Christopher R. Stephens
Marc Toussaint
Darrell Whitley
Peter F. Stadler (Eds.)

LNCS 4436

Foundations of Genetic Algorithms

9th International Workshop, FOGA 2007
Mexico City, Mexico, January 2007
Revised Selected Papers

Springer

Christopher R. Stephens Marc Toussaint
Darrell Whitley Peter F. Stadler (Eds.)

Foundations of Genetic Algorithms

9th International Workshop, FOGA 2007
Mexico City, Mexico, January 8-11, 2007
Revised Selected Papers

Springer

Volume Editors

Christopher R. Stephens
Universidad Nacional Autonoma de Mexico, Instituto de Ciencias Nucleares
Circuito Exterior, A. Postal 70-543, Mexico D.F. 04510, Mexico
E-mail: stephens@nucleares.unam.mx

Marc Toussaint
TU Berlin
Franklinstr. 28/29, 10587 Berlin, Germany
E-mail: mtoussai@cs.tu-berlin.de

Darrell Whitley
Colorado State University, Department of Computer Science
Fort Collins, CO 80523, USA
E-mail: whitley@cs.colostate.edu

Peter F. Stadler
Universität Leipzig, Institut für Informatik
Härtelstr. 16-18, 04107 Leipzig, Germany
E-mail: studla@bioinf.uni-leipzig.de

Library of Congress Control Number: 2007929644

CR Subject Classification (1998): F.1-2, I.2, I.2.6, I.2.8, D.2.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-73479-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73479-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12087204 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4436

Lecture Notes in Computer Science

For information about Vols. 1–4489

please contact your bookseller or Springer

Vol. 4600: H. Comon-Lundh, C. Kirchner, H. Kirchner, Rewriting, Computation and Proof. XVI, 273 pages. 2007.

Vol. 4595: D. Bošnački, S. Edelkamp (Eds.), Model Checking Software. X, 285 pages. 2007.

Vol. 4592: Z. Kedad, N. Lammari, E. Métais, F. Meziane, Y. Rezgui (Eds.), Natural Language Processing and Information Systems. XIV, 442 pages. 2007.

Vol. 4591: J. Davies, J. Gibbons (Eds.), Integrated Formal Methods. IX, 660 pages. 2007.

Vol. 4590: W. Damm, H. Hermanns (Eds.), Computer Aided Verification. XV, 562 pages. 2007.

Vol. 4589: J. Münch, P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. XII, 414 pages. 2007.

Vol. 4588: T. Harju, J. Karhumäki, A. Lepistö (Eds.), Developments in Language Theory. XI, 423 pages. 2007.

Vol. 4587: R. Cooper, J. Kennedy (Eds.), Data Management. XIII, 259 pages. 2007.

Vol. 4586: J. Pieprzyk, H. Ghodosi, E. Dawson (Eds.), Information Security and Privacy. XIV, 476 pages. 2007.

Vol. 4584: N. Karssemeijer, B. Lelieveldt (Eds.), Information Processing in Medical Imaging. XX, 777 pages. 2007.

Vol. 4583: S.R. Della Rocca (Ed.), Typed Lambda Calculi and Applications. X, 397 pages. 2007.

Vol. 4582: J. Lopez, P. Samarati, J.L. Ferrer (Eds.), Public Key Infrastructure. XI, 375 pages. 2007.

Vol. 4581: A. Petrenko, M. Veanes, J. Tretmans, W. Grieskamp (Eds.), Testing of Software and Communicating Systems. XII, 379 pages. 2007.

Vol. 4578: F. Masulli, S. Mitra, G. Pasi (Eds.), Fuzzy Logic and Applications. XVIII, 693 pages. 2007. (Sublibrary LNAI).

Vol. 4577: N. Sebe, Y. Liu, Y. Zhuang (Eds.), Multimedia Content Analysis and Mining. XIII, 513 pages. 2007.

Vol. 4576: D. Leivant, R. de Queiroz (Eds.), Logic, Language, Information, and Computation. X, 363 pages. 2007.

Vol. 4574: J. Derrick, J. Vain (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2007. XI, 375 pages. 2007.

Vol. 4573: M. Kauers, M. Kerber, R. Miner, W. Windsteiger (Eds.), Towards Mechanized Mathematical Assistants. XIII, 407 pages. 2007. (Sublibrary LNAI).

Vol. 4572: F. Stajano, C. Meadows, S. Capkun, T. Moore (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. X, 247 pages. 2007.

Vol. 4570: H.G. Okuno, M. Ali (Eds.), New Trends in Applied Artificial Intelligence. XXI, 1194 pages. 2007. (Sublibrary LNAI).

Vol. 4569: A. Butz, B. Fisher, A. Krüger, P. Olivier, S. Owada (Eds.), Smart Graphics. IX, 237 pages. 2007.

Vol. 4566: M.J Dainoff (Ed.), Ergonomics and Health Aspects of Work with Computers. XVIII, 390 pages. 2007.

Vol. 4565: D.D. Schmorrow, L.M. Reeves (Eds.), Foundations of Augmented Cognition. XIX, 450 pages. 2007. (Sublibrary LNAI).

Vol. 4564: D. Schuler (Ed.), Online Communities and Social Computing. XVII, 520 pages. 2007.

Vol. 4563: R. Shumaker (Ed.), Virtual Reality. XXII, 762 pages. 2007.

Vol. 4561: V.G. Duffy (Ed.), Digital Human Modeling. XXIII, 1068 pages. 2007.

Vol. 4560: N. Aykin (Ed.), Usability and Internationalization, Part II. XVIII, 576 pages. 2007.

Vol. 4559: N. Aykin (Ed.), Usability and Internationalization, Part I. XVIII, 661 pages. 2007.

Vol. 4549: J. Aspnes, C. Scheideler, A. Arora, S. Madden (Eds.), Distributed Computing in Sensor Systems. XIII, 417 pages. 2007.

Vol. 4548: N. Olivetti (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. X, 245 pages. 2007. (Sublibrary LNAI).

Vol. 4547: C. Carlet, B. Sunar (Eds.), Arithmetic of Finite Fields. XI, 355 pages. 2007.

Vol. 4546: J. Kleijn, A. Yakovlev (Eds.), Petri Nets and Other Models of Concurrency – ICATPN 2007. XI, 515 pages. 2007.

Vol. 4545: H. Anai, K. Horimoto, T. Kutsia (Eds.), Algebraic Biology. XIII, 379 pages. 2007.

Vol. 4544: S. Cohen-Boulakia, V. Tannen (Eds.), Data Integration in the Life Sciences. XI, 282 pages. 2007. (Sublibrary LNBI).

Vol. 4543: A.K. Bandara, M. Burgess (Eds.), Inter-Domain Management. XII, 237 pages. 2007.

Vol. 4542: P. Sawyer, B. Paech, P. Heymans (Eds.), Requirements Engineering: Foundation for Software Quality. IX, 384 pages. 2007.

Vol. 4541: T. Okadome, T. Yamazaki, M. Makhtari (Eds.), Pervasive Computing for Quality of Life Enhancement. IX, 248 pages. 2007.

Vol. 4539: N.H. Bshouty, C. Gentile (Eds.), Learning Theory. XII, 634 pages. 2007. (Sublibrary LNAI).

Vol. 4538: F. Escolano, M. Vento (Eds.), Graph-Based Representations in Pattern Recognition. XII, 416 pages. 2007.

Vol. 4537: K.C.-C. Chang, W. Wang, L. Chen, C.A. Ellis, C.-H. Hsu, A.C. Tsoi, H. Wang (Eds.), Advances in Web and Network Technologies, and Information Management. XXIII, 707 pages. 2007.

Vol. 4536: G. Concas, E. Damiani, M. Scotto, G. Succi (Eds.), Agile Processes in Software Engineering and Extreme Programming. XV, 276 pages. 2007.

Vol. 4534: I. Tomkos, F. Neri, J. Solé Pareta, X. Masip Bruin, S. Sánchez Lopez (Eds.), Optical Network Design and Modeling. XI, 460 pages. 2007.

Vol. 4531: J. Indulska, K. Raymond (Eds.), Distributed Applications and Interoperable Systems. XI, 337 pages. 2007.

Vol. 4530: D.H. Akehurst, R. Vogel, R.F. Paige (Eds.), Model Driven Architecture- Foundations and Applications. X, 219 pages. 2007.

Vol. 4529: P. Melin, O. Castillo, L.T. Aguilar, J. Kacprzyk, W. Pedrycz (Eds.), Foundations of Fuzzy Logic and Soft Computing. XIX, 830 pages. 2007. (Sublibrary LNAI).

Vol. 4528: J. Mira, J.R. Álvarez (Eds.), Nature Inspired Problem-Solving Methods in Knowledge Engineering, Part II. XXII, 650 pages. 2007.

Vol. 4527: J. Mira, J.R. Álvarez (Eds.), Bio-inspired Modeling of Cognitive Tasks, Part I. XXII, 630 pages. 2007.

Vol. 4526: M. Malek, M. Reitenspieß, A. van Moorsel (Eds.), Service Availability. X, 155 pages. 2007.

Vol. 4525: C. Demetrescu (Ed.), Experimental Algorithms. XIII, 448 pages. 2007.

Vol. 4524: M. Marchiori, J.Z. Pan, C.d.S. Marie (Eds.), Web Reasoning and Rule Systems. XI, 382 pages. 2007.

Vol. 4523: Y.-H. Lee, H.-N. Kim, J. Kim, Y. Park, L.T. Yang, S.W. Kim (Eds.), Embedded Software and Systems. XIX, 829 pages. 2007.

Vol. 4522: B.K. Ersbøll, K.S. Pedersen (Eds.), Image Analysis. XVIII, 989 pages. 2007.

Vol. 4521: J. Katz, M. Yung (Eds.), Applied Cryptography and Network Security. XIII, 498 pages. 2007.

Vol. 4519: E. Franconi, M. Kifer, W. May (Eds.), The Semantic Web: Research and Applications. XVIII, 830 pages. 2007.

Vol. 4517: F. Boavida, E. Monteiro, S. Mascolo, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 382 pages. 2007.

Vol. 4516: L. Mason, T. Drwiega, J. Yan (Eds.), Managing Traffic Performance in Converged Networks. XXIII, 1191 pages. 2007.

Vol. 4515: M. Naor (Ed.), Advances in Cryptology - EUROCRYPT 2007. XIII, 591 pages. 2007.

Vol. 4514: S.N. Artemov, A. Nerode (Eds.), Logical Foundations of Computer Science. XI, 513 pages. 2007.

Vol. 4513: M. Fischetti, D.P. Williamson (Eds.), Integer Programming and Combinatorial Optimization. IX, 500 pages. 2007.

Vol. 4511: C. Conati, K. McCoy, G. Paliouras (Eds.), User Modeling 2007. XVI, 487 pages. 2007. (Sublibrary LNAI).

Vol. 4510: P. Van Hentenryck, L. Wolsey (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 391 pages. 2007.

Vol. 4509: Z. Kobti, D. Wu (Eds.), Advances in Artificial Intelligence. XII, 552 pages. 2007. (Sublibrary LNAI).

Vol. 4508: M.-Y. Kao, X.-Y. Li (Eds.), Algorithmic Aspects in Information and Management. VIII, 428 pages. 2007.

Vol. 4507: F. Sandoval, A. Prieto, J. Cabestany, M. Graña (Eds.), Computational and Ambient Intelligence. XXVI, 1167 pages. 2007.

Vol. 4506: D. Zeng, I. Gotham, K. Komatsu, C. Lynch, M. Thurmond, D. Madigan, B. Lober, J. Kvach, H. Chen (Eds.), Intelligence and Security Informatics: Biosurveillance. XI, 234 pages. 2007.

Vol. 4505: G. Dong, X. Lin, W. Wang, Y. Yang, J.X. Yu (Eds.), Advances in Data and Web Management. XXII, 896 pages. 2007.

Vol. 4504: J. Huang, R. Kowalczyk, Z. Maamar, D. Martin, I. Müller, S. Stoutenburg, K.P. Sycara (Eds.), Service-Oriented Computing: Agents, Semantics, and Engineering. X, 175 pages. 2007.

Vol. 4501: J. Marques-Silva, K.A. Sakallah (Eds.), Theory and Applications of Satisfiability Testing – SAT 2007. XI, 384 pages. 2007.

Vol. 4500: N. Streitz, A. Kameas, I. Mavrommati (Eds.), The Disappearing Computer. XVIII, 304 pages. 2007.

Vol. 4499: Y.Q. Shi (Ed.), Transactions on Data Hiding and Multimedia Security II. IX, 117 pages. 2007.

Vol. 4498: N. Abdennahder, F. Kordon (Eds.), Reliable Software Technologies – Ada Europe 2007. XII, 247 pages. 2007.

Vol. 4497: S.B. Cooper, B. Löwe, A. Sorbi (Eds.), Computation and Logic in the Real World. XVIII, 826 pages. 2007.

Vol. 4496: N.T. Nguyen, A. Grzech, R.J. Howlett, L.C. Jain (Eds.), Agent and Multi-Agent Systems: Technologies and Applications. XXI, 1046 pages. 2007. (Sublibrary LNAI).

Vol. 4495: J. Krogstie, A. Opdahl, G. Sindre (Eds.), Advanced Information Systems Engineering. XVI, 606 pages. 2007.

Vol. 4494: H. Jin, O.F. Rana, Y. Pan, V.K. Prasanna (Eds.), Algorithms and Architectures for Parallel Processing. XIV, 508 pages. 2007.

Vol. 4493: D. Liu, S. Fei, Z. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part III. XXVI, 1215 pages. 2007.

Vol. 4492: D. Liu, S. Fei, Z. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part II. XXVII, 1321 pages. 2007.

Vol. 4491: D. Liu, S. Fei, Z.-G. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part I. LIV, 1365 pages. 2007.

Vol. 4490: Y. Shi, G.D. van Albada, J. Dongarra, P.M.A. Sloot (Eds.), Computational Science – ICCS 2007, Part IV. XXXVII, 1211 pages. 2007.

Editorial Introduction

Since their inception in 1990, the FOGA (Foundations of Genetic Algorithms) workshops have been one of the principal reference sources for theoretical developments in evolutionary computation (EC) and, in particular, genetic algorithms (GAs). The ninth such workshop, FOGA IX, was held at the Instituto de Ciencias Nucleares of the Universidad Nacional Autónoma de México, Mexico City during January 8–11, 2007.

One of the main reasons the FOGA series of conferences has had a large impact in EC has been its distinct profile as the only conference dedicated to theoretical issues of a "foundational" nature – both conceptual and technical. In this FOGA conference, and in keeping with this tradition, special attention was paid to the biological foundations of EC. The essential mathematical structure behind many evolutionary algorithms is the one familiar from population genetics, whose basic elements have been around now for at least 70 years. The last 20 years or so, however, have witnessed huge changes in our understanding of how genomes and other genetic structures work due to a plethora of new experimental techniques and results. How does this new phenomenology change our understanding of what genetic systems do and how they do it? And how can we design "better" ones?

In this spirit, the first 2 days of the conference consisted of organized discussions built around sets of lectures given by two world authorities on the "old" biology and the "new" biology – Reinhard Burger (University of Vienna) and Jim Shapiro (University of Chicago). The motivation behind this was that by a careful presentation of the main ideas, a useful transfer of knowledge of the latest developments and understanding of genetic dynamics in biology would be fruitful for the EC community in better understanding and designing artificial genetic systems. In particular the following questions were addressed:

- How do real genetic systems work?
- Why do they work that way?
- From this, what can we learn in order to design "better" artificial genetic systems?

One of the most important conclusions from this confrontation between the old and the new, was that the genotype – phenotype map and the huge variety of complex ways by which genomes can interchange and mix genetic material are not represented adequately in the standard "selection on a fixed fitness landscape, mutation and homologous recombination" picture so dominant in EC and, particularly, GAs. Secondly, it became clear that the canonical picture of population genetics was not an appropriate framework for considering "macroevolution" over long time scales, where the restructuring of genomes can be enormous. Both these facts potentially pose great challenges for EC. For instance, under what circumstances are all the diverse exchange and restructuring mechanisms for genomes useful in an EC setting? It is hard to imagine that optimizing the 3,456-city Travelling Salesman problem needs such sophisticated apparatus. Such a limited combinatorial optimization context is probably much more akin to the evolution of specific phenotypic characteristics, as treated in standard population genetics. No doubt that is one of the main reasons for the success of GAs in combinatorial optimization. However, it is not clear if such a paradigm is adequate for producing a more intelligent robot.

To understand then why biology uses certain representations and operators, it is necessary to understand what a biological system has to "do" when compared with EC systems. Surviving in an uncertain, time-dependent environment is surely an infinitely more complex task than finding a set of allele values that represent an optimal solution to a combinatorial optimization problem. In this sense, one may wonder if there are any biological systems that are at least similar to typical problems faced in EC. Peter Stadler presented probably one of the closest analogies – evolution of macromolecules in the context of an RNA world – where the fitness function for a particular RNA configuration is its replication rate. However, such simple chemical evolution seems far removed from the macroevolution of entire organisms. Hopefully, some of the fruits of this more intense examination of the relationship between biological evolution and EC will appear in the next FOGA.

The second two days of the conference were of a more standard FOGA format with contributed talks and ample time for discussion between them. For this workshop there were 22 submissions which were each sent in a double-blind review to three referees. Twelve high quality submissions that cover a wide range of theoretical topics were eventually accepted after two more rounds of revisions and are presented in this volume.

We would like to thank our co-organizers, Peter Stadler and Darrell Whitley, for their efforts and input. Katya Rodríguez formed part of the Local Organizing Committee and played an important role in making the conference run smoothly, as did Trinidad Ramírez and various student helpers. Thanks go to the Instituto de Ciencias Nucleares for providing its facilities and to the Macroproyecto Tecnologias para la Universidad de la Información y de la Computación for financial and technical support.

April 2007

Christopher R. Stephens
Marc Toussaint

Organization

FOGA 2007 was organized in cooperation with ACM/SIGEVO at the Instituto de Ciencias Nucleares, Universidad Nacional Autonoma de Mexico (UNAM), Mexico City, January 8–11, 2007.

Executive Committees

Organizing Committee:	Chris Stephens (UNAM) Darrell Whitley (Colorado State University) Peter Stadler (University of Leipzig) Marc Toussaint (University of Edinburgh)
Local Organizing Committee:	Chris Stephens (UNAM) Katya Rodriguez (UNAM)
Program Committee:	Chris Stephens (UNAM) Darrell Whitley (Colorado State University) Peter Stadler (University of Leipzig) Marc Toussaint (University of Edinburgh)

Referees

J.E. Rowe
W.B. Langdon
A. Prügel-Bennett
C. Witt
R. Drechsler
J. Branke
W.E. Hart
C. Igel
L.M. Schmitt
B. Mitavskiy
I. Wegener
A. Eremeev
R. Heckendorn
A.H. Wright
H.-G. Beyer
M. Pelikan
Y. Gao
M. Gallagher
J. Shapiro
J. He
S. Droste
A. Bucci
M. Schoenauer
J. Smith
T. Jansen
R. Poli
W. Gutjahr
A. Auger
P. Stadler
D. Whitley
M. Vose
O. Teytaud

Sponsoring Institutions

ACM Special Interest Group on Genetic and Evolutionary Computation, SIGEVO.
Instituto de Ciencias Nucleares, UNAM.
Macroproyecto Universitario "Tecnologias para la Universidad de la Informacion y la Computacion," UNAM.
Posgrado en Ciencia y Ingenieria de la Computacion, UNAM.

Table of Contents

Inbreeding Properties of Geometric Crossover and Non-geometric Recombinations

Alberto Moraglio and Riccardo Poli

Department of Computer Science, University of Essex, UK
{amoragn,rpoli}@essex.ac.uk

Abstract. Geometric crossover is a representation-independent generalization of traditional crossover for binary strings. It is defined in a simple geometric way by using the distance associated with the search space. Many interesting recombination operators for the most frequently used representations are geometric crossovers under some suitable distance. Showing that a given recombination operator is a geometric crossover requires finding a distance for which offspring are in the metric segment between parents. However, proving that a recombination operator is not a geometric crossover requires excluding that one such distance exists. It is, therefore, very difficult to draw a clear-cut line between geometric crossovers and non-geometric crossovers. In this paper we develop some theoretical tools to solve this problem and we prove that some well-known operators are not geometric. Finally, we discuss the implications of these results.

1 Introduction

A fitness landscape [23] can be visualised as the plot of a function resembling a geographic landscape, when the problem representation is a real vector. When dealing with binary strings and other more complicated combinatorial objects, e.g., permutations, however, the fitness landscape is better represented as a height function over the nodes of a simple graph [19], where nodes represent locations (solutions), and edges represent the relation of direct neighbourhood between solutions.

An abstraction of the notion of landscape encompassing all the previous cases is possible. The solution space is seen as a metric space and the landscape as a height function over the metric space [1]. A metric space is a set endowed with a notion of distance between elements fulfilling few axioms [3]. Specific spaces have specific distances that fulfil the metric axioms. The ordinary notion of distance associated with real vectors is the Euclidean distance, though there are other options, e.g., Minkowski distances. The distance associated to combinatorial objects is normally the length of the shortest path between two nodes in the associated neighbourhood graph [4]. For binary strings, this corresponds to the Hamming distance.

In general, there may be more than one neighbourhood graph associated to the same representation, simply because there can be more than one meaningful

C.R. Stephens et al. (Eds.): FOGA 2007, LNCS 4436, pp. 1–14, 2007.

notion of syntactic similarity applicable to that representation [10]. For example, in the case of permutations, the adjacent element swap distance and the block reversal distance are equally natural notions of distance. Different notions of similarity are possible because the same permutation (genotype) can be used to represent different types of solutions (phenotypes). For example, permutations can represent solutions of a problem where relative order is important. However, they can also be used to represent tours, where the adjacency relationship among elements is what matters [21].

The notion of fitness landscape is useful if the search operators employed are connected or matched with the landscape: the stronger the connection the more landscape properties mirror search properties. Therefore, the landscape can be seen as a function of the search operator employed [5]. Whereas mutation is intuitively associated with the neighbourhood structure of the search space, crossover stretches the notion of landscape leading to search spaces defined over complicated topological structures [5].

Geometric crossover and geometric mutation [9] are representation-independent search operators that generalise by abstraction many pre-existing search operators for the main representations used in EAs, such as binary strings, real vectors, permutations and syntactic trees. They are defined in geometric terms using the notions of line segment and ball. These notions and the corresponding genetic operators are well-defined once a notion of distance in the search space is defined. This way of defining search operators as function of the search space is the opposite to the standard approach in which the search space is seen as a function of the search operators employed. Our new point of view greatly simplifies the relationship between search operators and fitness landscape and allows different search operators to share the same search space.

The reminder of this paper is organized as follows. In section 2, we introduce the geometric framework. In section 3, we show that the definition of geometric crossover can be cast in two equivalent, but conceptually very different, forms: functional and existential. When proving geometricity the existential form is the relevant one. We use this form also to show why proving non-geometricity of an operator looks impossible. In section 4, we develop some general tools to prove non-geometricity of recombination operators. In section 5, we prove that three recombination operators for vectors of reals, permutations and syntactic trees representations are not geometric. Importantly this implies that there are two *non-empty* representation-independent classes of recombination operators: geometric crossovers and non-geometric crossovers. In section 6, we draw some conclusions and present future work.

2 Geometric Framework

2.1 Geometric Preliminaries

In the following we give necessary preliminary geometric definitions and extend those introduced in [9]. For more details on these definitions see [4].

The terms *distance* and *metric* denote any real valued function that conforms to the axioms of identity, symmetry and triangular inequality. A simple connected graph is naturally associated to a metric space via its *path metric*: the distance between two nodes in the graph is the length of a shortest path between the nodes. Distances arising from graphs via their path metric are called *graphic distances*. Similarly, an edge-weighted graph with strictly positive weights is naturally associated to a metric space via a *weighted path metric*.

In a metric space (S, d) a *closed ball* is a set of the form $B_d(x; r) = \{y \in S | d(x, y) \leq r\}$ where $x \in S$ and r is a positive real number called the radius of the ball. A *line segment* (or closed interval) is a set of the form $[x; y]_d = \{z \in S | d(x, z) + d(z, y) = d(x, y)\}$ where $x, y \in S$ are called extremes of the segment. Metric ball and metric segment generalize the familiar notions of ball and segment in the Euclidean space to any metric space through distance redefinition. These generalized objects look quite different under different metrics. Notice that the notions of metric segment and shortest path connecting its extremes (*geodesic*) do not coincide as it happens in the specific case of an Euclidean space. In general, there may be more than one geodesic connecting two extremes; the metric segment is the union of all geodesics.

We assign a structure to the solution set S by endowing it with a notion of distance d. $M = (S, d)$ is therefore a solution *space* (or search space) and $L = (M, g)$ is the corresponding *fitness landscape* where $g : S \to \mathbb{R}$ is the fitness function. Notice that in principle d could be arbitrary and need not have any particular connection or affinity with the search problem at hand.

2.2 Geometric Crossover Definition

The following definitions are *representation-independent* and, therefore, crossover is well-defined for any representation. Being based on the notion of metric segment, *crossover is only function of the metric* d associated with the search space.

A recombination operator OP takes parents p_1, p_2 and produces one offspring c according to a given conditional probability distribution:

$$Pr\{OP(p_1, p_2) = c\} = Pr\{OP = c | P_1 = p_1, P_2 = p_2\} = f_{OP}(c | p_1, p_2)$$

Definition 1 (*Image set*). *The* image set $Im[OP(p_1, p_2)]$ *of a genetic operator* OP *is the set of all possible offspring produced by* OP *with non-zero probability when parents are* p_1 *and* p_2.

Definition 2 (*Geometric crossover*). *A recombination operator* CX *is a geometric crossover under the metric* d *if all offspring are in the segment between its parents:* $\forall p_1, p_2 \in S : Im[CX(p_1, p_2)] \subseteq [p_1, p_2]_d$

Definition 3 (*Uniform geometric crossover*). *The uniform geometric crossover* UX *under* d *is a geometric crossover under* d *where all* z *laying between parents* x *and* y *have the same probability of being the offspring:*

$$\forall x, y \in S : f_{UX}(z|x, y) = \frac{\delta(z \in [x; y]_d)}{|[x; y]_d|}$$

$$Im[UX(x, y)] = \{z \in S | f_{UX}(z|x, y) > 0\} = [x; y]_d$$

where δ is a function that returns 1 if the argument is true, 0 otherwise.

A number of general properties for geometric crossover and mutation have been derived in [9].

2.3 Notable Geometric Crossovers

For vectors of reals, various types of blend or line crossovers, box recombinations, and discrete recombinations are geometric crossovers [9]. For binary and multary strings (fixed-length strings based on a n symbols alphabet), all mask-based crossovers (one point, two points, n-points, uniform) are geometric crossovers [9,13]. For permutations, PMX, Cycle crossover, merge crossover and others are geometric crossovers [10,11]. For Syntactic trees, the family of Homologous crossovers (one-point, uniform crossover) are geometric crossovers [12]. Recombinations for other more complicated representations such as variable length sequences, graphs, permutations with repetitions, circular permutations, sets, multisets partitions are geometric crossovers [15,9,10,14].

2.4 Geometric Crossover Landscape

Since our geometric operators are representation-independent, one might wonder as to the usefulness of the notion of geometricity and geometric crossovers in practical applications. To see this, it is important to understand the difference between problem and landscape.

Geometric operators are defined as functions of the distance associated to the search space. However, the search space does not come with the problem itself. The problem consists only of a fitness function to optimize, that defines what a solution is and how to evaluate it, but it does not give any structure over the solution set. The act of putting a structure over the solution set is part of the search algorithm design and it is a designer's choice. A fitness landscape is the fitness function plus a structure over the solution space. So, for each problem, there is one fitness function but as many fitness landscapes as the number of possible different structures over the solution set. In principle, the designer could choose the structure to assign to the solution set completely independently from the problem at hand. However, because the search operators are defined over such a structure, doing so would make them decoupled from the problem, hence turning the search into something very close to random search.

In order to avoid this one can exploit problem knowledge in the search. This can be achieved by carefully designing the connectivity structure of the fitness landscape. That is, the landscape can be seen as a knowledge interface between algorithm and problem [10]. In [10] we discussed three heuristics to design the connectivity of the landscape in such a way to aid the evolutionary search performed by geometric crossover. These are: i) pick a crossover associated to a good mutation, ii) build a crossover using a neighbourhood structure based on the small-move/small-fitness-change principle, and iii) build a crossover using a distance that is relevant for the solution interpretation.

Once the connectivity of the landscape is correctly designed, problem knowledge can be exploited by search operators to perform better than random search, even if the search operators are problem-independent (as in the case of geometric crossover and mutation). Indeed, by using these heuristics, we have *designed* very effective geometric crossovers for N-queens problem [11], TSP [11] [10], Job Shop Scheduling [11], Protein Motifs discovery [20], Graph Partitioning [6], Sudoku [16] and Finite State Machines [7].

3 Interpretations of the Definition of Geometric Crossover

In section 2, we have defined geometric crossover as function of the distance d of the search space. In this section we take a closer look at the meaning of this definition *when the distance d is not known*. We identify three fundamentally different interpretations of the definition of geometric crossover. Interestingly it will become evident that there is an inherent element of self-reference in the definition. We show that proving that a recombination operator is non-geometric may be impossible.

3.1 Functional Interpretation

Geometric crossover is function of a *generic distance*. If one considers a specific distance one can obtain a specific geometric crossover for that distance by functional application of the definition of geometric crossover to this distance. This approach is particularly useful when the specific distance is firmly rooted in a solution representation (e.g., edit distances). In this case, in fact, the specification of the definition of geometric crossover to the distance acts as a formal recipe that indicates how to manipulate the syntax of the representation to produce offspring from parents. This is a general and powerful way to get new geometric crossover for any type of solution representation. For example, given the Hamming distance on binary string by functional application of the definition of geometric crossover we obtain the family of mask-based crossover for binary strings. In particular, by functional application of the definition of uniform geometric crossover one obtains the traditional uniform crossover for binary strings.

3.2 Abstract Interpretation

The second use of the definition of geometric crossover does not require to specify any distance. In fact we do apply the definition of geometric crossover to a generic distance. Since the distance is a metric that is a mathematical object defined axiomatically, the definition of geometric crossover becomes an axiomatic object as well. This way of looking at the definition of geometric crossover is particularly useful when one is interested in deriving general theoretical results that hold for geometric crossover under any specific metric. We will use this abstract interpretation in section 4 to prove the inbreeding properties that are common to all geometric crossovers.

3.3 Existential Interpretation

The third way of looking at the definition of geometric crossover becomes apparent when the distance d is not known and we want to find it. This happens when we want to know whether a recombination operator RX, defined operationally as some syntactic manipulation on a specific representation, is a geometric crossover and for what distance. This question hides an element of self-reference of the definition of geometric crossover. In fact what we are actually asking is: given that *the geometric crossover is defined over the metric space it induces by manipulating the candidate solutions*, what is such a metric space for RX if any?

The self-reference arises from the fact that the definition of geometric crossover applies at two distinct levels at the same time: (a) at a representation level, as a manipulation of candidate solutions, and (b) at a geometric level, on the underlying metric space based on a geometric relation between points. This highlights the inherent *duality* between these two worlds: they are based on the *same* search space seen from opposite viewpoints, from the representation side and from the metric side.

Self-referential statements can lead to paradoxes. Since the relation between geometric crossover and search space is what ultimately gives it all its advantages, it is of fundamental importance to make sure that this relation sits on a firm ground. So, it is important to show that the definition of geometric crossover does not lead to any paradox. We show in the following that the element of self-reference can be removed and the definition of geometric crossover can be cast in existential terms making it paradox-free.

A non-functional definition of geometric crossover is the following: a recombination operator RX is a geometric crossover if the induced search space is a metric space on which RX can be defined as geometric crossover using the functional definition of geometric crossover. This is a self-referential definition. If a recombination operator does not induce any metric space on which it can be defined as geometric crossover, then it is a non-geometric crossover.

We can remove the element of self-reference from the previous definition and cast it in an existential form: a recombination RX is a geometric crossover if for any choice of the parents all the offspring are in the metric segment between them for some metric.

The existential definition is equivalent to the self-referential definition because if such a metric exists the operator RX can be defined as geometric crossover on such a space. On the other hand, if an operator is defined on a metric space as geometric crossover in a functional form, such a space exists by hypothesis and offspring are in the segment between parents under this metric by definition.

3.4 Geometric Crossover Classes

The functional definition of geometric crossover induces a natural existential classification of all recombination operators into two classes of operators:

- *geometric crossover class* $\mathcal{G}$: a recombination OP belongs to this class if there exists at least a distance d under which such a recombination is geometric: $OP \in \mathcal{G} \iff \exists d : \forall p_1, p_2 \in S : Im[OP(p_1, p_2)] \subseteq [p_1, p_2]_d$.
- *non-geometric crossover class* $\bar{\mathcal{G}}$: a recombination OP belongs to $\bar{\mathcal{G}}$ if there is no distance d under which such a recombination is geometric: $OP \in \bar{\mathcal{G}} \iff \forall d : \exists p_1, p_2 \in S : Im[OP(p_1, p_2)] \setminus [p_1, p_2]_d \neq \emptyset$.

For this classification to be meaningful we need these two classes to be non-empty. In previous work we proved that a number of recombination operators are geometric crossovers so $\mathcal{G}$ is not empty. What about $\bar{\mathcal{G}}$? To prove that this class is not empty we have to prove that at least one recombination operator is non-geometric. However, as we illustrate below this is not easy to do.

Let us first illustrate how one can prove that a recombination operator RX is in $\mathcal{G}$. We will use the self-referential definition of geometric crossover. The procedure is the following: guess a candidate distance d, then prove that all offspring of all possible pairs of parents are in the metric segment associated with d. If this is true then the recombination RX is geometric crossover under the distance d *because the operator RX can be defined as a geometric crossover on this space*. If the distribution of the offspring in the metric segments under d is uniform, RX is the uniform geometric crossover for the metric d *because the operator RX can be defined as the (unique) geometric uniform crossover on this space*. If one finds that some offspring are not in the metric segment between parents under the initially guessed distance d then the operator RX cannot be defined as geometric crossover over this space. However, this does not imply $RX \in \bar{\mathcal{G}}$ because there may exist another metric d' that fits RX and *makes it definable* as a geometric crossover on d'. So, one has to guess a new candidate distance for RX and start all over again until a suitable distance is found.

Although we developed some heuristics for the selection of a candidate distance, in general proving that a recombination operator is geometric may be quite hard (see for example [12] where we considered homologous crossover for GP trees). Nonetheless, the approach works and, in previous work, we proved that a number of recombination operators for the most frequently used representations are geometric crossover under suitable distances.

It is evident, however, that the procedure just described cannot be used to prove that a given recombination operator RX is non-geometric. This is because we would need to test and exclude all possible distances, which are infinitely many, before being certain that RX is not geometric. Clearly, this is not possible.

In the next section we build some theoretical tools based on the abstract interpretation of the definition of geometric crossover to prove non-geometricity in a more straightforward way.