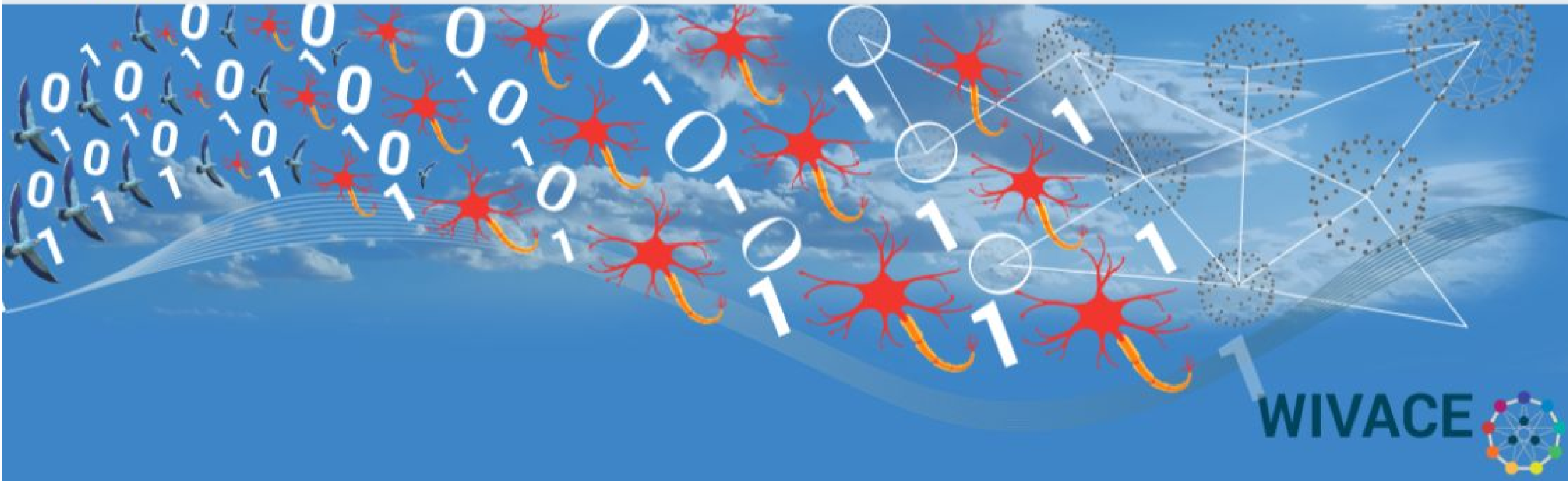
European Centre for Living Technology



WIVACE 2017

Book of Abstracts

Venice, 19-21 September 2017

For info please visit: http://wivace.org/2017

Università Ca'Foscari Venezia
Dipartimento di Scienze Ambientali, Informatica e Statistica

Springer

Ca'Foscari Zattere
Cultural Flow Zone

Algebraic perspectives of solutions spaces in combinatorial optimization

Marco Baioletti[1], Alfredo Milani[1,2], and Valentino Santucci[1]

[1] Department of Mathematics and Computer Science, University of Perugia, Italy
[2] Department of Computer Science, Hong Kong Baptist University, Hong Kong
{marco.baioletti,alfredo.milani,valentino.santucci}@unipg.it

1 Introduction

Motivated from the algebraic evolutionary algorithms proposed in [4] and [1], here we introduce novel algebraic perspectives for the search space of a large class of combinatorial optimization problems.

By moving from some simple concepts of group theory, we propose a framework that allows: (i) to use algebraic concepts in order to formally define what is a search move on a discrete space of solutions, (ii) to provide a rationale of the algebraic concepts by means of simple geometric arguments, and (iii) to derive a formal languages point-of-view in order to link algebraic and geometric views, other than to extend the framework to more general search spaces.

The rest of the abstract is organized as follows. Section 2 describes some algebraic background. Abstract algebraic perspectives of combinatorial search spaces are introduced in Section 3, while concrete spaces are depicted in Section 4. Section 5 describes applications of the proposed framework, while conclusion are drawn in Section 6.

2 Algebraic Background

A group $(X, \circ)$ is an algebraic structure consisting of a set of elements X and a binary operation $\circ : X \times X \to X$ satisfying the properties of closure, associativity, existence of a unique neutral element e X, and existence of a unique inverse element x^{-1} for every $x \in X$. If the operation $\circ$ is also commutative, then the group is Abelian. The group $(X, \circ)$ is finitely presented if there exists a presentation (G, R) such that: (i) $G \subseteq X$ is a finite set of elements, called generators, such that every element of X can be expressed as the composition (under the group operation $\circ$) of finitely many elements of G, i.e., for any $x \in X$ it is possible to write $x = g_1 \circ \cdots \circ g_k$ for some $k \in \mathbb{N}_0$ and $g_i \in G$ for $1 \leq i \leq k$, while (ii) R is a finite set of equivalence relations (made using the group operation $\circ$) among the generators in G. Therefore, we can denote a finitely presented group as $(X, \circ, G, R)$ with the properties described.

Geometrically, every finitely presented group $(X, \circ, G, R)$ can be interpreted as an arc-colored digraph $C(X, \circ, G, R)$ such that: (i) the vertices are uniquely identified by the elements of X, (ii) the arc-colors are identified by the generators in G, (iii) for any $x \in X$ and $g \in G$, the arc $x \to (x \circ g)$ is colored by g, and (iv) any relation in R corresponds to n cycles in the graph (one for each vertex). This graph is called Cayley graph and has the properties of being regular (every vertex has the same degree), strongly connected (for every ordered pair of vertices there is a path connecting them) and vertex-transitive (informally, it is not possible to recognize a vertex by simply looking at the colors of its incoming and outgoing arcs).

Finally, a group can be observed also from the point of view of formal languages. Indeed, the presentation (G, R) of the group $(X, \circ)$ allows to interpret: (i) the generators in G as a set of symbols, (ii) the elements in X as strings over the alphabet G, (iii) the operation $\circ$ as a concatenation of strings, and (iv) the relations in R as rewriting rules for equivalent strings.

3 Combinatorial Search Spaces

Generally, meta-heuristic algorithms for combinatorial optimization iteratively improve a set of candidate solutions (possibly one, as in the case of local search based algorithms) by performing moves from one candidate solution to another. Therefore, the search moves implicitly define neighborhood relations among the candidate solutions in the search space. These moves can be classified in simple and composite moves. Simple moves are the ones usually adopted by the local search based algorithms (e.g., bit-flip moves), while composite moves can be considered as a composition of simple moves (e.g., "jumps" in the space such as those performed by genetic mutation or crossover operators).

Using the concepts described in Section 2, it is possible to define a relationship between a discrete search space and a finitely presented group $(X, \circ, G, R)$ where: (i) the group elements in X are exactly all the candidate solutions in the search space, and (ii) the generators in G, through the group operation $\circ$ and the relations in R, exactly identify the simple search moves. As a consequence, composite search moves can be represented by combining generators, i.e., as strings of generators. Now, since a string of generators is also equivalent to a group element, thus to a candidate solution, we can use the same representation to dichotomously identify both the candidate solutions and the search moves.

Geometrically, the Cayley graph of the group $(X, \circ, G, R)$ connects the candidate solutions by labeling the arcs with the simple search moves needed to reach a solution from another. As a consequence, composite moves can be seen as a path on the Cayley graph. This interpretation allows to naturally define a distance among the candidate solutions, that is, the shortest path distance on the Cayley graph. Moreover, given two solutions $x, y \in X$, all the paths from x to y are represented by sequences of arc-colors. Notably, all these color sequences, by the properties of $\circ$ and the relations in R, evaluate to the same element $y^{-1} \circ x$. This allows to uniquely define the difference between two candidate solutions as $x \ominus y := y^{-1} \circ x$, and, since among all the paths from x to y there are also the shortest paths from x to y, it appears that the group element $y^{-1} \circ x$ can be decomposed to a minimal string of generators with length exactly equal to the distance between x and y. Similarly, other "vectori" operations such as addition and multiplication by scalar can be consistently defined.

Practically, this framework makes possible to represent movements between combinatorial solutions using algebraic or, more generally, linguistic operations.

4 Concrete Search Spaces

Most of the search spaces induced by the representations typically used to handle combinatorial optimization problems can be described by using a finitely presented group $(X, \circ, G, R)$. The main examples are:

- n-length bitstrings where: X is the set of all strings of n bits, $\circ$ is the bitwise *xor* operator, and the generators in G are all the strings with exactly one 1-bit and $n-1$ 0-bit. Therefore, every generator, when composed with a generic bitstring $x \in X$, corresponds to flipping one bit of x. The induced distance is the classical Hamming distance.
- n-length integer vectors where: X is the set of all n-length integer vectors, i.e., $X = \mathbb{Z}^n$, $\circ$ is the integer vector addition, and the generators in G are all the vectors with $n-1$ components equal to 0 and one component equal to ± 1. The induced distance is the classical Manhattan distance.
- permutations of the set $[n] = \{1, \dots, n\}$ where: X is the set of all the permutations of $[n]$, $\circ$ is the usual permutation composition operator, and the generators in G are all the permutations that differ from the identity permutation for one pair of adjacent items that are swapped. Therefore, every generator, when composed with a generic permutation $x \in X$, corresponds to swapping adjacent items of x. In this case, the induced distance is the popular Kendall-τ distance. It worths to note that permutations admit also different generating sets like, for example, those representing items' insertion and interchange moves (see [2]).

In order to build composite move operators, two algorithms are needed: the implementation of the composition operator, and an algorithm that decompose a generic elements in a minimum sequence of generators. In the cases above, it is possible to derive these algorithms. Furthermore, the linguistic perspective also allows to build operative procedures for more general search spaces. The idea is to represent elements directly as strings of generators. Hence, there is no need for decomposers, and composition reduces to string concatenation followed by a minimal rewriting operation (Knuth-Bendix algorithm [3] solves this problem for any finitely presented group).

5 Applications to Evolutionary Computation

The algebraic framework previously discussed has three main applications:

- defining combinatorial variants of many continuous evolutionary algorithms (e.g., differential evolution [4] and particle swarm optimization [1]) for combinatorial search spaces by linking the continuous and combinatorial geometry of the algorithm moves;

– interpreting many existing combinatorial genetic operators using algebraic operators induced by the proposed framework;
– defining new combinatorial operators.

Furthermore, the proposed algebraic perspectives make possible to have a unified view of different combinatorial spaces. Hence, from a theoretical and computational point-of-view, concrete spaces can be studied by analyzing the algebraic properties induced by their finite presentation.

6 Conclusion

Summarizing, this abstract sketched three perspectives of combinatorial search spaces from the point-of-views of algebra, geometry and formal languages. This interpretation applies to a variety of practical search spaces. Using well understood algebraic concepts in literature (such as product groups and rewriting systems) it is also possible to define finitely presented groups for more complex spaces. The main result is the single and dichotomous representation for both candidate solutions and search moves. This builds an analogy with continuous search spaces (numerical vectors in $\mathbb{R}^n$), thus providing, among the others, a mean to generalize meta-heuristic algorithms for continuous optimization to combinatorial problems. As future research avenues, we think that these algebraic perspectives also allow to derive some theoretical analysis of search algorithms for combinatorial spaces.

References

1. Baioletti, M., Milani, A., Santucci, V.: Algebraic particle swarm optimization for the permutations search space. In: Proc. of IEEE Congress on Evolutionary Computation CEC 2017 (in press)
2. Baioletti, M., Milani, A., Santucci, V.: An extension of algebraic differential evolution for the linear ordering problem with cumulative costs. In: Parallel Problem Solving from Nature - PPSN XIV - 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings. pp. 123–133 (2016), http://dx.doi.org/10.1007/978-3-319-45823-6_12
3. Knuth, D.E.: The genesis of attribute grammars. In: Attribute Grammars and Their Applications, pp. 1–12. Springer (1990)
4. Santucci, V., Baioletti, M., Milani, A.: Algebraic differential evolution algorithm for the permutation flowshop scheduling problem with total flowtime criterion. IEEE Transactions on Evolutionary Computation 20(5), 682–694 (2016), http://dx.doi.org/10.1109/TEVC.2015.2507785