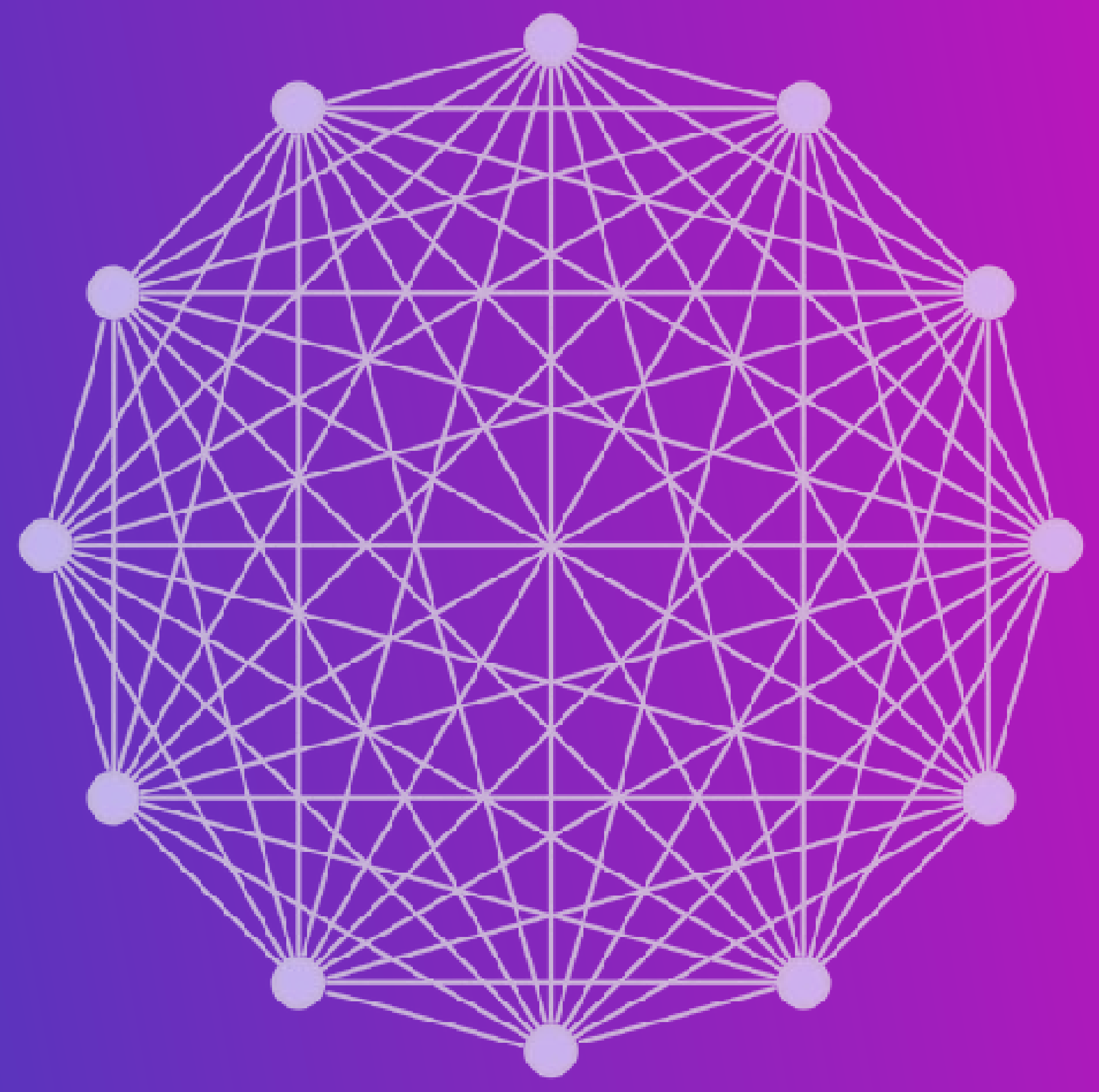


Combinatorial Design and Optimization: **The Oberwolfach Problem**

Gabriele Dragotto

Joint work with F. Salassa, T. Traetta, M. Buratti, F. Della Croce

CANADA EXCELLENCE RESEARCH CHAIR DATA SCIENCE FOR REAL-TIME DECISION-MAKING

POLITECNICO DI TORINO

ALCO

Outline

Content

- A gentle introduction to *Combinatorial Design*.
- Graph decompositions and the *Oberwolfach Problem*.
- Our contribution in pills.

Focus

Two fold:

- Exploit optimization tools to solve combinatorics problems
- Derive theoretical results from computational results

k-regular graph

A k-regular graph is a graph where each vertex has the same degree.

spanning subgraph

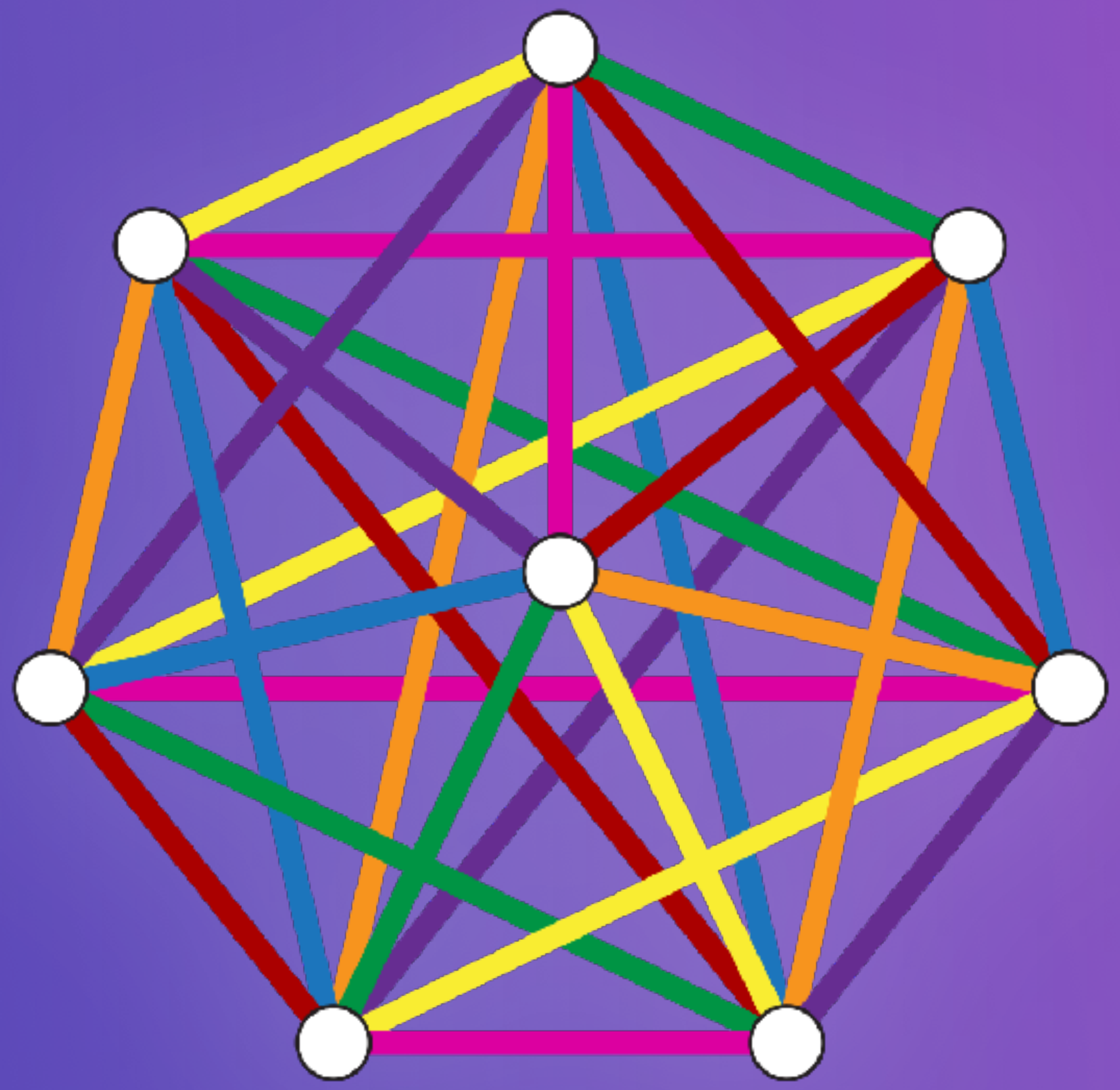
A spanning subgraph of *G* is a graph with the same vertex set of *G.*

k-factor

A k-factor of a graph *G* is a *k-regular spanning subgraph* of *G*.

k-factorization

A k-factorization partitions the edges of the graph into disjoint k-factors.



A *1-factorization* of K_8 (7-regular graph)
Each color is a single 1-factor, and there are 7 copies.

(With K_n we denote the complete graph over n vertices)

(Eppstein, 2011)

Combinatorial Design

“Branch of combinatorial mathematics dealing with the existence, construction and properties of finite sets whose arrangements satisfy criteria of balance and symmetry.”

Applications:

Tournaments design, software testing, algorithm design and analysis, networking, design of experiments, cryptography.

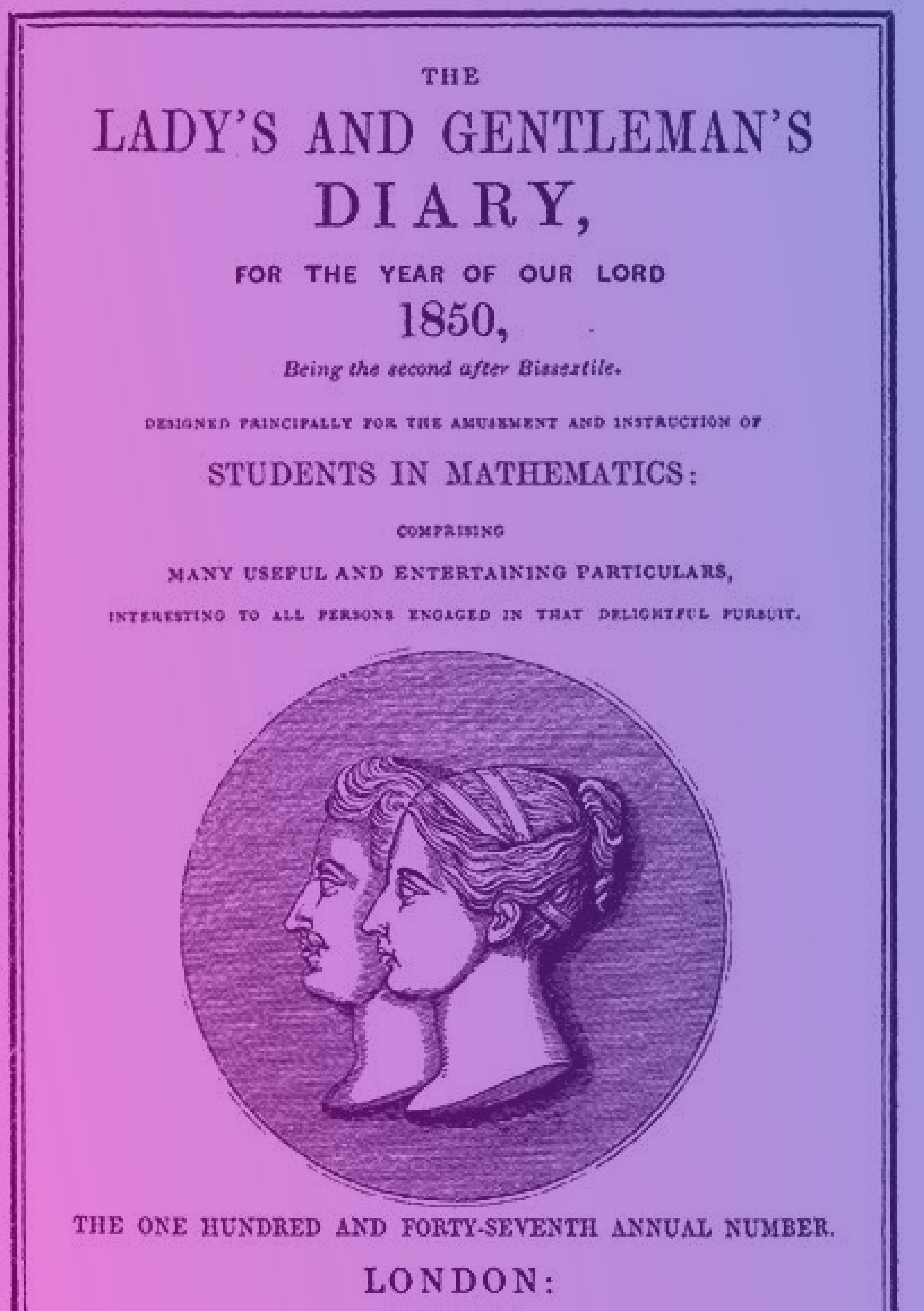
Kirkman's schoolgirl problem

"(*KSP*) 15 young ladies in a school walk out 3 abreast for 7 days in succession: it is required to arrange them daily so that no *two* shall walk twice abreast."

(Kirkman, 1850)

And more...

Steiner triple systems
Block designs
Latin squares *(Euler, 1723)*
Sudokus
Balanced tournament design (BTD)
Howell designs
Orthogonal codes
Covering arrays



THE

LADY'S AND GENTLEMAN'S DIARY,

FOR THE YEAR OF OUR LORD

1850,

Being the second after Bissextile.

DESIGNED PRINCIPALLY FOR THE AMUSEMENT AND INSTRUCTION OF

STUDENTS IN MATHEMATICS:

COMPRISING

MANY USEFUL AND ENTERTAINING PARTICULARS,

INTERESTING TO ALL PERSONS ENGAGED IN THAT DELIGHTFUL PURSUIT.

THE ONE HUNDRED AND FORTY-SEVENTH ANNUAL NUMBER.

LONDON:

The Oberwolfach Problem

Gerard Ringel surfing from Wikipedia

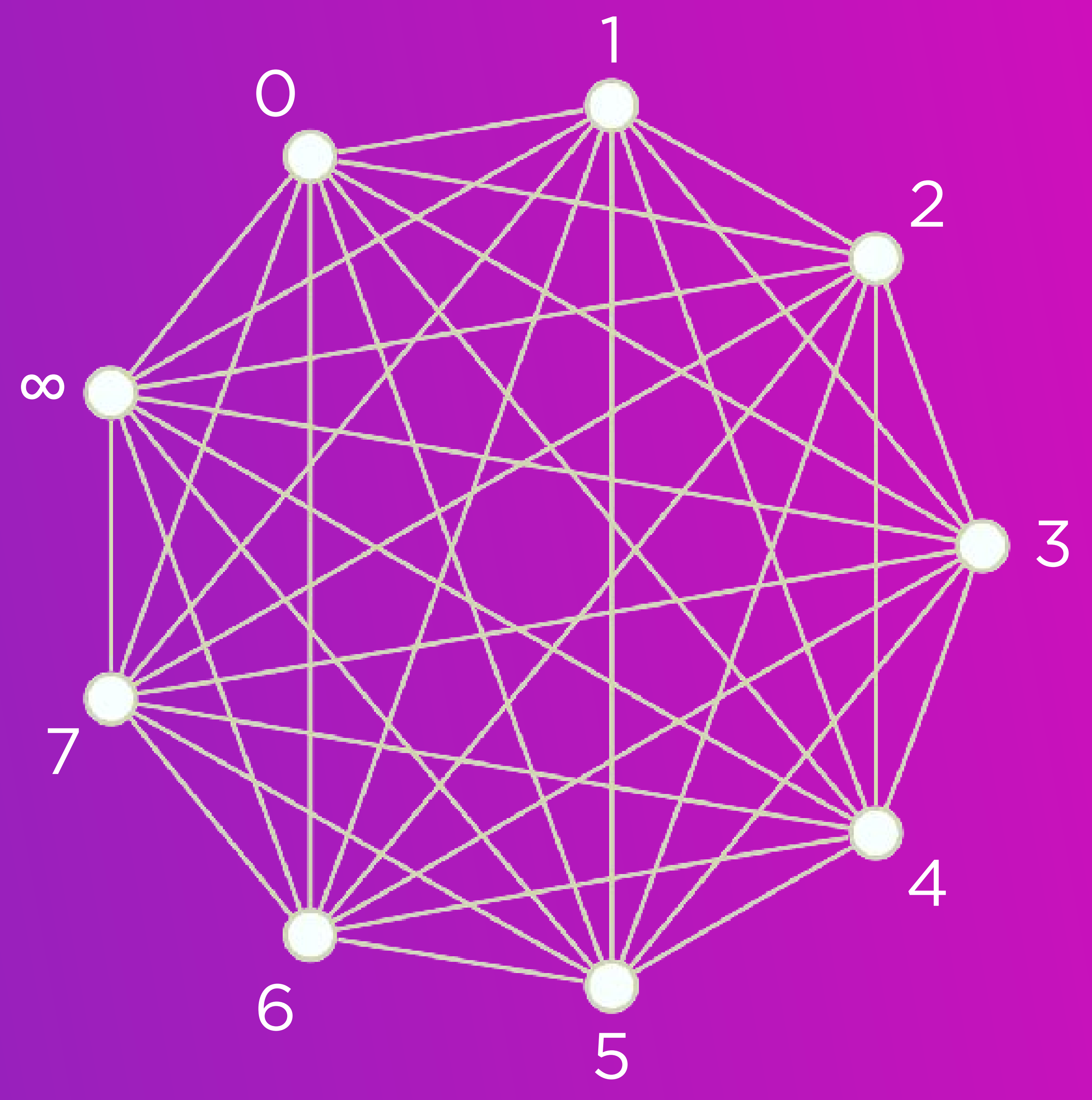
In conferences held at the Institute, participants usually dine together in a room with circular tables of different sizes, and each participant has an assigned seat.

Gerhard Ringel asked whether there exists a seating arrangement for an odd number v of people and $(v-1)/2$ meals so that each participant is seated next to every other participant exactly once.

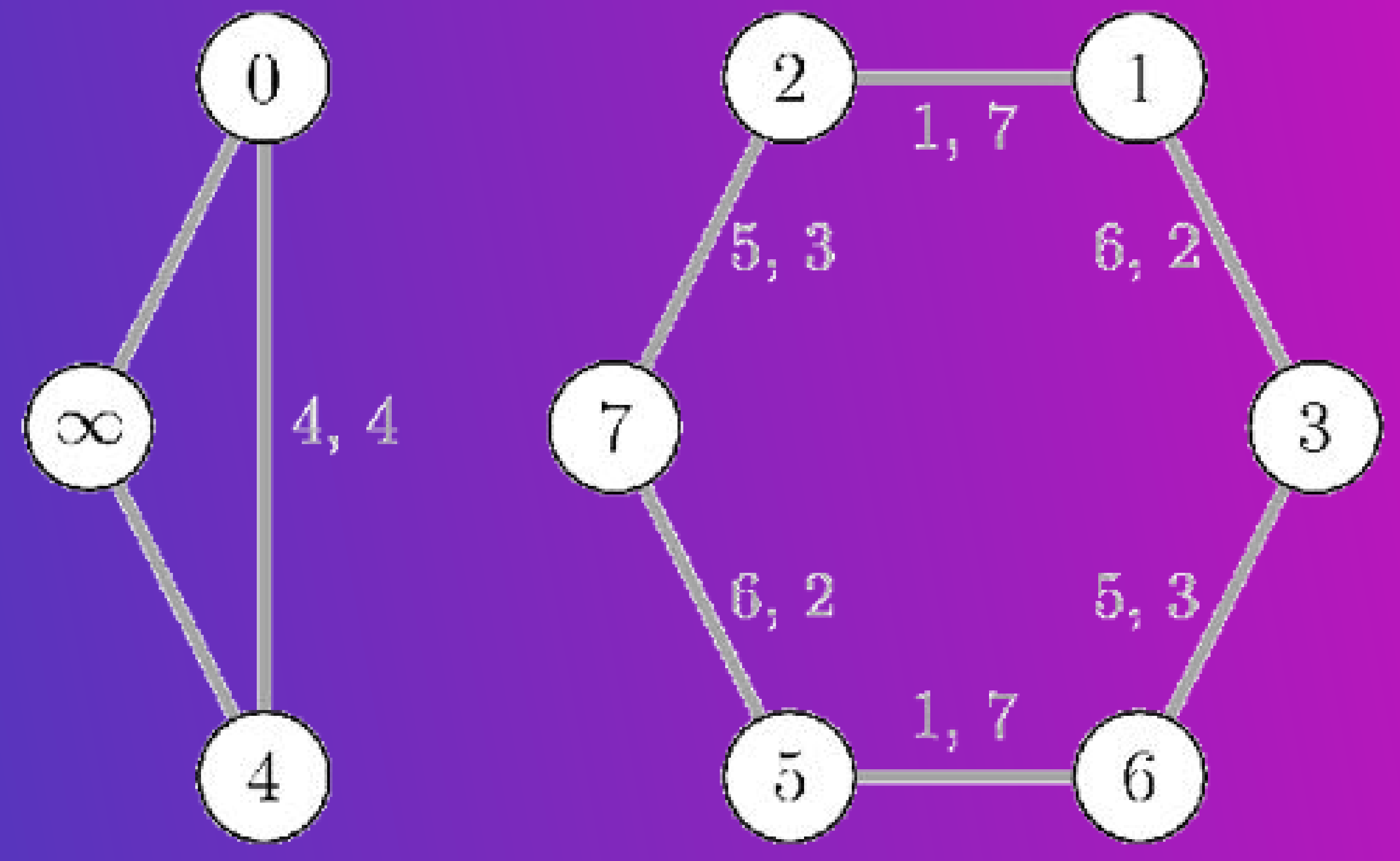
KSP = Oberwolfach with 15 people and all tables of 3.

Arranging meals: an example *OP(3,6)*

2 Tables respectively for 3, and 6 guests.
Since there are 9 participants, the problem corresponds to the 2-factorization of K_9 into *4* disjoint copies of a factor $F = [3,6]$



Difference Methods



A labeled factor F=[3,6] for the *OP(3,6)*. Namely the first meal or the *special 2-factor*.

How can we generate 2-factorization?

We label the graph exploiting the so-called *difference methods.* Such labelling allows us to generate only a special 2-factor, and derive the remaining ones with roto-translations.

Each edge inherit 2 labels, namely 2 differences, from an algebraic operation between labels of adjacent nodes.

By imposing specific rule on the *set(s) of differences*, we are able to solve the OP.

differences

Algebraic operations between node labels performed in a cyclic abelian group

The first "*well-behaved*" 2-factor is the *generator* for all the other copies (2-factors)

special 2-factor

The *special* 2-factor is the one having a *difference-set* with a particular configuration.

Then...

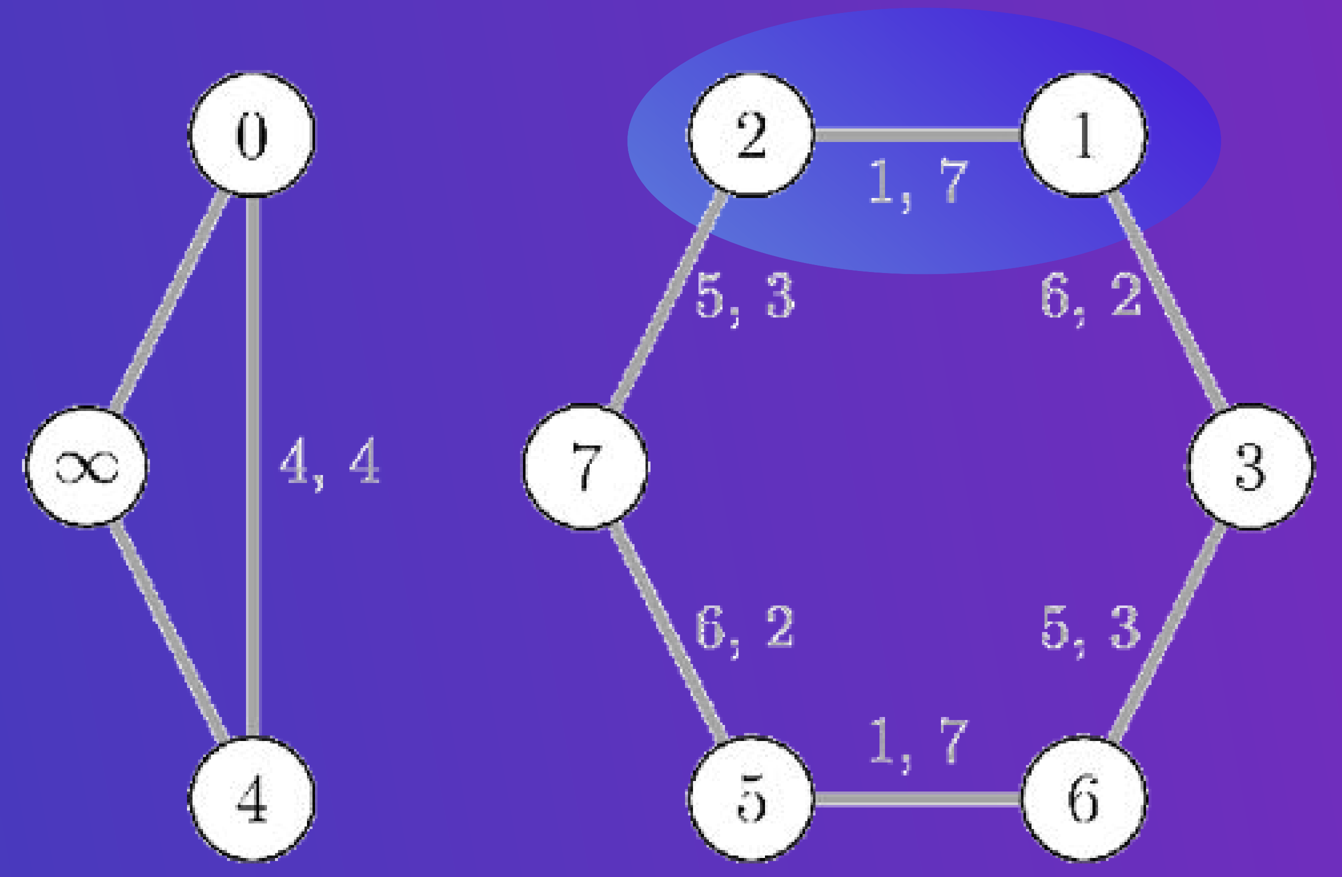
special 2-factor

The difference method approach reduces the problem to finding one *well-structured 2-factor* and build complete factorizations thanks to (roto)-translations.

Outputs

(n-1)/2 copies of F, generated by a *special 2-factor*

The *well-structured 2-factor*



Let's consider nodes 1 and 2. There are two differences involved:

$$\delta_1 = 1 - 2 = -1 \mod (\gamma) = 7 \qquad \gamma = \frac{9-1}{2}$$

$$\delta_2 = 2 - 1 = 1 \mod (\gamma) = 1$$

Each difference is given by an algebraic operation over the cyclic group

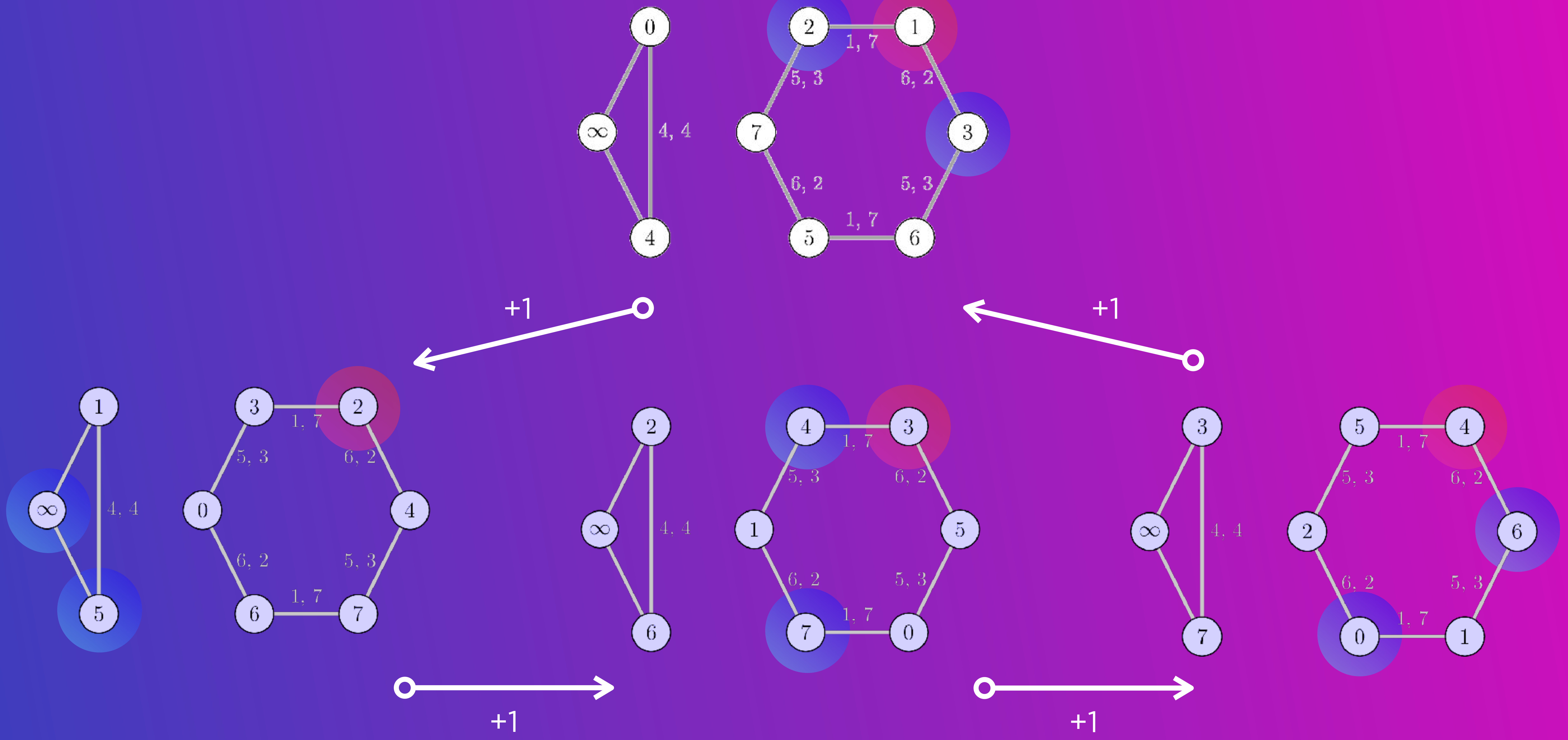
$$\Delta F = \{a - b \mod (\gamma) : a, b \in V(F)\}$$

Each node has a label in $\mathbb{Z}_{2\gamma}$. We seek to assign labels so that the *difference-set* is as follow:

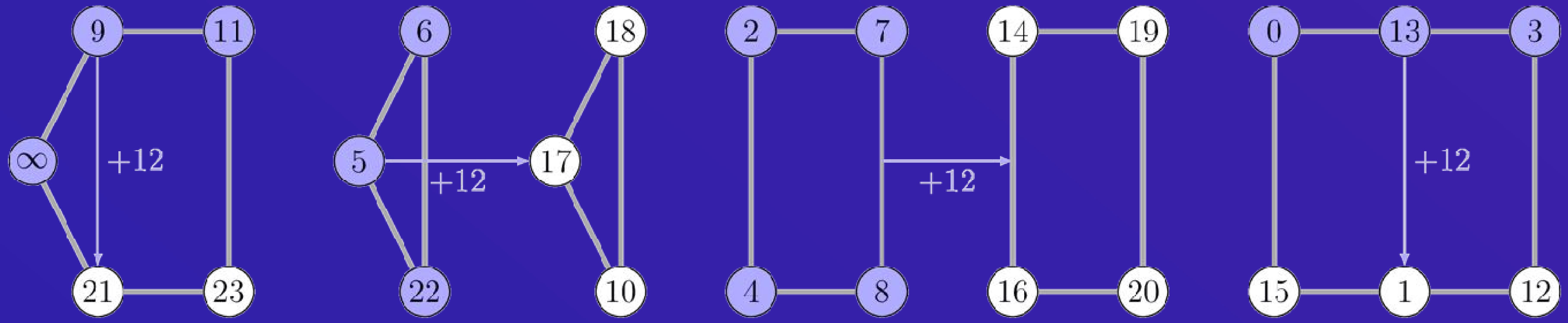
$$\Delta F = \{\delta_1, \delta_2, ..., \delta_{13}, \delta_{14}\} = \{1, 1, 2, 2, ..., 7, 7\}$$

Plus, there is a fixed node named ∞, which does not produces differences nor translates in different factors.

Our meal problem



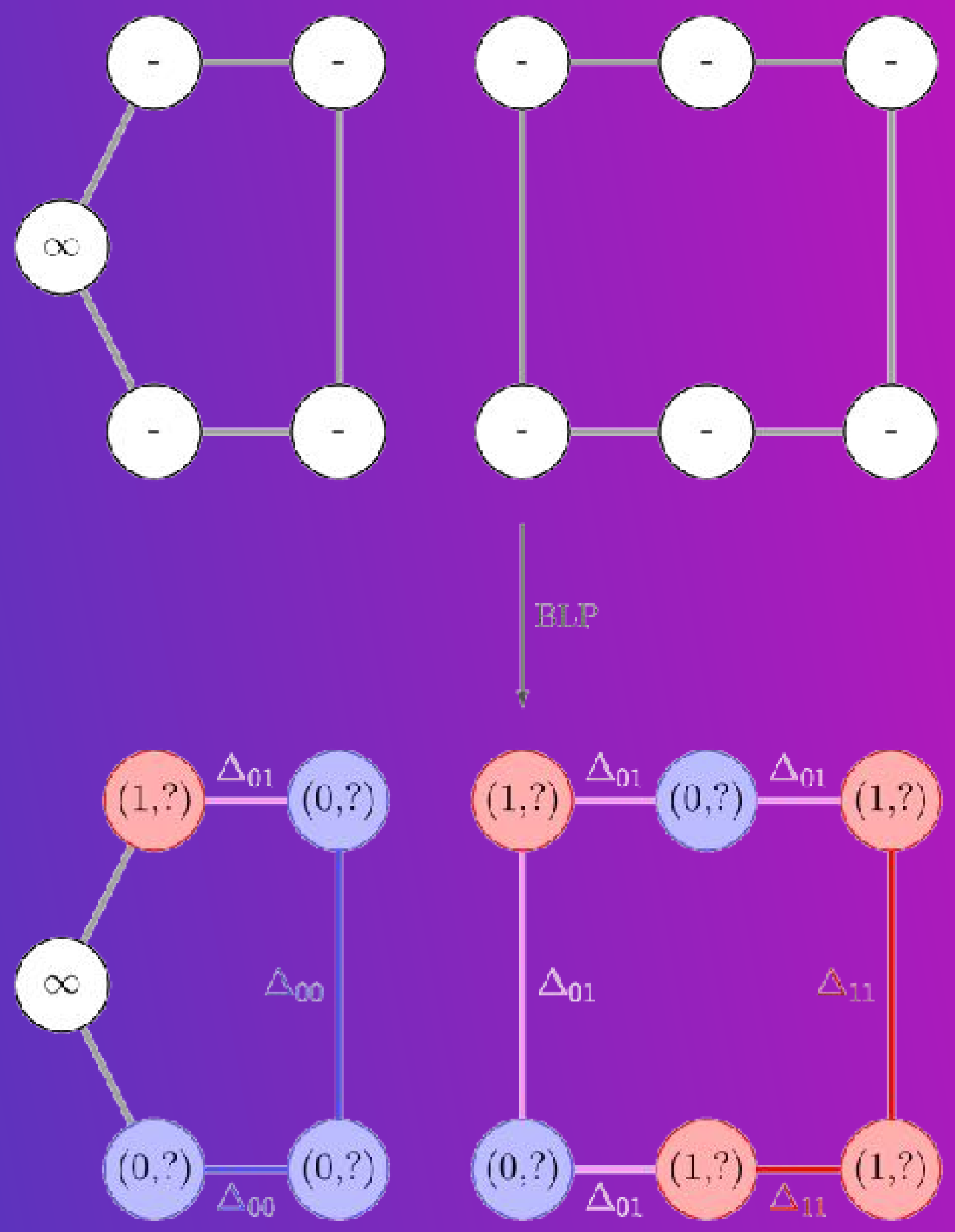
Exploiting symmetries



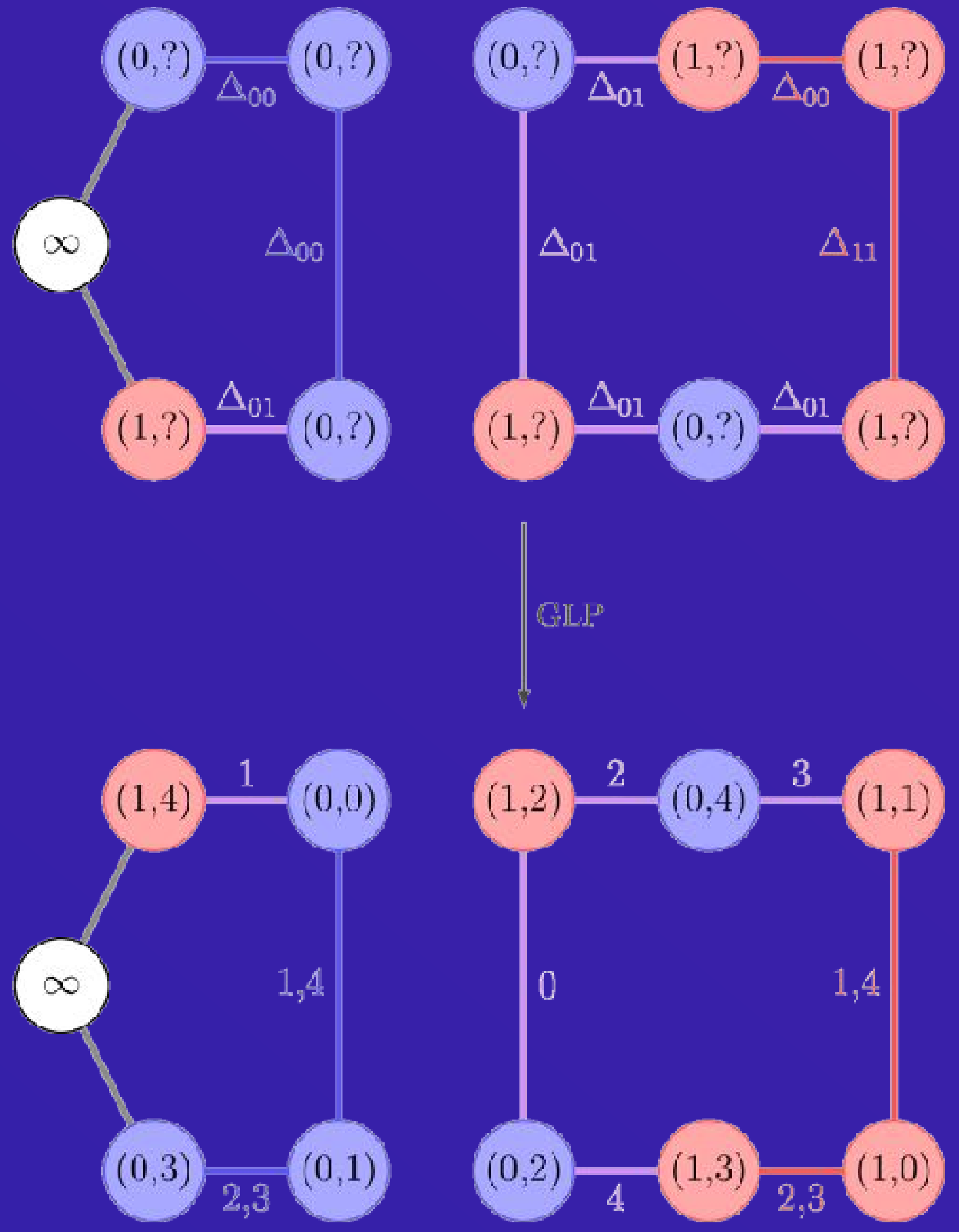
Graphs of specific orders can be further simplified exploiting symmetries.

$$4t + 1, 4t + 2 \quad t \in \mathbb{N}$$

2-rotational methods



2(n)-rotational difference methods, namely several difference sets



Two-step formulation to label the *special* 2-factor.

Summing up

Combinatorial Design provides the algebraic methods to construct and prove the existence of 2-factorizations, exploiting *difference methods* and *symmetries.*

But

How do we generate a specific well-structured 2-factor?

literature

Does not provide computational works for the OP, with the exception of *Deza et al, 2008*.

Our contribution, in pills

INTEGRATE **EXISTING METHODS** WITH **OPTIMIZATION** TOOLS

- Tackles the OP by searching for special 2-factors, providing extensive results.

- Exploits Constraint Programming and Integer Programming to model difference methods and symmetries (+specific algo)

- Provides theoretical contributions stemming from the computed solutions

In particular, we solve the complete Oberwolfach Problem with all the graphs of orders in *[40,60]*

Constraint Programming

Since we are dealing with a feasibility problems where variables have mutually exclusive integer values.

$$K_{2n+1} \Rightarrow F : |F| = 2n + 1$$

From the complete graph K_n we extract an unlabelled *2-factor F*

$$\Delta\Gamma = \{x - y \mid xy \in E(\Gamma \setminus \{\infty\})\}$$

Several (one) difference-sets inherit their elements from the integer labels assigned to vertices.

$$V(F) = \mathbb{Z}_{2n} \cup \{\infty\}$$
$$\Delta F \supset \mathbb{Z}_{2n} \setminus \{0\}$$
$$F + n = F$$

These difference-sets should well-behave and fulfil certain properties. The well-behaved labelling is the sought-after *special 2-factor.*

Several CP models

1-rotational

- A simple modulo translation between different 2-factors (*meals*).
- Underlying symmetries allows to work on a simplified graph.

2-rotational

- A more complex roto-translation links different 2-factors (*meals*).
- Each node has a label made up of two coordinates.
- The labelling is decomposed in two subproblems. We propose a polynomial-time algorithm to solve the first one.

$$
\begin{aligned}
 & V = \{n_i \mid n_i \in G\} \\
\texttt{alldifferent(V)} \quad & dom(V) = [0, 2\gamma) \\
\texttt{card}(V \mid n_i) + \texttt{card}(V \mid (n_i + \gamma 2\gamma)) = 1 \quad & \forall n_i \in \mathbb{Z}_\gamma \\
 & D = dE \cup dO \\
dE = \{(n_\alpha - n_\beta 2\gamma)\} \quad & \forall \alpha, \beta \in V \wedge [\alpha, \beta] \\
dO = \{\omega_i - \eta, \eta - \omega_i 2\gamma\} \quad & \forall o_i = [\omega_1, ..., \omega_i] \in O, \\
 & \eta = \omega_1 + \gamma 2\gamma \\
\texttt{alldifferent(D)} \quad & dom(D) = (0, 2\gamma) \setminus \{\gamma\}
\end{aligned}
$$

Combinatorial Mysteries

Handbook of CD
(Colburn and Dinitz, 2007)

Two-factorization of the Complete Graph
(Rosa, 2003)

The oberwolfach problem and factors of uniform odd length cycles
(Alspach et al., 1989)

Untersuchungen über das Oberwolfacher Problem
(Piotrowski., 1979)
UNPUBLISHED!

For the specific case of $OP(^{2}3,5)$, is likely that no solution exists. Despite this fact, apparently no proof (either computational or analytical) has been published for $OP(^{2}3,5)$.

CP not effective

Our best formulation for the complete problem did not provide results with days of computing time with CP.

Exploiting IP

The *OP* $(^{2}3, 5)$ is the problem of arranging 11 people in 2 tables of 3 and 1 table of 5 for 5 meals. Each person has a label in [0,10].
The IP formulation enumerates every feasible combination of labels for tables of 3 (*triplets T*) and tables of 5 (*5-sets F*).

Afterwards, it seeks to select for each meal, one 5-set and two triplets so that each node is *seated* next to every other node exactly once over all the meals.

The continuous relaxation has no solution!

$$min(-)$$

$$S.T.$$

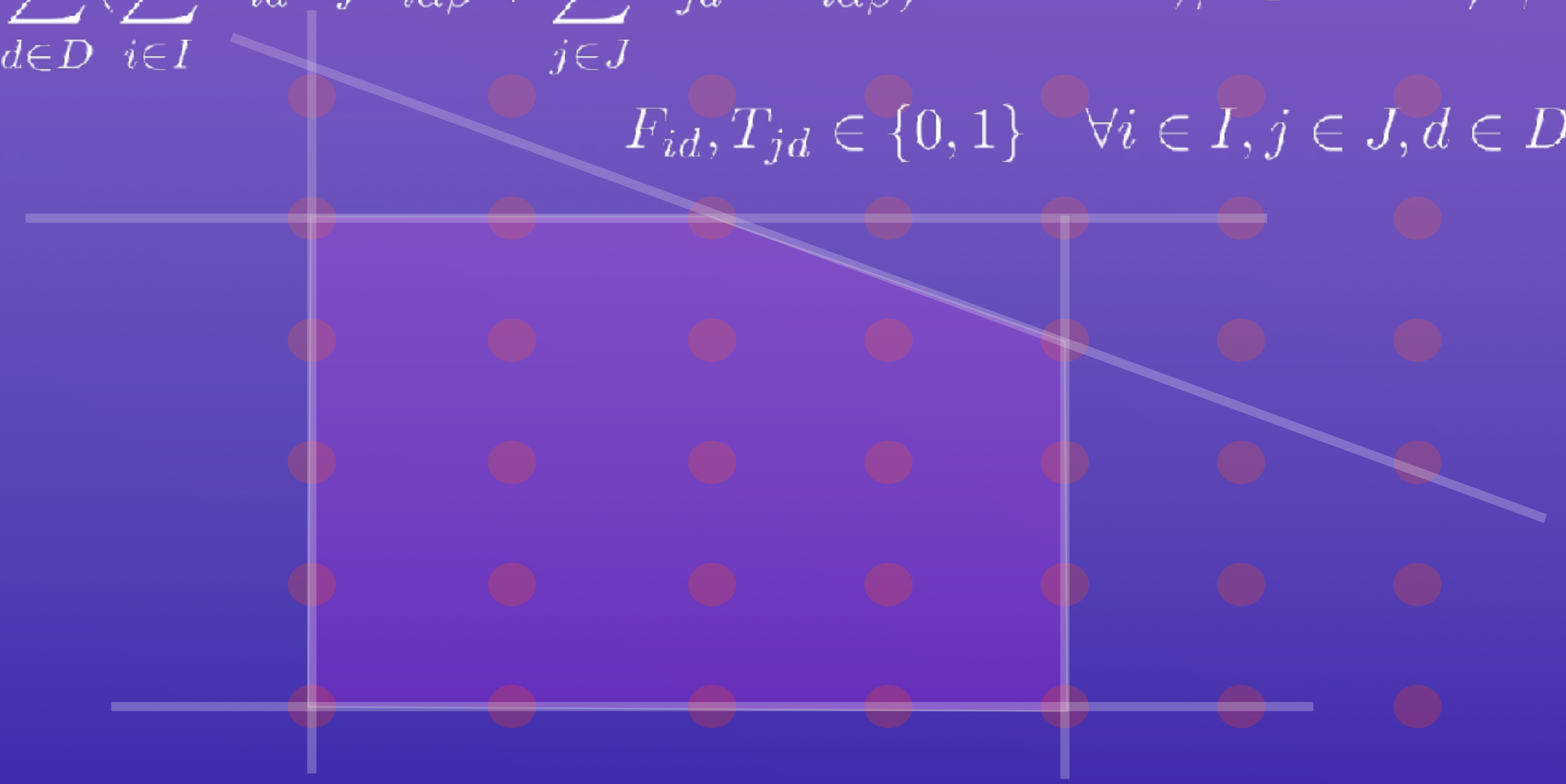
$$\sum_{i\in I} F_{id} = 1 \qquad \forall d \in D$$

$$\sum_{j\in J} T_{jd} = 2 \qquad \forall d \in D$$

$$\sum_{i\in I} F_{id}\cdot fl_{il} + \sum_{j\in J} T_{jd}\cdot tl_{jl} = 1 \qquad \forall d \in D, \forall l \in L$$

$$\sum_{d\in D}\Big(\sum_{i\in I} F_{id}\cdot fa_{i\alpha\beta} + \sum_{j\in J} T_{jd}\cdot ta_{i\alpha\beta}\Big) = 1 \qquad \forall \alpha,\beta \in L \land \alpha \neq \beta$$

$$F_{id}, T_{jd} \in \{0,1\} \qquad \forall i \in I, j \in J, d \in D$$



Computational results

Previous results from Deza et al. (2008) solved the *OP* with 18≤n≤40 with *undisclosed* algorithms, running the tests on SHARCNET (namely a *Compute Canada cluster*)

*For each order *n* of a complete graph, we solve the *OP* on all the configurations given by the *integer partitions* of *n*.

We solve the *OP* with 40≤n≤60 on a single household machine with CPLEX 12.7
Intel Core i5-3550 @ 3.30GHz with 4GB of RAM

- Each instance takes less than a few seconds (w.r.t. *difference method*).
- We are able to factorize graph up to the order of *n=120* in less than a minute.

Table of results

#	Type	Method	Time (s)	Partitions	Solved	Avg. Time (s.ms)
40	4t	(Derived from 39)	911	1756	1756	00.519
41	4t+1		807	2056	2056	00.393
		1 Rotational	90		1433	00.063
		A-2 Rotational	717		623	01.151
42	4t+2	(Derived from 41)	90	2418	2418	00.037
43	4t+3	A-2 Rotational	2462	2822	2822	00.872
44	4t	(Derived from 43)	2462	3302	3302	00.746
45	4t+1		3268	3851	3851	00.849
		1 Rotational	1406		2547	00.552
		A-2 Rotational	1862		1304	01.428
46	4t+2	(Derived from 45)	1406	4488	4488	00.313
47	4t+3	A-2 Rotational	6348	5215	5215	01.217
48	4t	(Derived from 47)	6348	6072	6072	01.045
49	4t+1		5587	7033	7033	00.794
		1 Rotational	460		4417	00.104
		A-2 Rotational	5127		2616	01.960

For each order *n* of a complete graph, we solve the *OP* on all the configurations given by the *integer partitions* of *n*.

Table of results

50	4t+2	(Derived from 49)	460	8158	8158	00.056
51	4t+3	A-2 Rotational	16705	9441	9441	01.769
52	4t	(Derived from 51)	16705	10920	10920	01.530
53	4t+1		18998	12600	12600	01.508
		1 Rotational	4246		7513	00.565
		A-2 Rotational	14752		5087	02.900
54	4t+2	(Derived from 53)	4246	14552	14552	00.292
55	4t+3	A-2 Rotational	57043	16753	16753	03.405
56	4t	(Derived from 55)	57043	19296	19296	02.956
57	4t+1		42700	22183	22183	01.925
		1 Rotational	2519		12557	00.201
		A-2 Rotational	40181		9626	04.174
58	4t+2	(Derived from 57)	2519	25491	25491	00.099
59	4t+3	A-2 Rotational	105258	29241	29241	03.600
60	4t	(Derived from 59)	105258	33552	33552	03.137

Table 1: Computational results for the OP with $n \in [40, 60]$, with more than 3 cycles per instance

Theoretical results

- We presented a new theorem holding on the existence of *1-rotational solution,* which was suggested by the computational evidence.
- We proposed a polynomial-time approach to solve a restricted labelling problem for *2-rotational* methods.
- Computational proof for *OP* (^{2}*3, 5)*

Theorem. Let F be a *2-regular* graph of order *2n + 1* and let *r* denote the number of cycles in *F* of even length. If *F* satisfies the assumptions of *Proposition 1* and its cycle passing through ∞ has length *3*, then either $n \equiv 0 \pmod 4$ or $n-1+r$ is an even integer.

$$\exists! i : (\ell_i = 3 \wedge u_i \text{ is odd}) \Rightarrow$$
$$2t \equiv 0 \pmod 4 \quad \vee$$
$$\left(\frac{2t-1}{2} + \sum_k w_i \right) \equiv 0 \pmod 2$$

Nutshelly references

F Salassa, G Dragotto, T Traetta, M Buratti, F Della Croce. Merging Combinatorial Design and Optimization: the Oberwolfach Problem. [Submitted] [pre-print on arxiv].

W. L. Piotrowski. The solution of the bipartite analogue of the Oberwolfach problem. Discrete Mathematics, 1991. ISSN 0012365X. doi: 10.1016/0012-365X(91)90449-C.

Wiktor L. Piotrowski. Untersuchungen uber das oberwolfacher problem. unpublished, 1979.

A Deza, F Franek, W Hua, M Meszka, and A Rosa. Solutions to the Oberwolfach problem for orders 18 to 40. Journal of Combinatorial Mathematics and Combinatorial Computing, 2010. ISSN 08353026.

C J Colbourn and Jeffrey H Dinitz. The CRC handbook of combinatorial designs. CRC Press series on discrete mathematics and its applications., 2007.

Brian Alspach, Paul J Schellenberg, Douglas R Stinson, and David Wagner. The oberwolfach problem and factors of uniform odd length cycles. Journal of Combinatorial Theory, Series A, 52(1):20–43, 1989.

Thanks!

This keynote is available at:
www.dragotto.net

Brainstorming Time