MŰEGYETEM 1782

RESTRICTED GENERATION OF QUADRANGULATIONS AND SCHEDULING PARAMETER SWEEP APPLICATIONS

RICHÁRD KÁPOLNAI

Thesis Booklet

Advisor: Dr. Imre Szeberényi

21 October 2014

Department of Control Engineering and Information Technology
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

ACKNOWLEDGEMENTS

I would like to express my gratitude to my external advisor, Gábor Domokos for his support during my research.

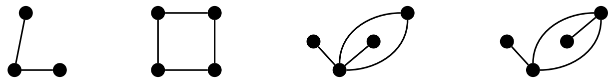
1 GENERATING QUADRANGULATIONS

1.1 RESEARCH GOAL

Recently, Várkonyi and Domokos [6] introduced a mechanical classification system for convex, homogeneous bodies. They map each body into its *primary equilibrium class* (or shortly *primary class*) defined by the numbers of the stable and unstable equilibrium points of the body surface. Informally, an *equilibrium* is a surface point on which the body remains at rest on a horizontal plane. It is stable, if the body returns to its position despite a small perturbation from any direction, and it is unstable, if the body never returns. For instance, the cube has six stable and eight unstable equilibria, corresponding to the faces and the vertices, respectively. The edges correspond to saddle equilibria which are not considered for now.

Várkonyi and Domokos [6] constructed the geometry of a mono-monostatic body, also known as *Gömböc*, which has only one stable and one unstable equilibrium. They also designed specific geometric transformations called *Columbus' algorithm*, modifying the body only at the vicinity of one equilibrium and increasing the number of equilibrium points. Starting from the Gömböc, Columbus' algorithm is able to generate a representative body in any primary class, so Gömböc is the ancestor of every primary class.

If, beyond the numbers of the equilibria, their full topology defined by the Morse–Smale complex of the body surface is considered, we arrive at the more refined *secondary equilibrium classes*



(a) P_2 (b) C_4 (c) Q_3 (d) Q_4

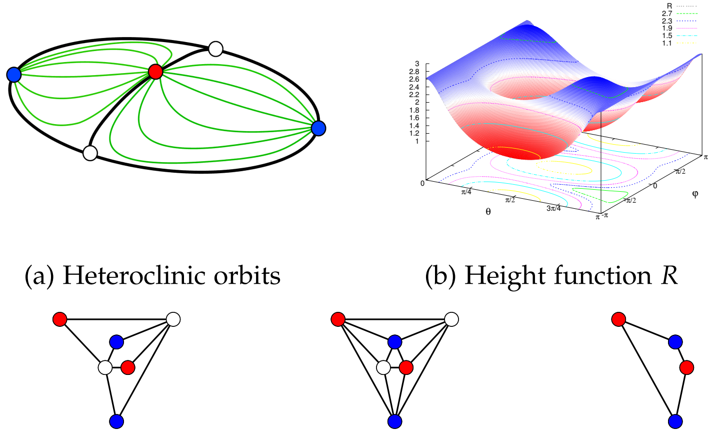
Figure 1.1. The four smallest quadrangulations

(or shortly *secondary classes*). While Columbus' algorithm proved that every primary class contains some bodies [6], Domokos, Lángi and Szabó [1] showed recently that every secondary class contains some bodies as well, because their extension of Columbus' algorithm can generate all of them.

A *multiquadrangulation* (or shortly *quadrangulation*) is a graph embedded in the sphere with every face bounded by a closed walk of length 4. All quadrangulations with at most 4 vertices are shown in Figure 1.1. A secondary class, i.e. the topology of the equilibria can be genuinely represented by a vertex-coloured quadrangulation [2, 4], making the extended Columbus' algorithm a graph operation called *coloured splitting*. This enables us to study the hierarchy of secondary classes in a purely combinatorial context. Figure 1.2 illustrates some intermediate steps of the derivation process of the surface graph of a convex, homogeneous body.

In the dissertation, we consider a carefully chosen restriction of the coloured splitting called *monotone coloured splitting*. The monotone coloured splitting possesses a significant combinatorial property, and their corresponding geometric transformations also possess an unusual geometric property.

The first research goal is to find out which secondary classes can be generated from others by the monotone coloured splitting.



(a) Heteroclinic orbits (b) Height function R

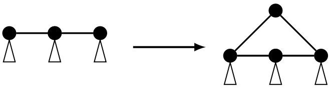
(c) Morse–Smale graph (d) Eq. triangulation (e) Quasi-dual

Figure 1.2. Secondary class of the ellipsoid (e). Red: stable, blue: unstable. Figure 1.2a is created by Tímea Szabó [A2].

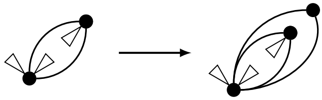
1.2 RESEARCH METHOD

All cases of monotone splitting is shown in Figure 1.3, where the *degree* of the splitting is roughly the degree of the introduced vertex. The *coloured monotone splitting* consists of a monotone splitting and the proper colouring of the introduced vertex. So the statements made on quadrangulations and monotone splittings have a direct geometric interpretation in the context of mechanical equilibria of convex bodies.

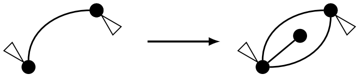
A secondary class is an *irreducible ancestor*, if it cannot be constructed by the monotone coloured splitting. While monotone coloured splitting generates all primary classes from one single ancestor, it can only generate a limited range of secondary classes from the same ancestor, admitting additional ancestors besides the Gömböc. This uncovers a complex hierarchy of secondary classes.



(a) simple splitting of degree 2

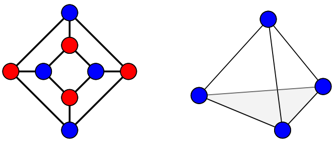


(b) parallel spl. of degree 2

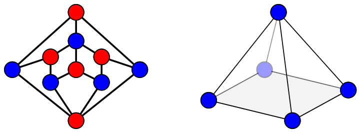


(c) splitting of degree 1

Figure 1.3. Variants of monotone splitting.



(a) Tetrahedron



(b) Square pyramid

Figure 1.4. Secondary class and skeleton of two minimal polyhedra

It turns out that the minimal polyhedra play a key role in this hierarchy. A polyhedron is a *minimal polyhedron* if every face contains one stable and each of its vertex is an unstable equilibrium. The two smallest minimal polyhedra are shown in Figure 1.4. It is easy to see that they are irreducible because the monotone splitting introduces a vertex of degree 1 or 2, however, the minimum degree of their secondary class is 3.

1.3 RESULTS

The results are summarized below, and the discovered hierarchy is visualized in Figure 1.5.

MAIN RESULT 1 (Published in: [A2] [C4, C5]): I showed that in the family of the secondary classes, the monotone coloured splitting admits other irreducible ancestors than the Gömböc. I also

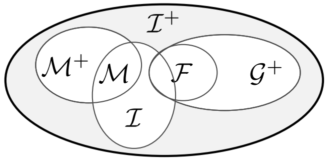


Figure 1.5. Hierarchy of secondary classes. Notation: $\mathcal{I}^+$: all secondary classes, $\mathcal{G}^+$: generated from Gömböc, $\mathcal{F}$: secondary classes with less than 8 equilibria, $\mathcal{I}$: irreducible ancestors, $\mathcal{M}$: minimal polyhedra, $\mathcal{M}^+$: generated from minimal polyhedra.

showed how the ancestors can define a natural partition of the secondary classes.

Subresult 1.a: There are only three irreducible ancestors of size at most 10: the secondary classes of Gömböc, the regular tetrahedron and the right square pyramid. Consequently the Gömböc is the ancestor of the secondary classes of size less than 8.

Subresult 1.b: The secondary classes of the minimal polyhedra are irreducible.

Subresult 1.c: Every secondary class has a unique irreducible ancestor, so any reduction sequence produces it.

2 ENUMERATING QUADRANGULATIONS

2.1 RESEARCH GOAL

Refining the primary classes evidently poses the question: how many different secondary classes are the primary classes divided into? And specifically, what are the possible secondary classes? As the definition of a secondary class involves a quadrangulation, a strongly linked question is: how many different quadrangulations exist for a given size limit?

The literature offers only partial answers to these questions. The family of the *2-coloured* quadrangulations has already been counted by Wormald [9] with a polynomial time algorithm, and has also been exhaustively enumerated by Walsh [7, 8] with given size limit, without storing them in the memory. However, up to my best knowledge, no one has ever counted or enumerated the quadrangulations *without colouring*.

Another direction to exhaustively enumerate certain graph families without storing them is the approach of the software Plantri. It generates the members of a family from a starting set by applying the given expansions systematically. It avoids duplications by ensuring that every graph is allowed to be constructed only by one unique sequence of expansions. However, the closest family which can be generated by Plantri is the family of simple quadrangulations (no parallel edges allowed).

The second research goal is to give algorithms to enumerate the quadrangulations (allowing parallel edges), and present the number of possible quadrangulations of a given size.

2.2 RESEARCH METHOD

Three algorithms are proposed to count and enumerate the quadrangulations of given size. The first is to use Plantri to enumerate a larger set, containing a subset equivalent to the quadrangulations. The larger set is the set of triangulations, the subset are the ones which can coloured as an equilibrium triangulation (see Figure 1.2d). Hence the output, i.e. the larger set can be filtered by checking each instance if it belongs to the subset.

The second method is to extend Plantri to directly generate the desired set of quadrangulations with graph expansions. To achieve that, one has to define the necessary graph expansions,

(a) Simple case (b) Parallel case

Figure 2.1. 3-splitting

a starting set, and some unique construction sequence for any quadrangulation. The chosen expansions are the 1-, 2- and 3-splitting, denoted by $S_{1,3}$. The first two are shown in Figure 1.3, and 3-splitting is shown in Figure 2.1.

The third method is based on filtering the output of the enumeration of Walsh [8]. To present the list of quadrangulations without the colouring, one has to remove colouring obviously. However, a quadrangulation may admit multiple, non-isomorphic 2-colourings, so the repetition has to be avoided in this case. If all possible 2-colourings of a quadrangulation are isomorphic, then we say it is a *self-dual*.

2.3 RESULTS

The theoretical results below are followed by computational results attained by executing the developed method, showing the cardinalities in question (Table 2.1).

Main Result 2 (Published in: [A2] [C4, C5]): I developed and implemented a parallel method to enumerate both the multi-quadrangulations and the secondary classes. Executing it on a parallel system, I gave some exact cardinalities. I also gave more effective parallel and sequential algorithms.

Subresult 2.a: My first method built on filtering the triangulation enumeration feature of `plantri` enumerates the quadrangulations. Its execution in the European Grid Infrastructure (EGI)

	$\mathcal{Q}(n)$	$e_{SD}(n)$	$\sum_s e(s, n-s)$	$\mathcal{I}(n)$
$n = 3$	1	-	2	0
$n = 4$	3	2	4	0
$n = 5$	7	-	14	0
$n = 6$	30	8	52	0
$n = 7$	124	-	248	0
$n = 8$	733	50	1416	1
$n = 9$	4586	-	9172	0
$n = 10$	33373	380	66366	1

Table 2.1. Cardinalities of multiquadrangulations ($\mathcal{Q}$), self-dual secondary classes (e_{SD}), secondary classes ($\sum e$) and irreducibles ($\mathcal{I}$)

with the Saleve tool gave the cardinalities of quadrangulations for different sizes up to 10 vertices. The generated data set confirms Subresult 1.a.

Subresult 2.b: I proved that the family of multiquadrangulations is generated by the splitting $S_{1,3}$. Thus, I gave a parallel algorithm to enumerate both the multiquadrangulations without colouring and the 2-coloured multiquadrangulations, which is built on the canonical construction path method.

Subresult 2.c: I gave a sequential algorithm to enumerate the multiquadrangulations built on enumerating the 2-coloured multiquadrangulations by the algorithm of Walsh, but ensuring that each multiquadrangulation is listed only with one colouring.

3 SCHEDULING JOBS OF UNKNOWN LENGTH

3.1 RESEARCH GOAL

As Plantri supports dividing the task into independent parts, it could be executed in parallel in a grid infrastructure using the Saleve tool [5]. To attain the computation results from the previous section, the overall processing time would have taken about eighty days on an average single machine, however, it turned out that achieving high speedup by executing on a parallel infrastructure is rather complicated because of two difficulties. First, dividing *optimally* a computational task into parts with similar processing time is known to be NP-hard, meaning that only the smallest, most trivial cases are solvable in reasonable time. Secondly, there is no *a priori* information on the processing times of the parts, so this computationally hard problem is to be solved under uncertainty. It is important to notice that these troubles are not specific to the execution of the graph enumeration, they may also accompany a wide range of parallel computations, including the parameter sweep applications (PSAs) which the thesis focuses on. I also study the batches of PSAs (BPSA) model where the jobs are executed in batches to be preceded by a sequence independent setup work, and in this case the approximation ratio of 3 is reached.

The third research goal is to design a framework to deal with uncertainty and complexity of this scheduling problem. In addition, the theoretical iterative scheduling algorithm has to be implemented to manifest a proof of concept.

3.2 RESEARCH METHOD

In order to maintain computational tractability, a trade-off is accepted: the underlying optimization problem is approximated by quickly yielding a solution guaranteed to be “good enough”, whose cost is at most two times the optimal cost. The cost is the makespan, i.e. the maximal completion time (roughly, the wall clock time between submission and completion). As a means to deal with uncertainty, the first step is adopting a presumption observed by Downey and Feitelson [3]: a user developing a PSA tends to repeat the execution several times. This behaviour may originate from various reasons such as testing, fixing bugs, adding new features such as more detailed output, improved precision etc. Anyway, it allows us to build a historical database by measuring the individual machine completion times, so the processing time characteristics of the problem at hand can be charted gradually. For practical reasons, the historical database and the machine assignment descriptions are also kept brief.

The Saleve tool [5] used for implementation supports developing a lightweight PSA that can be executed both on a personal computer (standalone mode) and on a distributed infrastructure (client-server mode). Saleve has been developed since 2005 by a small group at the BME including the author of the dissertation [C1, C2, A1, C3].

3.3 RESULTS

The working of the algorithm is illustrated in the following example, which is a trace of enumerating quadrangulations of size 9. The results are summarized after the illustration.

EXAMPLE: The processing times of the 49566 jobs of a plane graph generation [A2] are shown in Figure 3.1 (top), which are *unknown in advance*. The jobs has to be scheduled on 12 identical parallel machines, minimizing the makespan.

The whole process is illustrated in Figure 3.1. Horizontal axes correspond to the job number j, vertical axes denote time (length). The first diagram (on top) visualizes the processing times. The second diagram of Figure 3.1 shows the initial schedule and the completion times of each machine. The initial schedule assigns the uniform-sized chains to the 12 machines and the assignment intervals (chains) are separated by vertical lines, and their machine indices are also displayed. There are 12 rectangles on the second diagram, and the width of the i^{th} rectangle corresponds to the *size* (number of jobs) of the assignment of machine i, while its height corresponds to the *length* (processing time) of this assignment. Obviously the makespan is the height of the highest rectangle, which is 760177.

The first iteration splits the first and the second chains, estimate their sizes (not shown in the figure) and computes the next schedule, executes it and measures its completion times, shown in the third diagram. Note that some assignments were contracted (wider rectangles), while the longest ones (highest rectangles) were split into smaller (narrower rectangles) ones. The makespan is still 433898, so the next iteration comes. The framework stops when the makespan reaches 365471 in the third schedule (last diagram of Figure 3.1).

MAIN RESULT 3 (Published in: [A1, A3, A4] [C1–C3, C6, C7]): I developed and implemented a generic framework to schedule PSAs of unknown lengths for identical parallel machines. I proved that the makespan converges to an approximation, assuming that the runtime profile of the task does not change between the executions.

SUBRESULT 3.a: The framework reaches the 2-approximation for any PSA after a finite number of iterations, based on building the

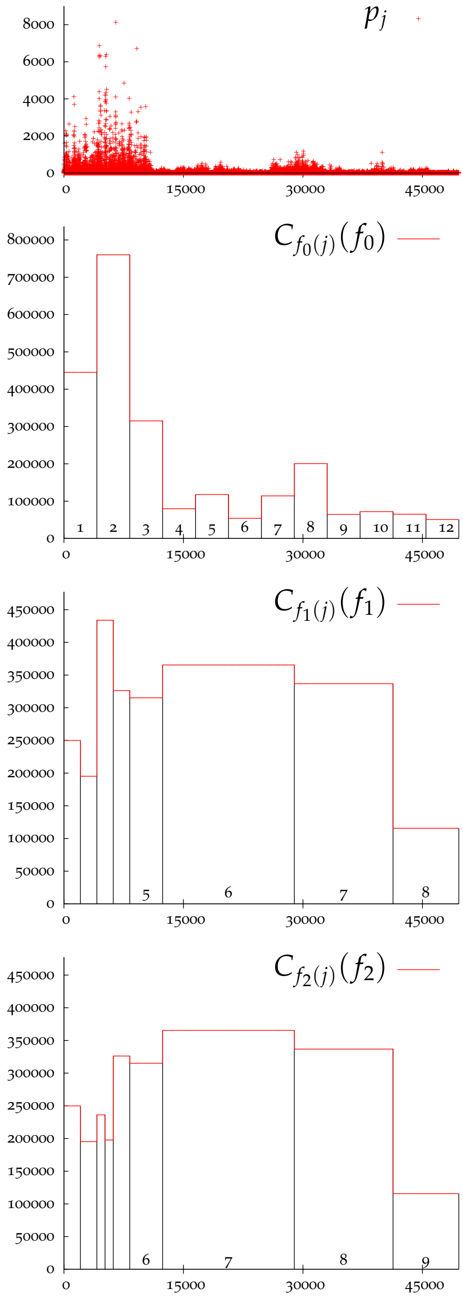


Figure 3.1. Processing times p_j and machine completion times $\vec{C}(f_q)$.

historical database only from the measured machine completion times.

Subresult 3.b: The Saleve tool has been extended with this framework to measure and collect runtimes and re-balance machine loads at the next execution. As a result, the user experiences faster completion time and that the partition of the parameter space is computed automatically instead of involving the user. The theoretical properties of the framework are confirmed by experimental test cases.

Subresult 3.c: I proved in theory that the framework can be generalized to handle BPSAs, for which it guarantees a 3-approximation, assuming that the setup times are known in advance.

4 BIBLIOGRAPHY

JOURNAL ARTICLES OF THE DISSERTATION

[A1] P. Dóbé, R. Kápolnai and I. Szeberényi. "Saleve: toolkit for developing parallel grid applications". In: *Híradástechnika* LXIII.1 (2008), pp. 60–64.

[A2] R. Kápolnai, G. Domokos and T. Szabó. "Generating spherical multiquadrangulations by restricted vertex splittings and the reducibility of equilibrium classes". In: *Periodica Polytechnica Electrical Engineering and Computer Science* 56.1 (2012), pp. 11–20. DOI: 10.3311/PPee.7074.

[A3] R. Kápolnai and I. Szeberényi. "Scheduling jobs and batches based on historical data". In: *International Journal of Computers and Communications* 7.3 (2013), pp. 55–63.

[A4] T. Kis and R. Kápolnai. "Approximations and auctions for scheduling batches on related machines". In: *Operation Research Letters* 35.1 (2007), pp. 61–68. DOI: 10.1016/j.orl.2006.01.005.

CONFERENCE PROCEEDINGS

[C1] P. Dóbé, R. Kápolnai, A. Sipos and I. Szeberényi. "Applying the improved Saleve framework for modeling abrasion of pebbles". In: *Large-Scale Scientific Computing*. Vol. 5910. Lecture Notes in Computer Science. Sozopol, Bulgaria: Springer, 2010, pp. 467–474. DOI: 10.1007/978-3-642-12535-5_55.

[C2] P. Dóbé, R. Kápolnai and I. Szeberényi. "Saleve: supporting the deployment of parameter study tasks in the grid". In: *Cracow Grid Workshop*. Krakow, Poland, 2007, pp. 276–282.

[C3] P. Dóbé, R. Kápolnai and I. Szeberényi. "Simple grid access for parameter study applications". In: *Large-Scale Scientific Computing*. Vol. 4818. Lecture Notes in Computer Science. Sozopol, Bulgaria: Springer, 2008, pp. 470–475. DOI: 10.1007/978-3-540-78827-0_53.

[C4] R. Kápolnai and G. Domokos. "Inductive generation of convex bodies". In: *The 7th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications*. Kyoto, Japan, 2011, pp. 170–178.

[C5] R. Kápolnai, G. Domokos and T. Szabó. "Másodlagos egyensúlyi osztályok gráfelméleti származtatása". In: *XI. Magyar Mechanikai Konferencia*. In Hungarian. 2011.

[C6] R. Kápolnai, I. Szeberényi and B. Goldschmidt. "Approximation of repeated scheduling chains of independent jobs of unknown length based on historical data". In: *Recent Advances in Computer Science*. Rhodes, Greece: WSEAS Press, 2013, pp. 41–46.

[C7] T. Kis and R. Kápolnai. "A 2-approximation algorithm and a truthful mechanism for scheduling batches on machines running at different speeds". In: *7th Workshop on Models and Algorithms for Planning and Scheduling Problems*. Siena, Italy, 2005, pp. 157–160.

OTHER PUBLICATIONS

[O1] P. Dóbé, R. Kápolnai and I. Szeberényi. "Saleve: párhuzamos grid-alkalmazások fejlesztőeszköze". In: *Híradástechnika* LXII.12 (2007). In Hungarian, pp. 32–36.

[O2] R. Kápolnai and G. Domokos. “Morphology classes of convex bodies based on static equilibria. (Konvex testek egyensúlyi morfológiaosztályainak feltérképezése)”. In: *Networkshop*. In Hungarian. 2010.

[O3] *Konvex testek származtatása gráfok induktív generálásával*. Seminar Talk. In Hungarian. BME Dept. of Computer Science and Information Theory. 2010.

[O4] *Konvex testek származtatása gráfok induktív generálásával*. Seminar Talk. In Hungarian. BME Dept. of Control Engineering and Information Technology. 2010.

[O5] *Konvex testek származtatása gráfok induktív generálásával*. Seminar Talk. In Hungarian. BME Dept. of Mechanics, Materials & Structures. 2011.

[O6] *Négyszögelések előállíthatósága monoton csúcsosztással*. Seminar Talk. In Hungarian. ELTE Egerváry Research Group on Combinatorial Optimization. 2014.

[O7] D. Németh, R. Kápolnai and I. Szeberényi. “Grid az oktatásban”. In: *Networkshop*. In Hungarian. 2007.

REFERENCES

[1] G. Domokos, Zs. Lángi and T. Szabó. “The genealogy of convex solids”. Manuscript, http://arxiv.org/abs/1204.5494, last access: 2014.06.06. 2012.

[2] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci and J. C. Hart. “Spectral surface quadrangulation”. In: *ACM Transactions on Graphics* 25 (3 2006), pp. 1057–1066. DOI: 10.1145/1141911.1141993.

[3] A. B. Downey and D. G. Feitelson. “The elusive goal of workload characterization”. In: *SIGMETRICS Performance Evaluation Review* 26.4 (1999), pp. 14–29. ISSN: 0163-5999. DOI: 10.1145/309746.309750.

[4] H. Edelsbrunner, J. Harer and A. Zomorodian. "Hierarchical Morse complexes for piecewise linear 2-manifolds". In: *Proceedings of the Seventeenth Annual Symposium on Computational Geometry*. SCG '01. ACM, 2001, pp. 70–79. DOI: 10.1145/378583.378626.

[5] Zs. Molnár and I. Szeberényi. "Saleve: simple web-services based environment for parameter study applications". In: *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*. 2005, pp. 292–295. DOI: 10.1109/GRID.2005.1542757.

[6] P. Várkonyi and G. Domokos. "Static equilibria of rigid bodies: dice, pebbles and the Poincaré–Hopf theorem". In: *Journal of Nonlinear Science* 16 (2006), pp. 255–281. DOI: 10.1007/s00332-005-0691-8.

[7] T. R. Walsh. "Generating nonisomorphic maps and hypermaps without storing them". In: *GASCom*. 2012.

[8] T. R. Walsh. "Generating nonisomorphic maps without storing them". In: *SIAM Journal on Algebraic and Discrete Methods* 4.2 (1983), pp. 161–178. DOI: 10.1137/0604018.

[9] N. C. Wormald. "Counting unrooted planar maps". In: *Discrete Mathematics* 36.2 (1981), pp. 205–225. DOI: 10.1016/S0012-365X(81)80016-7.