_{ki}^1 = \emptyset. \quad (5)$$

Now, let $x \in \phi_{ij}^1 \cap \phi_{ki}^1$, and let $y \in \phi_{jk}^1 \cap \phi_{ki}^1$. It follows that the similarity set of x and y is a subset of ϕ_{ki} . Then, being ϕ_{ij}^1 and ϕ_{jk}^1 two compatible formae according to Eq.(4), two solutions $x \in \phi_{ij}^1$ and $y \in \phi_{jk}^1$ exist for which $\Sigma(\{x, y\}) \cap \phi_{ij}^1 \cap \phi_{jk}^1 = \emptyset$ (recall Eq.(5)). Hence, Φ is not separable. \square

The non-separability of Φ implies that a recombination operator processing the induced representation must manipulate macro-units of information in order to preserve feasibility in the descendant being created. These macro-units are termed *transmission sets*, and can be defined as follows:

Definition 1. *The transmission set $T(\Psi, \eta)$ of an allele η is the closure of the following expressions:*

$$\eta \triangleright T(\Psi, \eta) \quad (6)$$

$$[\exists! \eta' \in \Xi_{\phi'} : \Psi \cap T(\Psi, \eta) \cap \eta' \neq \emptyset] \Rightarrow \eta' \triangleright T(\Psi, \eta), \quad (7)$$

where Ψ is the partially constructed descendant.

Transmission sets are a more general version of *compatibility sets* [3], in which the descendant is not forced to belong to the dynastic potential of the solutions x and y being recombined. Nevertheless, transmission sets can be shown to preserve feasibility *within* $\Gamma(\{x, y\})$ as long as Ψ is initialized as the similarity set $\Sigma(\{x, y\})$. This is formally established in the following proposition:

Proposition 3. *Let $|\Xi_{\phi}| = 2$ for all $\phi \in \Psi$. Then, $\Upsilon(\Psi, \eta, x, y) \equiv T(\Psi, \eta)$ for any two solutions x and y , whenever $\Psi \triangleright \Sigma(\{x, y\})$.*

Proof. (Sketch) Essentially, a compatibility set $\Upsilon(\Psi, \eta, x, y)$ is the intersection of all those alleles present in x or y that must be injected in the descendant $z \in \Psi$, so as to ensure feasibility, simultaneously forcing $z \in \Gamma(\{x, y\})$. It can be seen that when $|\Xi_{\phi}| = 2$, either $x, y \in \eta$ for some $\eta \in \phi$ (and hence $\eta \triangleright \Sigma(\{x, y\})$), or $x \in \eta$ and $y \in \eta'$, $\Xi_{\phi} = \{\eta, \eta'\}$. This means that any feasible solution $z \in \Sigma(\{x, y\})$ also belongs to $\Gamma(\{x, y\})$, i.e., $\Sigma(\{x, y\}) \equiv \Gamma(\{x, y\})$. Thus, a transmission set will ensure feasibility within $\Gamma(\{x, y\})$ whenever $\Psi \triangleright \Sigma(\{x, y\})$. \square

This initial transmission of common alleles has been the approach considered in this work. Within this context, transmission sets must be computed taking into account that if an edge (i, j) is added to the descendant, any edge (j, k) –where k is an ancestor of i in the partially-specified solution– is forbidden. On the contrary, excluding an edge (i, j) from the descendant does not impose any restriction on the addition/exclusion of other edges. This can be formalized as follows: let Ψ be a partially specified descendant; let $\Omega(\Psi)$ be a DAG such that $\phi_{ij}^1 \triangleright \Psi \Leftrightarrow (i, j) \in E(\Omega(\Psi))$, i.e., $\Omega(\Psi)$ is a graph exclusively composed of the edges already added to the descendant; let C_D be the adjacency matrix of a digraph D , and let C_D^∞ be the transitive closure of C_D ; then

$$T(\Psi, \phi_{ij}^0) = \phi_{ij}^0, \quad \text{and} \quad (8)$$

$$T(\Psi, \phi_{ij}^1) = \phi_{ij}^1 \cap \bigcap_{C_{rs}^\oplus=1} \phi_{sr}^0 \quad (9)$$

where $C^\oplus = C_{\Omega(\Psi)}^\infty \text{ XOR } C_{\Omega(\Psi \cap \phi_{ij}^1)}^\infty$. Now, two main approaches to recombination can be considered on the basis of these units. Both start from $\Sigma(\{x, y\})$; subsequently, an unspecified gene is selected and an allele for that gene (and the

corresponding transmission set) is taken from either of the parents. The process is repeated until all genes are specified (gene transmission – GT), or until a certain feasibility criterion is met, being unspecified genes assigned a default value (allele transmission – AT). In the former situation, only *positive* alleles (i.e., ϕ_{ij}^1 -like alleles) are explicitly transmitted, being the default value a *negative* allele (i.e., ϕ_{ij}^0 -like). Both approaches are studied below.

4 Experimental Results

The first issue to be tackled is the computational cost of the operators. Calculating the connectivity matrix of a graph can be naïvely done in $O(n^3)$ using Dijkstra’s algorithm. Since the number of compatibility sets to be calculated is $O(n^2)$, this would amount to $O(n^5)$, but this is a very overestimated bound (incompatible worst-cases are superimposed).

To obtain a more realistic bound, an empirical assessment has been done. To be precise 200 random DAGs of fixed density and increasing size (5 up to 55 nodes) have been generated and recombined. The results are shown in Fig. 1 for GT and in in Fig. 2 for AT ². As it can be seen, a cubic behavior in the number of vertices is found. This cubic pattern is present in both GT and AT, with independence of the graph density of the DAGs recombined. Obviously, the larger the density of the DAGs being recombined, the larger the number of transmission sets that must be computed using Eq. (9). This results in a larger proportionality coefficient. In any case, the grow of this coefficient ceases and stabilizes around $\delta = .5$ (see Fig. 4-left); no significant increase of computational cost is observed in GT beyond that point. Notice also that this cubic behavior in the number of vertices implies a sub-quadratic behaviour ($O(|x|^{3/2})$) in the DAG size since $|x| \in O(n^2)$.

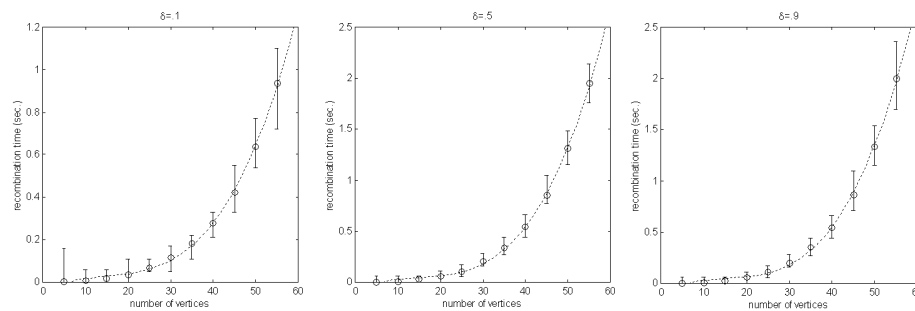


Fig. 1. Time required to recombine two DAGs using GT for increasing graph sizes. The results are shown for different graph densities: $\delta = .1$ (left), $\delta = .5$ (middle), and $\delta = .9$ (right). A fit to a cubic polynomial is shown.

² All times have been measured using MatLab[®] in a Pentium[®] III 600MHz.

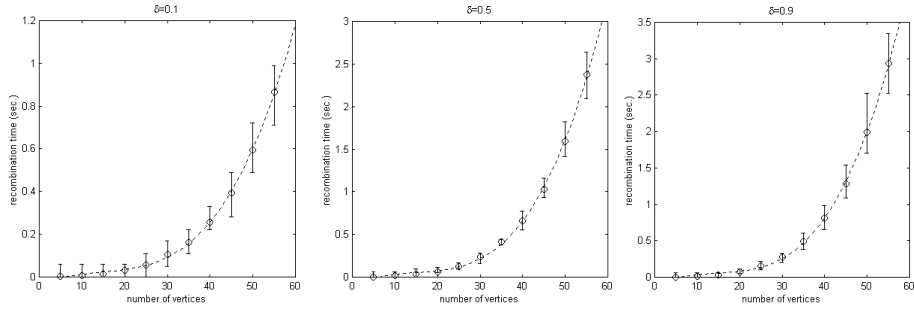


Fig. 2. Time required to recombine two DAGs using AT for increasing graph sizes. The results are shown for different graph densities: $\delta = .1$ (left), $\delta = .5$ (middle), and $\delta = .9$ (right). A fit to a cubic polynomial is shown.

Next, the transmission rates are studied. Due to the problem constraints, the density of the descendant is skewed to be lower than the mean density of the parents when using GT (according to Eq. (9) transmitting an edge of a parent can rule out a number of edges from the other parent). The higher the parents' mean density, the stronger this effect as shown in Fig. 3). The same is true for fixed density and increasing number of vertices. Actually, the descendant density seems to be related to the parents' mean density through an exponential function $\delta_{\text{descendant}} = \delta_{\text{parents}}^\alpha$ for some $\alpha < 1$. A fit to such a function reveals that -as in the previous case- the magnitude of α stabilizes around $\delta = .4$ or $\delta = .5$ (see Fig. 4-right). This skew in the generation of descendants can have a detrimental influence in the performance of a GA using GT, were high density DAGs sought. This fact is explored below.

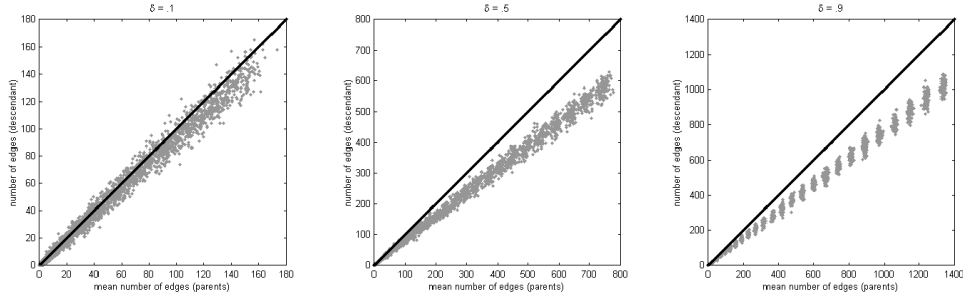


Fig. 3. Progeny density as a function of parental density using GT. The results are shown for different graph densities: $\delta = .1$ (left), $\delta = .5$ (middle), and $\delta = .9$ (right).

Experiments have been done using a steady-state genetic algorithm (binary tournament selection, $popsize=100$, $p_c = .9$, $p_m = 1/n^2$) in a DAG matching

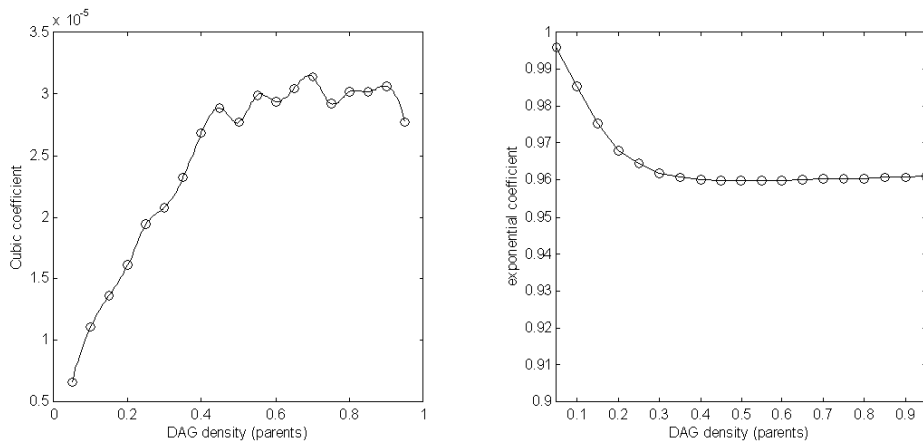


Fig. 4. (Left) Value of the cubic coefficient in the fitted-to-time polynomial. (Right) Value of the exponential coefficient in the fitted-to-density curve.

problem. Three scenarios are considered, consisting of finding a target DAG of a given density ($\delta \in \{.1, .5, .9\}$ in these experiments). The number of vertices is $n = 15$, and the fitness function is the number of different entries in the adjacency matrices of the actual DAG the target DAG. Besides GT and AT, two additional approaches are considered for comparison purposes: UX plus penalty term, and UX plus a repair function. The penalty term is computed in the first case as the $n^2 \cdot v$, where n is the total number of vertices, and v is the number of vertices connected to an ancestor of itself. As to the repair function, it consists of deleting edges from the graph until no loops remain. In both cases, the initial population is entirely composed of feasible solutions. Finally, the AT operator is adjusted to transmit a number of positive alleles following a distribution centered in the parents' mean number of edges (binomial, $p = 1/2$). This adjusted version of AT produces non-skewed descendants as shown in Fig. 5.

The results averaged for 50 runs are shown in Fig. 6. As it can be seen, AT performs similarly to GT when the density of the target DAG is low (the skew of GT is very small in this case). However, AT is increasingly better when the density of the target DAG is higher, being clearly superior for $\delta = .9$. The same holds with respect to the repairing and penalizing approaches. Again, the difference is stronger when the density of the target DAG is higher, since the chances of getting into the feasible region are lower using random recombination. Notice also that these two latter approaches are straightforwardly defined on this simple problem, but they can be difficult to apply in a more complex situation, e.g., the fitness function might be undefined for infeasible solutions, the repair function might introduce excessive disruption, etc.

Experiments have been also done with GAR (GT starting from a fully unspecified descendant rather than from $\Sigma(\{x, y\})$). The results are virtually identical

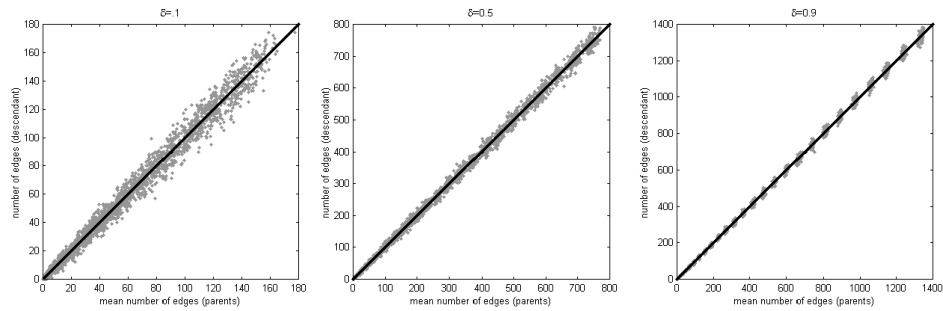


Fig. 5. Progeny density as a function of parental density using adjusted AT. The results are shown for different graph densities: $\delta = .1$ (left), $\delta = .5$ (middle), and $\delta = .9$ (right).

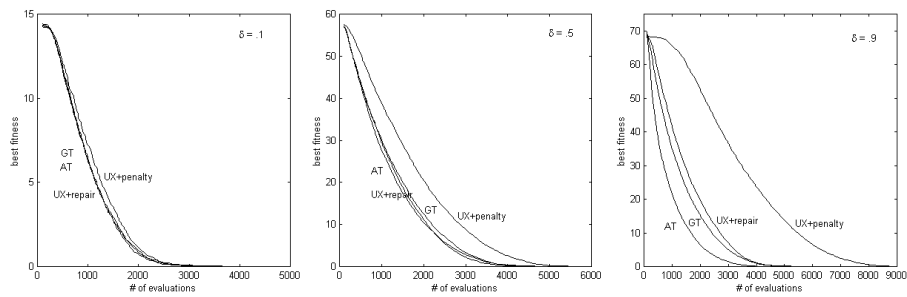


Fig. 6. Results (averaged for 50 runs) of a genetic algorithm using different recombination operators. From left to right: $\delta = .1$, $\delta = .5$, and $\delta = .9$.

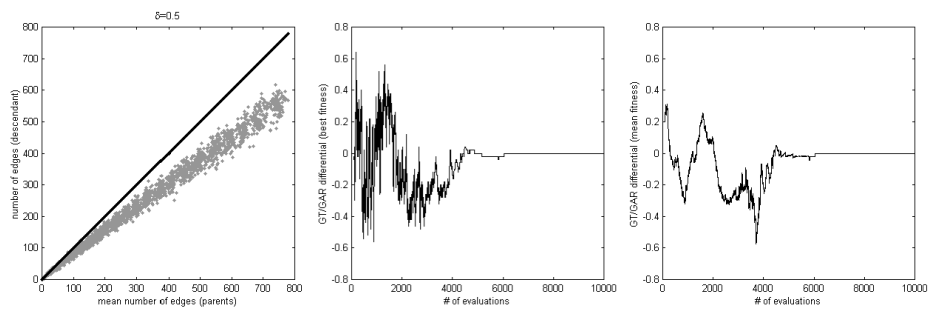


Fig. 7. (Left) Progeny density as a function of parental density using adjusted GAR for $\delta = .5$. (Middle) Best-fitness differential for GT vs. GAR (results averaged for 50 runs). (Right) Mean-fitness differential for GT vs. GAR (results averaged for 50 runs).

to GT (see Fig. 7-middle/right; the fitness differential closely oscillates around zero). This is an expected result, since an off-line assessment shows that the transmission rates of GAR are very similar to GT (Fig. 7-left).

5 Conclusions

This work has studied an edge-based representation of DAGs, and the properties of recombination operators working on it. Regarding this representation, its non-separability has been established, and the structure of the basic units that must be processed in order to maintain feasibility has been described. As to the operators, it has been shown that recombination in the feasible space can be done in sub-quadratic time in the DAG size. It has been also shown that a gene-transmission operator (GT) is skewed to generate DAGs with lower density than the parents average; this can be remedied by using an adjusted version of an allele-transmission operator (AT) that is shown to perform better than both GT and other approaches on a DAG matching problem.

Future work will be directed to confirm these results of other test problems. An especially interesting case is restricting the search to a subset of \mathcal{S}_V (e.g., designing causal networks using low-order statistics to build an undirected skeleton of the network [4]).

Acknowledgement

This work is partially supported by the Spanish *Comisión Interministerial de Ciencia y Tecnología* (CICYT) under grant TIC99-0754-C03-03.

References

1. K. Chen, J. Cong, Y. Ding, A. Kahng, and P. Trajmar. DAG-Map: Graph-based FPGA technology mapping for delay optimization. *IEEE Design and Test of Computers*, pages 7–20, September 1992.
2. M. Cosnard and M. Loi. Automatic task graphs generation techniques. *Parallel Processing Letters*, 5(4):527–538, 1995.
3. C. Cotta and J.M. Troya. On the influence of the representation granularity in heuristic forma recombination. In J. Carroll, E. Damiani, H. Haddad, and D. Oppenheim, editors, *ACM Symposium on Applied Computing 2000*, pages 433–439. ACM Press, 2000.
4. L.M. de Campos and J.F. Huete. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning*, 24(1):11–37, 2000.
5. H. Ehrenburg. Improved direct acyclic graph handling and the combine operator in genetic programming. In J.R. Koza, D.E. Goldberg, D.B. Fogel, and R.L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 285–291, Stanford University, CA, 1996. MIT Press.

6. S. Handley. On the use of a directed acyclic graph to represent a population of computer programs. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, pages 154–159, Orlando, FL, 1994. IEEE Press.
7. D. Heckerman and M. Wellman. Bayesian networks. *Communications of the ACM*, 38:27–30, 1995.
8. D. Kuck, R. Kuhn, B. Leasure, D. Padua, and M. Wolfe. Dependence graphs and compiler optimizations. In *Proceedings of the Eighth Annual ACM Symposium on Principles of Programming Languages*, pages 207–218, 1981.
9. P. Larrañaga, C. Kuijpers, R. Murga, and Y. Yurramendi. Learning bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on System, Man and Cybernetics*, 26(4):487–493, 1996.
10. K. Murphy. An introduction to graphical models. Technical report, Intel Research, May 2001.
11. J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Palo Alto, CA, 1988.
12. R. Poli. Parallel distributed genetic programming. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 403–432. McGraw-Hill, 1999.
13. N.J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–205, 1991.
14. N.J. Radcliffe. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10:339–384, 1994.
15. R. Turner, Song Li, and E. Gobbetti. Metis - an object-oriented toolkit for constructing virtual reality applications. *Computer Graphics Forum*, 18(2):121–130, 1999.