Digenes, a new tool using genetic algorithms for directed extremal graph theory

Romain Absil, Hadrien Mélot

University of Mons

UMONS Université de Mons

Faculté des Sciences

Computers in Scientific Discovery 6 - 2012

Table of contents

1 Introduction

2 Graph theory discovery tools

3 Extremal graph theory

4 A new tool
- Digenes
- Genetic algorithms

5 Conclusion and perspectives

Table of contents

1 Introduction

2 Graph theory discovery tools

3 Extremal graph theory

4 A new tool
- Digenes
- Genetic algorithms

5 Conclusion and perspectives

Table of contents

1 Introduction

2 Graph theory discovery tools

3 Extremal graph theory

4 A new tool
- Digenes
- Genetic algorithms

5 Conclusion and perspectives

Table of contents

1 Introduction

2 Graph theory discovery tools

3 Extremal graph theory

4 A new tool
- Digenes
- Genetic algorithms

5 Conclusion and perspectives

Table of contents

1 Introduction

2 Graph theory discovery tools

3 Extremal graph theory

4 A new tool
- Digenes
- Genetic algorithms

5 Conclusion and perspectives

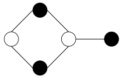
Invariants

Definition

A *graph invariant* is a (numerical) value preserved by isomorphism.

Example

- number n of vertices, number m or arrows or edges,
- chromatic number χ,
- stability number α, etc.



$n = m = 5$

$\chi = 2$

$\alpha = 3$

Graph theory research helping systems (1/2).

- Since 1981 [Cvetković, Kraus et Simić, 1981] [1], there exists dozens of tools used for graph theory [Hansen et Mélot, 2002] [2], such as
  - Graffiti [Fajtlowicz, 1987] [3],
  - AutoGraphiX (AGX) [Caporossi et Hansen, 2000] [4],
  - GraPHedron (GPH) [Mélot, 2008] [5].

1. Discussing Graph Theory with a Computer, I : Implementation of Graph Theoretic Algorithms. *Univ. Beograd Publ. Elektrotehn. Fak, Ser. Mat. Fiz. No. 734* (1981), 100 - 104.

2. Computers and Discovery in Algebraic Graph Theory. *Linear Algebra and its Applications 356*, 211 - 230.

3. On Conjectures of Graffiti. *Congr. Numer. 60*, 187 - 197.

4. Variable Neighborhod Search for Extremal Graphs 1. The AutoGraphiX System. *Discrete Math. 212*, 29 - 44.

5. Facet Defining Inequalities among Graph Invariants : the system GraPHedron. *Discrete Applied Mathematics 156*, 1875 - 1891

Graph theory research helping systems (2/2).

- More concretely, most of these systems allows to :
  - automatically generate conjectures given a specified class of graphs,
  - refute or support conjectures,
  - help in proof making ("good examples" identification, large enumeration, etc.).
- Hansen [6] states the systems Graph, AGX and Graffiti led alone to the publication of more than 200 papers.

6. How Far Should, Is And Could Be Conjecture-Making Automated in Graph Theory ? In *Graphs and Discovery*, Fajtlowicz, S. *et al.*, Ed. vol 69 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. AMS, Providence, 2005, pp. 189 - 230.

Graph theory research helping systems (2/2).

- More concretely, most of these systems allows to :
  - automatically generate conjectures given a specified class of graphs,
  - refute or support conjectures,
  - help in proof making ("good examples" identification, large enumeration, etc.).
- Hansen [6] states the systems Graph, AGX and Graffiti led alone to the publication of more than 200 papers.

6. How Far Should, Is And Could Be Conjecture-Making Automated in Graph Theory ? In *Graphs and Discovery*, Fajtlowicz, S. *et al.*, Ed. vol 69 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. AMS, Providence, 2005, pp. 189 - 230.

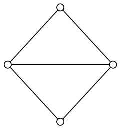
Conjectures' form

- Results in graph theory : often equalities or inequalities among graph invariants [Hansen, Aouchiche, Caporossi, Mélot, Stevanovic, 2005] [7].

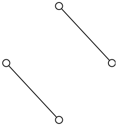
Example

If G is connected,

$$m \geqslant n - 1$$



$5 \geqslant 3$



$2 < 3$

7. What Forms Do Interesting Conjectures Have in Graph Theory ? *DIMACS Series in Discrete Mathematics and Theoretical Computer Science 69* 231 - 232

Extremal graph theory

- Maximize or minimize graph invariants,
- Find (tight) higher or lower bounds on invariants,
- Find extremal graphs for these inequalities, etc.

Extremal graph theory

- Maximize or minimize graph invariants,
- Find (tight) higher or lower bounds on invariants,
- Find extremal graphs for these inequalities, etc.

Extremal graph theory

- Maximize or minimize graph invariants,
- Find (tight) higher or lower bounds on invariants,
- Find extremal graphs for these inequalities, etc.

Túran problem (1/2)

Problem

What is the minimum number m of edges in a graph with n vertices and given stability number α ?

Theorem (Túran[8], 1941)

For all graph G with n vertices and stability number α,

$$m \geqslant m(T(n, \alpha)).$$

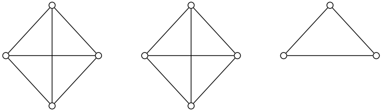
8. On an extremal problem in graph theory (in Hungarian). *Matematikai és Fizikai Lapok 48* 446-552.

Túran's problem (2/2)

Definition

A Túran graph $T(n, \alpha)$ with n vertices is the union of α balanced disjoint complete graphs (of size $\lfloor \frac{n}{\alpha} \rfloor$ ou $\lceil \frac{n}{\alpha} \rceil$).

$T(11, 3)$



Extremal graph theory historical problem.

Túran problem variant

Problem (Ore[9], 1962)

What is the minimum number m of edges in a *connected* graph with n vertices and given stability number α?

Solution (Labbé *et al.*[10], 2008)

$$m \geqslant \left(\left\lceil \frac{n}{2} \right\rceil\right)\left(n - \frac{\alpha}{2}\left\lceil \frac{n}{\alpha} \right\rceil\right) + \alpha - 1$$

- Established with GraPHedron.

9. Theory of graphs, *American Mathematical Society Colloquium Publications, 38*, American Mathematical Society, Providence, R.I.

10. Christophe J., Dewez S., Doignon J.-P., Fasbender G., Grégoire P., Huygens D., Labbé M., Elloumi S., Mélot H. et Yaman H. Linear inequalities among graph invariants : Using GraPHedron to uncover optimal relationships. *Networks 52* 287–298.

About Metaheuristics

- Exacts methods : ensures you find the global optimum,
  - Example : enumeration.
- Metaheuristics : general and ensures to find an (assumed good) answer quickly.

Local search

1. Idea : for all x, define the *neighbourhood* $N(x)$ de x.
2. Iteratively seek $N(x)$.
3. Improve current solution if possible.

- Two classes : *single solution* or *population*.

About Metaheuristics

- Exacts methods : ensures you find the global optimum,
  - Example : enumeration.
- Metaheuristics : general and ensures to find an (assumed good) answer quickly.

Local search

1. Idea : for all x, define the *neighbourhood* $N(x)$ de x.
2. Iteratively seek $N(x)$.
3. Improve current solution if possible.

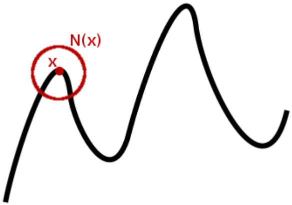
- Two classes : *single solution* or *population*.

About Metaheuristics

- Exacts methods : ensures you find the global optimum,
  - Example : enumeration.
- Metaheuristics : general and ensures to find an (assumed good) answer quickly.

Local search

1. Idea : for all x, define the *neighbourhood* $N(x)$ de x.
2. Iteratively seek $N(x)$.
3. Improve current solution if possible.



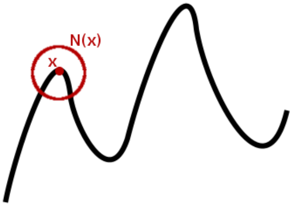
- Two classes : *single solution* or *population*.

About Metaheuristics

- Exacts methods : ensures you find the global optimum,
  - Example : enumeration.
- Metaheuristics : general and ensures to find an (assumed good) answer quickly.

Local search

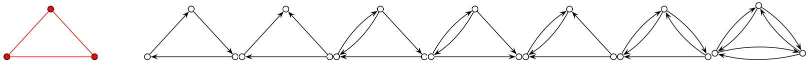
1. Idea : for all x, define the *neighbourhood* $N(x)$ de x.
2. Iteratively seek $N(x)$.
3. Improve current solution if possible.



- Two classes : *single solution* or *population*.

Field overview

- There are no systems dealing with directed graphs.
- An exact approach, such as enumeration (e.g., used in GrapPHedron), is inapplicable in the directed case.



Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Digenes

- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

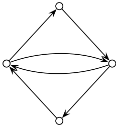
Digenes

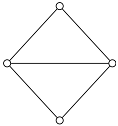
- A system helping discovery in graph theory using genetic algorithms.
- Population metaheuristic : diversification, particular individuals, etc.
- This system allows to
  - follow invariant evolution under graph transformation,
  - find (supposed) extremal graphs for a given invariant,
  - "validate" transformations.
- Includes directed and undirected graphs.
- Works at fixed order.
- Trivially provides tools to help the study of directed and undirected graphs, as well as comparisons between the two models.

Price of orientation

Definition

Graph $\widehat{G}$: undirected version of G.





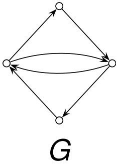
Price of orientation problem

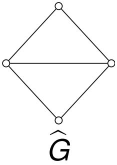
Let G_n the space of graphs of order n et $\mathcal{I}$ an invariant, what are the $G \in G_n$ maximizing or minimizing $\mathcal{I}(G) - \mathcal{I}(\widehat{G})$? What are the $G \in G_n$ maximizing or minimizing $\frac{\mathcal{I}(G)}{\mathcal{I}(\widehat{G})}$?

Price of orientation

Definition

Graph $\widehat{G}$: undirected version of G.



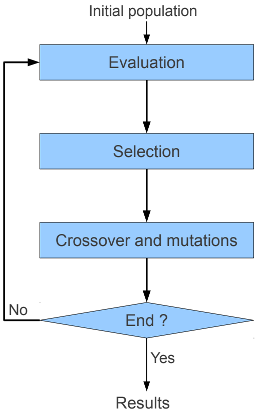


Price of orientation problem

Let G_n the space of graphs of order n et $\mathcal{I}$ an invariant, what are the $G \in G_n$ maximizing or minimizing $\mathcal{I}(G) - \mathcal{I}(\widehat{G})$? What are the $G \in G_n$ maximizing or minimizing $\frac{\mathcal{I}(G)}{\mathcal{I}(\widehat{G})}$?

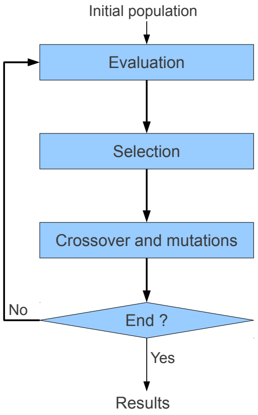
Genetic algorithms

- A solution (feasible or not) = an *individual*.
- Definition of *fitness* for each individual.
- Crossover, selection and mutation operators.
- Idea : emulate Darwin evolution theory.
- Use in practical cases (NASA, Airbus, etc.).



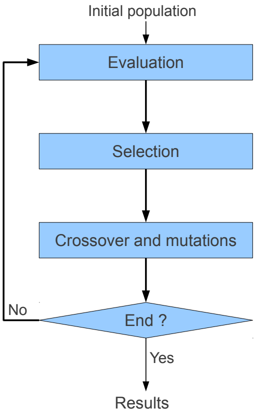
Genetic algorithms

- A solution (feasible or not) = an *individual*.
- Definition of *fitness* for each individual.
- Crossover, selection and mutation operators.
- Idea : emulate Darwin evolution theory.
- Use in practical cases (NASA, Airbus, etc.).



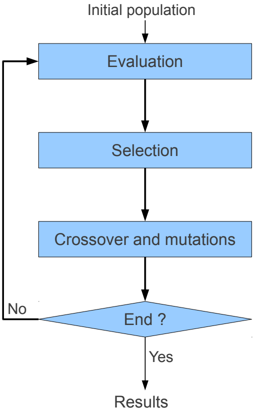
Genetic algorithms

- A solution (feasible or not) = an *individual*.
- Definition of *fitness* for each individual.
- Crossover, selection and mutation operators.
- Idea : emulate Darwin evolution theory.
- Use in practical cases (NASA, Airbus, etc.).



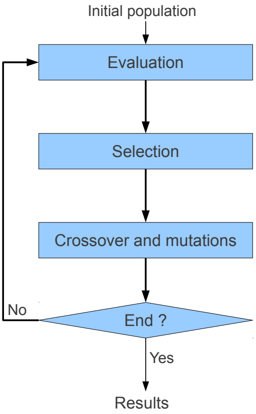
Genetic algorithms

- A solution (feasible or not) = an *individual*.
- Definition of *fitness* for each individual.
- Crossover, selection and mutation operators.
- Idea : emulate Darwin evolution theory.
- Use in practical cases (NASA, Airbus, etc.).



Genetic algorithms

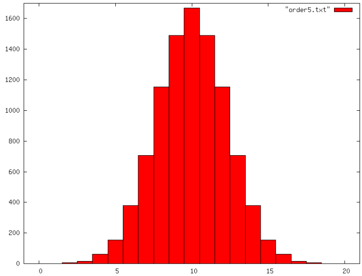
- A solution (feasible or not) = an *individual*.
- Definition of *fitness* for each individual.
- Crossover, selection and mutation operators.
- Idea : emulate Darwin evolution theory.
- Use in practical cases (NASA, Airbus, etc.).



Initial population

Problem

- How to spread a population well ?
  - Consider isomorphisms
- How to do it quickly ?

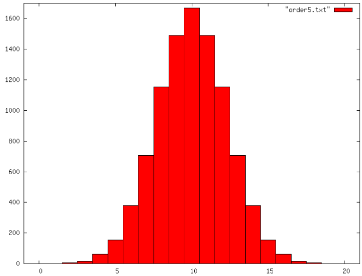


- Impossible to define a (polynomial) relevant norm or distance.
- Simplify the problem.

Initial population

Problem

- How to spread a population well ?
  - Consider isomorphisms
- How to do it quickly ?

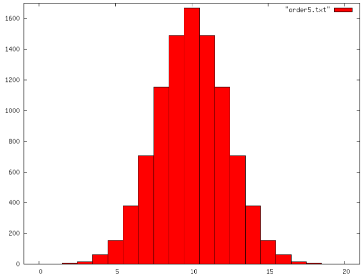


- Impossible to define a (polynomial) relevant norm or distance.
- Simplify the problem.

Initial population

Problem

- How to spread a population well ?
  - Consider isomorphisms
- How to do it quickly ?



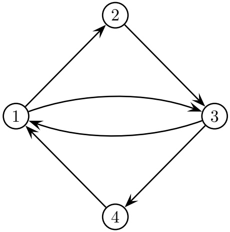
- Impossible to define a (polynomial) relevant norm or distance.
- Simplify the problem.

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.

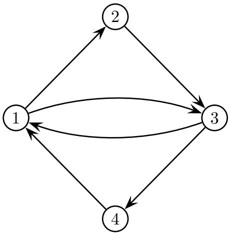


Adjacency matrix A_{ij}

	1	2	3	4
1	0	1	1	0
2	0	0	1	0
3	1	0	0	1
4	1	0	0	0

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.



Degree sequence

In-degrees	2	2	1	1
Out-degrees	2	2	1	1

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.
- Random arrow

Principle

- Let $p \in]0, 1[$, each arrow (i, j) has a probability p to be in G.
- An average of $pn(n-1)$ arrows is selected.

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.
- Random size

Principle

- A size $k \in \left[0, \frac{n(n-1)}{2}\right]$ is randomly generated.
- A random graph of the specified size is then generated.

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.
- Blocks of graphs of fixed size.

Principle

- n_k graphs with k arrows are generated, with $k \in \left[0, \frac{n(n-1)}{2}\right]$.
- Wisely choose the n_k (depends on k).

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.
- Uniform degree sequence

Principle

- Uniformly generate lists in the list space,
- If the sequence if a graph, keep it, otherwise, drop it and restart.

Initial population

- Spread over the graph coding :
  - Adjacency matrix A_{ij},
  - Degree sequence.
- Non uniform degree sequence

Principle

- For each element i in the sequence, randomly generate i in $\left[0, \frac{n(n-1)}{2}\right]$.
- If the sequence if a graph, keep it, otherwise, drop it and restart.

Initial population : House of Graphs [Brinkmann *et al.*, 2012] [11]

- Principle overview : some (undirected) graphs are more interesting than others (cycles, paths, complete, bipartite, etc.).
- Database (of small size) of these graphs.
- Idea : use these graphs in the initial population.
- Build relevant (as well as random) orientations of these graphs.

11. House of Graphs : a database of interesting graphs *Discrete Math.*, To be published.

Initial population : House of Graphs [Brinkmann *et al.*, 2012] [11]

- Principle overview : some (undirected) graphs are more interesting than others (cycles, paths, complete, bipartite, etc.).
- Database (of small size) of these graphs.
- Idea : use these graphs in the initial population.
- Build relevant (as well as random) orientations of these graphs.

11. House of Graphs : a database of interesting graphs *Discrete Math.*, To be published.

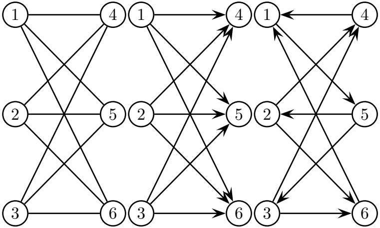
Initial population : House of Graphs [Brinkmann *et al.*, 2012] [11]

- Principle overview : some (undirected) graphs are more interesting than others (cycles, paths, complete, bipartite, etc.).
- Database (of small size) of these graphs.
- Idea : use these graphs in the initial population.
- Build relevant (as well as random) orientations of these graphs.

11. House of Graphs : a database of interesting graphs *Discrete Math.*, To be published.

Initial population : House of Graphs [Brinkmann *et al.*, 2012] [11]

- Principle overview : some (undirected) graphs are more interesting than others (cycles, paths, complete, bipartite, etc.).
- Database (of small size) of these graphs.
- Idea : use these graphs in the initial population.
- Build relevant (as well as random) orientations of these graphs.



11. House of Graphs : a database of interesting graphs *Discrete Math.*, To be published.

Results among several populations : average distance

Optimal fitness : 5, on the graph $\mathcal{C}_{10}$.

Problem 1	Average fitness	Average time [1]	Success
Random arrow	4.860444	1065.236	0.635
Random size	4.894944	1183.69	0.695
Constant size blocks	4.8975	1113.07	0.74
Uniform size blocks	4.9728	893.56	0.86
Uniform degrees	4.847222	1160.638	0.595
Non uniform degrees	4.884722	1132.106	0.705
House of Graphs	4.949277	769.321	0.87

1. Time is given as number of iterations needed to reach the optimum.

Results among different populations : diameter

Optimal fitness : 9, on several graphs.

Problem 2	Average fitness	Average time [2]	Success
Random arrow	9.0	168.4	1.0
Random size	9.0	105.53	1.0
Constant size blocks	9.0	119.24	1.0
Uniform size blocks	9.0	105.6	1.0
Uniform degrees	9.0	220.665	1.0
Non uniform degrees	9.0	229.105	1.0
House of Graphs	9.0	2.0	1.0

2. Time is given as number of iterations needed to reach the optimum.

Results among different populations : price of orientation and diameter

Optimal fitness : 8, on the graph $\mathcal{B}_{10}$.

Problem 3	Average fitness	Average time [2]	Success
Random arrow	7.0	NaN	0.0
Random size	7.0	NaN	0.0
Constant size blocks	7.0	NaN	0.0
Uniform size blocks	7.0	NaN	0.0
Uniform degrees	7.0	NaN	0.0
Non uniform degrees	7.0	NaN	0.0
House of Graphs	7.0	NaN	0.0

3. Time is given as number of iterations needed to reach the optimum.

Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

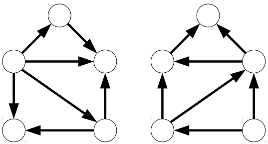
Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

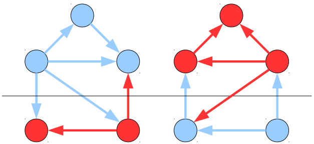
High level crossover



Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

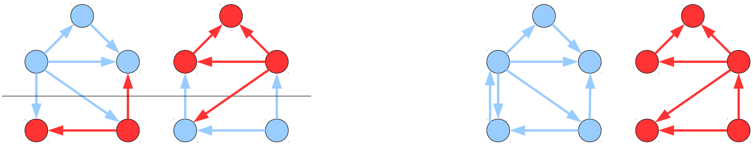
High level crossover



Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

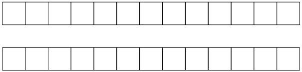
High level crossover



Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

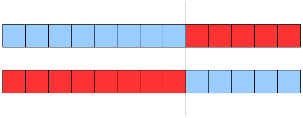
Low level crossover : one-point



Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

Low level crossover : one-point



Crossovers

- Idea : how to mate individuals ?
- Might be "high level" (structural) or "low level" (coding).

Low level crossover : one-point

Selection

- Idea : how to mate progenitors ? How to decide who will live ?
- Several schemes in literature.
- Global principle : an healthy individual (high fitness) has higher chances to mate / survive than a sick (low fitness) individual.

Selection

- Idea : how to mate progenitors ? How to decide who will live ?
- Several schemes in literature.
- Global principle : an healthy individual (high fitness) has higher chances to mate / survive than a sick (low fitness) individual.

Selection

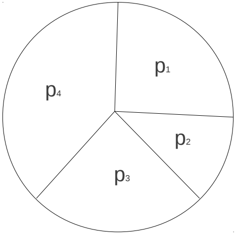
- Idea : how to mate progenitors ? How to decide who will live ?
- Several schemes in literature.
- Global principle : an healthy individual (high fitness) has higher chances to mate / survive than a sick (low fitness) individual.

Selection

- Idea : how to mate progenitors ? How to decide who will live ?
- Several schemes in literature.
- Global principle : an healthy individual (high fitness) has higher chances to mate / survive than a sick (low fitness) individual.

Roulette selection

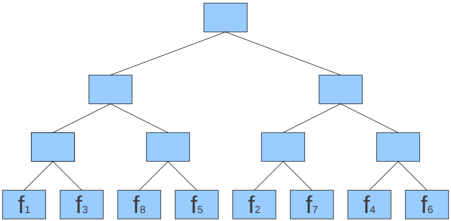
An individual i has a probability $p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$ to be selected.



Selection

- Idea : how to mate progenitors ? How to decide who will live ?
- Several schemes in literature.
- Global principle : an healthy individual (high fitness) has higher chances to mate / survive than a sick (low fitness) individual.

Tournament selection
Randomly put individuals on a direct elimination tournament grid.



Mutations

- Idea : individuals have chances to spontaneously mutate.
- Two schools about mutations : few "big" mutations, or many "small" mutations.
- Small mutations : flip the value of one or more bits in the coding.
- Big mutations : partially modify the structure :
  - Trees : add / remove a sub-tree, sub-tree rotations, etc.
  - Graphs : make k local disturbances : arrow rotations, 2-Opt, add / remove arrows, etc.

Mutations

- Idea : individuals have chances to spontaneously mutate.
- Two schools about mutations : few "big" mutations, or many "small" mutations.
- Small mutations : flip the value of one or more bits in the coding.
- Big mutations : partially modify the structure :
  - Trees : add / remove a sub-tree, sub-tree rotations, etc.
  - Graphs : make k local disturbances : arrow rotations, 2-Opt, add / remove arrows, etc.

Mutations

- Idea : individuals have chances to spontaneously mutate.
- Two schools about mutations : few "big" mutations, or many "small" mutations.
- Small mutations : flip the value of one or more bits in the coding.
- Big mutations : partially modify the structure :
  - Trees : add / remove a sub-tree, sub-tree rotations, etc.
  - Graphs : make k local disturbances : arrow rotations, 2-Opt, add / remove arrows, etc.

Mutations

- Idea : individuals have chances to spontaneously mutate.
- Two schools about mutations : few "big" mutations, or many "small" mutations.
- Small mutations : flip the value of one or more bits in the coding.
- Big mutations : partially modify the structure :
  - Trees : add / remove a sub-tree, sub-tree rotations, etc.
  - Graphs : make k local disturbances : arrow rotations, 2-Opt, add / remove arrows, etc.

Mutations

- Idea : individuals have chances to spontaneously mutate.
- Two schools about mutations : few "big" mutations, or many "small" mutations.
- Small mutations : flip the value of one or more bits in the coding.
- Big mutations : partially modify the structure :
  - Trees : add / remove a sub-tree, sub-tree rotations, etc.
  - Graphs : make k local disturbances : arrow rotations, 2-Opt, add / remove arrows, etc.

Mutations

- Idea : individuals have chances to spontaneously mutate.
- Two schools about mutations : few "big" mutations, or many "small" mutations.
- Small mutations : flip the value of one or more bits in the coding.
- Big mutations : partially modify the structure :
  - Trees : add / remove a sub-tree, sub-tree rotations, etc.
  - Graphs : make k local disturbances : arrow rotations, 2-Opt, add / remove arrows, etc.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Conclusions

- Initial population switching has far more impact than mutation, crossover and selection operator switching.
- Switching initial population from a generator to another may have huge impact on convergence.
- Random uniform generators are less efficient (10%) then non uniform.
  - Some particular graphs are on "the edges of the space".
- *House of Graphs* improvement is very efficient.
  - Give good results even when removing some relevant orientations.
  - Fails when removing a relatively big part of the orientations.
- Results highly depend on the underlying problem.

Price of orientation and average distance

Conjecture

Let $G \in \mathscr{G}_n$, we have

$$\mu(\mathcal{C}_n) - \mu(\widehat{\mathcal{C}_n}) \geqslant \mu(G) - \mu(\widehat{G})$$

$$\frac{\mu(\mathcal{C}_n)}{\mu(\widehat{\mathcal{C}_n})} \geqslant \frac{\mu(G)}{\mu(\widehat{G})}$$

with equality if and only if $G \simeq \mathcal{C}_n$.

- Really complicated.

Price of orientation and average distance

Conjecture

Let $G \in \mathscr{G}_n$, we have

$$\mu(\mathcal{C}_n) - \mu(\widehat{\mathcal{C}_n}) \geqslant \mu(G) - \mu(\widehat{G})$$

$$\frac{\mu(\mathcal{C}_n)}{\mu(\widehat{\mathcal{C}_n})} \geqslant \frac{\mu(G)}{\mu(\widehat{G})}$$

with equality if and only if $G \simeq \mathcal{C}_n$.

- Really complicated.

Questions ?