Generating Fullerenes at Random

Bor Plestenjak and Tomaž Pisanski
IMFM/TCS, University of Ljubljana,
Jadranska c. 19, SI-61111 Ljubljana, Slovenia

and

Ante Graovac
The R. Bošković Institute, HR-41001 Zagreb, POB 1016, Croatia.

Received:

Abstract

In the present paper a method for generating fullerenes at random is presented. It is based on the well known Stone-Wales (SW) transformation. The method could be further generalised so that other trivalent polyhedra with prescribed properties are generated.

1. INTRODUCTION

Fullerenes and other pure carbon cages remain a subject of rigorous research. The mechanism of fullerenes growth is still not fully understood although much has been learned[1]. In the present paper a method for generating fullerenes at random is presented. It is based on the well known Stone-Wales (SW) transformation[2, 3] and it has been successfully implemented as a part of the VEGA: a system for manipulating discrete mathematical structures.[4, 5] The method could be further generalised so that it is able to narrow the population of carbon cages with special properties.

From a mathematical standpoint a fullerene is planar trivalent graph whose faces are pentagons and hexagons. It turns out that the number of pentagons is always equal to 12. Let n denote the number of vertices, m the number of edges and r the number of faces. Furthermore, let r_5 denotes the number of pentagons and h the number of hexagons in a fullerene. Then the following is true:

$$\begin{aligned} n - m + r &= 2 && \text{(Euler polyhedral formula for the sphere)} \\ r_5 + h &= r && \text{(each face is either pentagon or hexagon)} \\ 2m &= 3n && \text{(graph is trivalent)} \\ 2m &= 5r_5 + 6h && \text{(each edge is shared between two faces)} \end{aligned} \tag{1}$$

This linear system of equations readily gives:

$$r_5 = 12. \tag{2}$$

In addition, for a fullerene with h hexagons the number of faces, vertices and edges is given by:

$$\begin{aligned} r &= 12 + h \\ n &= 20 + 2h \\ m &= 30 + 3h. \end{aligned} \tag{3}$$

From the above it follows that n is an even number; $n \geq 20$. It can be shown that fullerenes exist for any even $n \geq 20$ with exception of $n = 22$. [6]

Usually a fullerene with n vertices is denoted by C_n. The smallest fullerene is dodecahedral C_{20} (it is unique and contains pentagons only).

C_{20} and C_{24} fullerenes are depicted in Figure 1 as the generalised Petersen graphs $GP(10, 2)$ and $GP(12, 2)$, respectively.

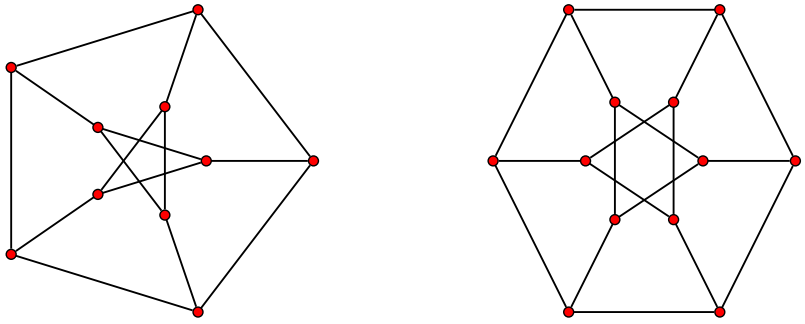

Figure 1. *C_{20} and C_{24} fullerenes as generalised Petersen graphs.*

In general, the number of non-isomorphic isomers of C_n fullerenes, $c(n)$, grows rapidly with n. For instance, $c(30) = 3$, $c(40) = 60$, $c(60) = 1812$.

The isomers can be further divided in several classes according to the characteristics p and q, where p is the number of edges shared by two pentagons and q is the number of vertices shared by three pentagons. Obviously, even the class with the same characteristics p and q contains non-isomorphic fullerenes. The smallest such case can be found for $n = 32$ where two non-isomorphic fullerenes with $p = 18$ and $q = 8$ exist. From a chemical viewpoint the most interesting fullerenes are those with $p = q = 0$. They are so called isolated pentagons (IP) fullerenes. The smallest and most attractive case is renowned buckminsterfullerene, the icosahedral C_{60}.

2. METHOD

The basic idea of *the polyhedral Stone-Wales (PSW) transformation* applies to an edge e of a polyhedron P giving a new polyhedron P':

$$P' = PSW(P, e). \tag{4}$$

The transformation is depicted in Figure 2. It is obvious that the number of vertices and the edges does not change at this transformation: $n(P) = n(P')$, $m(P) = m(P')$. As one can see, the size of two faces A and B having e as a common edge in P is decreased by 1 in P', while the size of two faces, C and D, is increased by 1 upon transformation. Other faces of the polyhedron

remain unchanged. Let f_F denote the size of face F. Then we have:

$$\begin{aligned} f_{A'} &= f_A - 1 \\ f_{B'} &= f_B - 1 \\ f_{C'} &= f_C + 1 \\ f_{D'} &= f_D + 1 \end{aligned} \tag{5}$$

In order to have no loops or parallel edges in the dual graph of P', faces C and D should not coincide or have an edge in common and sizes of faces A and B should be greater or equal 4. Two planar trivalent polyhedra P_1 and P_2 are said to be *equivalent* (under PSW-transformations) if they can be transformed into each other, up to homeomorphism, by a finite sequence of PSW-transformations.

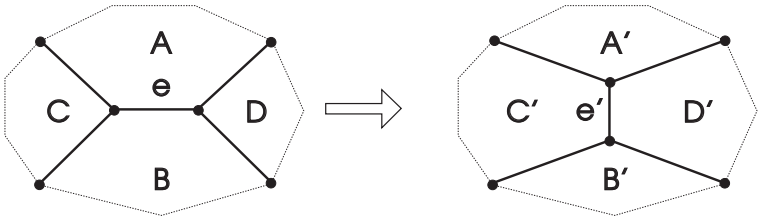

Figure 2. *The polyhedral Stone-Wales transformation.*

A dual of a trivalent planar polyhedron is a triangulation of a sphere. In paper [7] Negami defined the operation *diagonal flip* on triangulations, depicted on Figure 3.

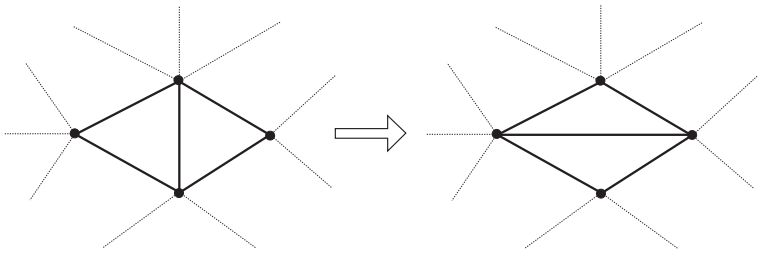

Figure 3. *The diagonal flip transformation.*

Theorem 1 *Two trivalent cubic polyhedra P_1 and P_2 are equivalent under*

PSW-transformations if $r(P_1) = r(P_2) \geq 4$.

Proof. This theorem follows from Theorem 1 in [7]. In fact, our theorem is an interpretation of Theorem 1 in a dual form. It is easy to see that the diagonal flip is a dual transformation to the PSW-transformation (Figure 4) and in the case of the sphere the constant N in [7] is $N = 4$.

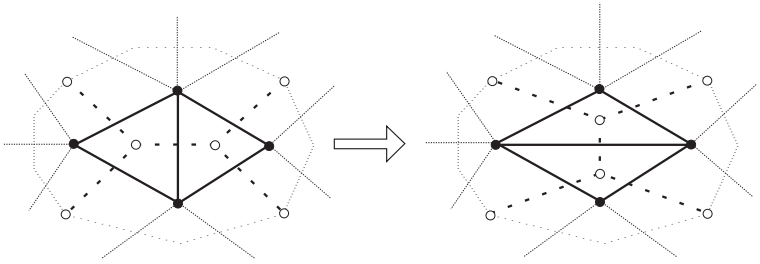

Figure 4. *The diagonal flip transformation is a dual transformation to the PSW-transformation.*

The following corollary follows from Theorem 1.

Corollary 1 *Two fullerenes* F_1 *and* F_2 *are equivalent under PSW-transformations if* $n(F_1) = n(F_2)$.

In order to generate fullerenes at random one can proceed as follows. Start with an arbitrary, planar, trivalent, connected graph on n vertices which has $12 + h$ faces and then apply repeatedly a number of suitable PSW-transformations, until eventually a fullerene is obtained. Theorem 1 guarantees that a fullerene can be generated by a finite sequence of PSW-transformations.

If we want to generate a fullerene with n vertices, m edges and r faces we have to start with a planar polyhedral graph having these parameters. For a fullerene with h hexagons, one can start the process with the $(10 + h)$-gonal prism, which is the Cartesian product of K_2 and the cycle on $10 + h$ vertices; see Figure 5. At each step of the process, one has to choose an edge e on which the PSW-transformation is performed. Our method uses an energy function which helps choosing the best edge.

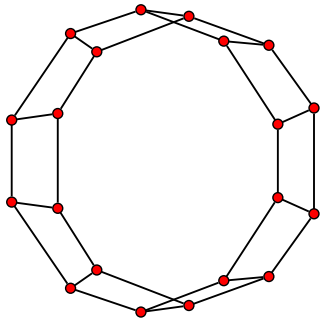

Figure 5. *The Cartesian product of K_2 and the cycle on 10 vertices.*

Let f_i denote the size of the i-th face ($1 \leq i \leq 12+h$) of P and $\overline{f}$ equals:

$$5 \leq \overline{f} = 5 + \frac{h}{12+h} \leq 6. \tag{6}$$

Note that $\overline{f}$ is the average face size in any fullerene and $\overline{f} = 5$ if and only if P is dodecahedron and $\overline{f} = 6$ only for an infinite graphite lattice.

Now define the energy function $E(P)$ of a polyhedron P as:

$$E(P) = \sum_{i=1}^{12+h} |f_i - \overline{f}| - \frac{24h}{12+h}. \tag{7}$$

The term $24h/(12+h) = 12(\overline{f}-5) + h(6-\overline{f})$ is subtracted in order to guarantee the following:

$$\begin{array}{lll} 1. & E(P) \geq 0, & \\ 2. & E(P) = 0 & \text{if and only if } P \text{ is a fullerene.} \end{array} \tag{8}$$

In order to see whether the PSW transformation increases or decreases the energy of P, one has to consider only the energy difference $\Delta_e E(P)$:

$$\begin{array}{rcl} \Delta_e E(P) & = & E(P') - E(P) = E(PSW(P,e)) - E(P) = \\ = E(P) & + & |f_{A'} - \overline{f}| - |f_A - \overline{f}| + |f_{B'} - \overline{f}| - |f_B - \overline{f}| + \\ & + & |f_{C'} - \overline{f}| + |f_C - \overline{f}| + |f_{D'} - \overline{f}| + |f_D - \overline{f}| = \\ = E(P) & + & |f_A - 1 - \overline{f}| - |f_A - \overline{f}| + |f_B - 1 - \overline{f}| - |f_B - \overline{f}| + \end{array}$$

$$+ \quad |f_C + 1 - \overline{f}| + |f_C - \overline{f}| + |f_D + 1 - \overline{f}| + |f_D - \overline{f}| \qquad (9)$$

which contains constant (independent of h) number of terms.

We say that the edge e is "better" than the edge e' if $\Delta_e E(P) < \Delta_{e'} E(P)$. We say that some edge e is "best" if $\Delta_e E(P) < \Delta_{e'} E(P)$ for any edge e'.

The following selection rule is applied: if $\Delta_e E(P) < 0$ for any best edge e than e is selected. Otherwise, an edge among the first ν edges with the lowest value of $\Delta_e E(P)$ is selected with equal probability. Here ν is an experimentally chosen constant. It turns out that $\nu = 3$ is appropriate for small fullerenes and $\nu = 5$ for large ($n \geq 80$) fullerenes.

If we want to generate fullerenes with specified p and q parameters the following function is selected:

$$E(P) = \sum_{i=1}^{12+h} |f_i - \overline{f}| - \frac{24h}{12+h} + |p - \overline{p}| + |q - \overline{q}| \qquad (10)$$

where p and q are the characteristics of the current polyhedron P and $\overline{p}$ and $\overline{q}$ are the target values. This energy function is used for instance for generating IP fullerenes.

One step of the algorithm has time complexity $\mathcal{O}(n)$. We have to update only energy differences $\Delta_e(P)$ for all edges e in faces influenced by PSW (A', B', C', D' from Figure 2). Other values $\Delta_e(P)$ remain unchanged and we can found the first ν edges with the lowest value of $\Delta_e E(P)$ in $\mathcal{O}(n)$ time complexity.

The energy function was generalised in order to generate arbitrary trivalent polyhedra. Let $\overline{r_i}$ denotes the target number of faces of size i and r_i the current value of faces of size i. Now, the energy function $E(P)$ is defined as:

$$E(P) = \sum_{i=3}^{\infty} |r_i - \overline{r_i}|. \qquad (11)$$

The selection will remain the same as before. However, the computation process is slower since more book keeping is needed. Due to the different energy function (11) energy differences $\Delta_e(P)$ for all edges e have to be updated in each step of the algorithm. Time complexity remains $\mathcal{O}(n)$.

Using this generalised algorithm cubic graphs imbedded in other oriented surfaces can be generated using different starting graphs.

We have experimented with further generalisation of the energy function where the parameters p_s and q_s are added and they refer to faces of arbitrary size s (not necessarily pentagons):

$$E(P) = \sum_{i=3}^{\infty} |r_i - \overline{r_i}| + |p_s - \overline{p_s}| + |q_s - \overline{q_s}|. \tag{12}$$

3. NUMERICAL EXPERIMENTS

Now we are ready to present the results of numerical experiments.

In Table 1 the minimum and the average number of iterations together with the average computational time necessary for generating a fullerene on n vertices, $n = 20, 30, \ldots, 110$ and using the energy function (9) is given. For each n one hundred random fullerenes have been generated. So, the average number of iterations and the average time of generation was calculated using these 100 experiments. The computations have been performed on 66 MHz 486/DX2 PC. Figures 6-9 present the same data fitted with least square polynomials.

Table 1. *The average number of iterations, the minimum number of iterations and the average computational time for generating a fullerene on* $n = 20, 30, \ldots, 110$ *vertices calculated using fifty experiments.*

n	aver. number of iterations	min. number of iterations	aver. comput. time in seconds	aver. comput. time per one iteration in 1/1000 seconds
20	13.09	11	0.022	1.70
30	26.00	21	0.076	3.60
40	39.35	31	0.158	4.00
50	57.49	43	0.288	5.01
60	76.79	58	0.453	5.89
70	122.29	76	0.826	6.75
80	162.38	91	1.225	7.54
90	229.67	112	1.933	8.42
100	265.84	126	2.470	9.29
110	324.59	155	3.270	10.07

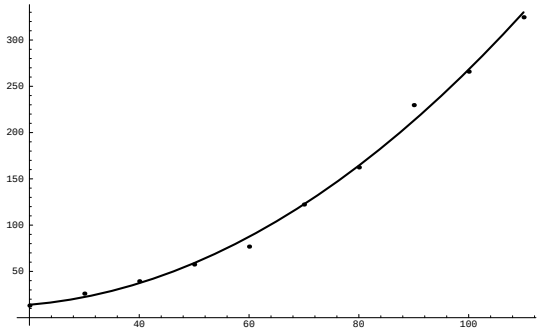

Figure 6. *The average number of iterations for generating a fullerene on* $n = 20, 30, \ldots, 110$ *fitted with the least square parabola* $y = 17.45 - 0.8454x + 0.03352x^2$.

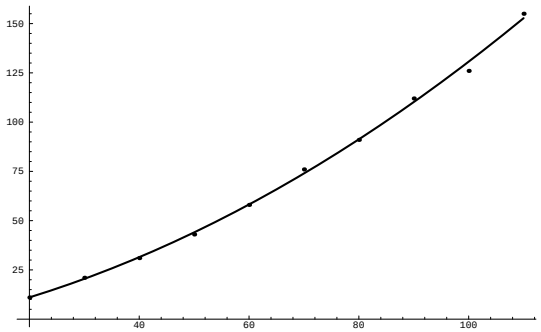

Figure 7. *The minimum number of iterations for generating a fullerene on* $n = 20, 30, \ldots, 110$ *fitted with the least square parabola* $y = -3.028 + 0.5454x + 0.007917x^2$.

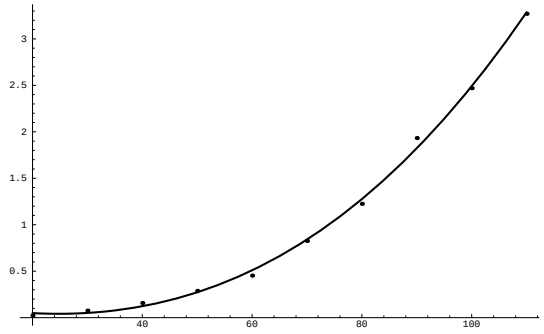

Figure 8. *The average computational time for generating a fullerene on* $n = 20, 30, \ldots, 110$ *fitted with the least square cubic polynomial* $y = 0.2204 - 0.01358x + 0.0002222x^2 + 1.40610^{-6}x^3$.

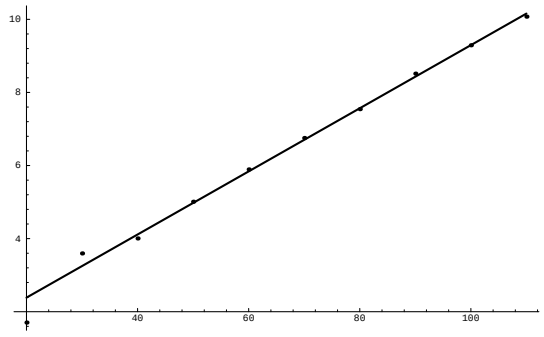

Figure 9. *The average computational time per one iteration for generating a fullerene on* $n = 20, 30, \ldots, 110$ *fitted with the least square linear polynomial* $y = 0.6579 + 0.08636x$.

4. CONCLUSIONS

A variety of algorithms to generate fullerenes is offered until now, some of them being reliable and efficient. [8, 9] Here, a new algorithm is presented which uses the polyhedron Stone-Wales transformation and random local search for local minima of adequately chosen energy function. Moreover, by suitable modifications of energy function we have shown how to generate other trivalent polyhedra with prescribed properties such as face structure, avoidance of some faces to be adjacent, etc.

The method raises a series of questions concerning the distribution of randomly generated fullerene isomers with the respect to the choice of initial polyhedron. Further open question deals with whether one can reproduce realistic distribution of a given property from the subset of randomly generated isomers.

ACKNOWLEDGEMENTS

The authors thank Prof. Patrick Fowler for helpful discussions. This work was supported by the Ministry of Science and Technology of Slovenia (Grant No. P1-0206-101-92) and the Ministry of Science and Technology of Croatia (Grant No. 1-07-159). One of the authors (A.G.) also acknowledges the support of the Open Society-Croatia Institute.

References

(1) Amelinckx, S.; Zhang, X. B.; Bernaerts D.; Zhang, X. F.; Ivanov, V.; Nagy, J. B. A Formation Mechanism for Cataclytically Grown Helix-Shaped Graphite Nanotubes. *Science* **1994**, 265, 635-639.

(2) Manolopoulos, D. E.; Fowler, P. W.; Ryan, R. P. Hypothetical Isomerations of LaC_{82}. *J. Chem. Soc. Faraday Trans.* **1992**, 88, 1225-1226.

(3) Fowler, P. W.; Manolopoulos, D. E.; Orlandi, G.; Zerbetto, F. Energetics and Isomerization Pathways of a Lower Fullerene. The Stone-Wales Map for C_{40}. *J. Chem. Soc. Faraday Trans.* **1995**, 91, 1421-1423.

(4) Pisanski, T.; *et al.* Vega: System for manipulating discrete mathematical structures. *Available via anonymous ftp from* ftp.mat.uni-lj.si.

(5) Pisanski, T.; Plestenjak, B.; Graovac, A. NiceGraph Program and its Applications in Chemistry. *Croat. Chem. Acta* **1995**, 68, 283-292.

(6) Pisanski, T. On Planar Graphs with 12 Vertices of Degree 5. *Glasnik matematički (Zagreb)* **1977**, 12, 233-235.

(7) Negami, S. Diagonal Flips in Triangulations of Surfaces, *Disc. Math.* **1994**, 135, 225-232.

(8) Brinkmann, G.; Dress, A. W. M. PentHex Puzzles. A Reliable and Efficient Top-Down Approach to Fullerene-Structure Enumeration. *Proceedings of the Academy of Sciences of America*. In press.

(9) Brinkmann, G.; Dress, A. W. M. A Constructive Enumeration of Fullerenes. *Journal of Algorithms*. Submitted.