Where the Really Hard Quadratic Assignment Problems Are: the QAP-SAT instances

Sébastien Verel, Sarah Thomson, and Omar Rifki

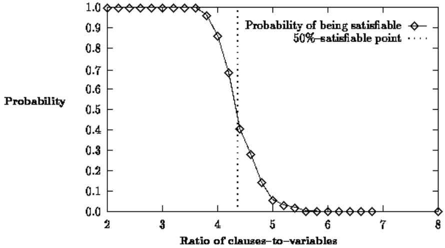
Laboratoire d'Informatique, Signal et Image de la Côte d'opale (LISIC)
Université du Littoral Côte d'Opale, Calais, France
Edinburgh Napier University, Scotland, United Kingdom
http://www-lisic.univ-littoral.fr/~verel/

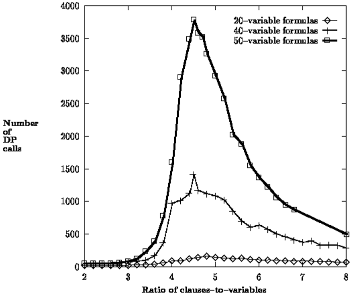
EvoCOP conference,
April, 5, 2024

Phase transition, and solvers : SAT problems

"Where the Really Hard Problems Are", Cheeseman *et al.*, IJCAI-91

Mitchell, Selman & Levesque, AAAI-92





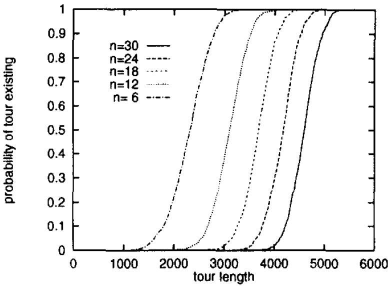
Phase transition in decision problems :

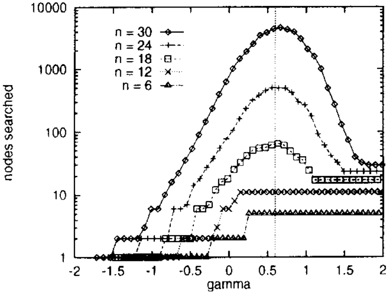
- Satisfiability drops quickly around a phase parameter transition
- Link to optimization difficulty

For rnd. SAT, ratio clause-to-variable. For 3-SAT, $\alpha = m/n \approx 4.3$

Phase transition : TSP problem

TSP phase transition, Gent & Walsh, AI, 1996





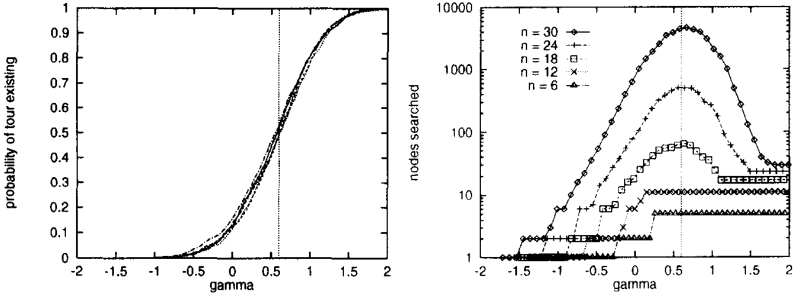
Phase transition in optimization problems : $\Pr(\exists \sigma : f_{TSP}(\sigma) \leq \ell)$

- Satisfiability drops quickly around a phase parameter transition
- Link to optimization difficulty

For random TSP, n cities in area A, $\gamma = (\frac{\ell}{\sqrt{nA}} - 0.78)n^{1/1.5}$

Phase transition : TSP problem

TSP phase transition, Gent & Walsh, AI, 1996



Phase transition in optimization problems : $\Pr(\exists \sigma : f_{TSP}(\sigma) \leq \ell)$

- Satisfiability drops quickly around a phase parameter transition
- Link to optimization difficulty

For random TSP, n cities in area A, $\gamma = (\frac{\ell}{\sqrt{nA}} - 0.78)n^{1/1.5}$

Objectives of this work

Quadratic Assignement Problem (QAP)

- Well-known problem in evolutionary computation
- Very challenging problem

Goals

- Show a phase transition in "pure" QAP
  First phase transition to our best knowledge
- Benchmark of easy/difficult instances
- Propose a design principle to better understand difficulty in QAP

Definition : QAP [Koopmans, 1957]

Assignment problem (minimization), quadratic costs :

$$\forall \sigma \in \mathcal{S}_n,\ \ Q_{A,B}(\sigma) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} B_{\sigma_i \sigma_j} = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} B_{ij}^{\sigma}$$

$\mathcal{S}_n$: Search space, symmetric group of dim. n (permutations)
A, and B matrices $n \times n$,
A_{ij} "flow" (cost) between objects i, and j,
B_{ij} "distance" (cost) between positions i, and j

Here, $A_{ij} \geq 0$, and $B_{ij} \geq 0$
B not necessary distance matrix, not necessary symmetric,
but $B_{ii} = 0$, and $A_{ii} = 0$.

QAP benchmark instances

- Many applications in real-world [3]
- Many benchmark instances :
  to understand difficulty, and design better algorithm

QAPLib [1]

Collection of real-world small size, artificial larger ones

- Taillard instances (*Taia*, and *Taib*)
- *Taie* and *Dre* instances : difficult for metaheuristics [5]
- Stützle *et al.* instances : flow dominance and sparsity [12]

Problems difficulty in QAP

Matrices features

Flow dominance [14] : imbalance in matrices ("variance" in matrix)
Sparsity [11] : number of zero-entries as a proportion of the n^2

Fitness landscape features

Correlation length, fitness-distance correlation [9]
Information metrics estimate with random walks [10]
Autocorrelation and plateaus size $\sim$ number of similar values [13]
Local Optima Network [4]

Fourier features

B&Bound which operates in the Fourier space [8]
Elementary landscape decomposition [2] Fourier decomposition [6]

Problems difficulty in QAP

Matrices features

Flow dominance [14] : imbalance in matrices ("variance" in matrix)
Sparsity [11] : number of zero-entries as a proportion of the n^2

Fitness landscape features

Correlation length, fitness-distance correlation [9]
Information metrics estimate with random walks [10]
Autocorrelation and plateaus size $\sim$ number of similar values [13]
Local Optima Network [4]

Fourier features

B&Bound which operates in the Fourier space [8]
Elementary landscape decomposition [2] Fourier decomposition [6]

Surprisingly, features are tuned for each matrix independently

From SAT to QAP-SAT

Phase transition in rnd. SAT/MAX-SAT

$f(x) = \sum_{i=1}^{m} c_i(x_{i_1}, x_{i_2}, x_{i_3})$: sum of clauses *i.e.* low dim. problems

One clause is satisfied when $c(x_{i_1}, x_{i_2}, x_{i_3}) = 1$, the upper bound

When the num. of clauses increases, transition to unsatisfiability

Difficulties with QAP

- QAP space is **not** a vector space. Bi-linear property :

$$\begin{aligned} Q_{A+A',B+B'} &= Q_{A,B} + Q_{A,B'} + Q_{A',B} + Q_{A',B'} \\ \text{but,} \quad Q_{A+A',B} &= Q_{A,B} + Q_{A',B} \end{aligned}$$

- Sub-spaces of $\mathcal{S}_n$:

  Subspaces are **not** isomorphic to $\mathcal{S}_3$, given by (i_1, i_2, i_3),
  but, depend on other values/objects $\{1, \ldots, n\}$

Design components : A-clauses and B-clauses

A-clause and B-clause of size $k > 0$

A-clause : $\exists V_A \subset [n]$ of size k s. t. : $\forall i \in [n]\ A_{ii} = 0$, $\forall (i,j) \in V_A^2,\ i \neq j,\ A_{ij} > 0$, and $\forall (i,j) \notin V_A^2,\ A_{ij} = 0$.

B-clause : $\exists V_B \subset [n]$ of size k s. t. : $\forall i \in [n]\ B_{ii} = 0$, $\forall (i,j) \in V_B^2,\ i \neq j,\ B_{ij} = 1$, and $\forall (i,j) \notin V_B^2,\ B_{ij} = \textsc{m}$.

$V_A = \{2, 3, 5\}$

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} \qquad A^{(3)} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix}$$

$V_B = \{1, 2, 5\}$

$$B_3 = \begin{bmatrix} 0 & 1 & \textsc{m} & \textsc{m} & 1 \\ 1 & 0 & \textsc{m} & \textsc{m} & 1 \\ \textsc{m} & \textsc{m} & 0 & \textsc{m} & \textsc{m} \\ \textsc{m} & \textsc{m} & \textsc{m} & 0 & \textsc{m} \\ 1 & 1 & \textsc{m} & \textsc{m} & 0 \end{bmatrix} \qquad B^{(3)} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Design components : aggregation of clauses

A is composed of m A-clauses when :

$A = A_1 + \ldots + A_m$ with $A_1, \ldots, A_m$ A-clauses

nota : $Q_{A_i,B}$ is a clause

B is composed of m_1 B-clauses when :

$B = B_1 \odot \ldots \odot B_{m_1} \odot C$ with $B_1, \ldots, B_{m_1}$ B-clauses,

$C_{ij} > 1$, $C_{ii} = 0$. $\odot$ minimum element by element

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 3 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 1 & 3 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 3 & 4 & 1 & 0 \end{bmatrix}$$

Design components : aggregation of clauses

A is composed of m A-clauses when :

$A = A_1 + \ldots + A_m$ with $A_1, \ldots, A_m$ A-clauses

nota : $Q_{A_i,B}$ is a clause

B is composed of m_1 B-clauses when :

$B = B_1 \odot \ldots \odot B_{m_1} \odot C$ with $B_1, \ldots, B_{m_1}$ B-clauses,

$C_{ij} > 1,\ C_{ii} = 0$. $\odot$ minimum element by element

$$B = \begin{bmatrix} 0 & 1 & M & M & 1 \\ 1 & 0 & M & M & 1 \\ M & M & 0 & M & M \\ M & M & M & 0 & M \\ 1 & 1 & M & M & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & 1 & 1 & M & M \\ 1 & 0 & 1 & M & M \\ 1 & 1 & 0 & M & M \\ M & M & M & 0 & M \\ M & M & M & M & 0 \end{bmatrix} \odot C = \begin{bmatrix} 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 0 & 3 & 2 \\ 3 & 2 & 2 & 0 & 5 \\ 1 & 1 & 2 & 4 & 0 \end{bmatrix}$$

Design components : aggregation of clauses

A is composed of m A-clauses when :

$A = A_1 + \ldots + A_m$ with $A_1, \ldots, A_m$ A-clauses

nota : $Q_{A_i,B}$ is a clause

B is composed of m_1 B-clauses when :

$B = B_1 \odot \ldots \odot B_{m_1} \odot C$ with $B_1, \ldots, B_{m_1}$ B-clauses,

$C_{ij} > 1$, $C_{ii} = 0$. $\odot$ minimum element by element

QAP-SAT

$Q_{A,B}$ is a *QAP-SAT* with m A-clauses and m_1 B-clauses when :

A is composed of m A-clauses, and

B is composed of m_1 B-clauses

Satisfiability

Lower bound of single clauses (A-clause, B-clause) :

$$Q_{A^{(3)},B^{(3)}}(\sigma) = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix} . \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = 10$$

Satisfiability

Lower bound of single clauses (A-clause, B-clause) :

$$Q_{A_3,B_3}(\sigma = (13)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 & 1 & \text{M} & \text{M} & 1 \\ 1 & 0 & \text{M} & \text{M} & 1 \\ \text{M} & \text{M} & 0 & \text{M} & \text{M} \\ \text{M} & \text{M} & \text{M} & 0 & \text{M} \\ 1 & 1 & \text{M} & \text{M} & 0 \end{bmatrix}^{\sigma}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 & \text{M} & \text{M} & \text{M} & \text{M} \\ \text{M} & 0 & 1 & \text{M} & 1 \\ \text{M} & 1 & 0 & \text{M} & 1 \\ \text{M} & \text{M} & \text{M} & 0 & \text{M} \\ \text{M} & 1 & 1 & \text{M} & 0 \end{bmatrix} = 10$$

Indeed any σ s.t. $\sigma(\{2,3,5\}) = \{1,2,5\}$, is an optimal solution

Satisfiability

Lower bound of single clauses (A-clause, B-clause) :

$$Q_{A_3,B_3}(\sigma = (13)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 & 1 & M & M & 1 \\ 1 & 0 & M & M & 1 \\ M & M & 0 & M & M \\ M & M & M & 0 & M \\ 1 & 1 & M & M & 0 \end{bmatrix}^{\sigma}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 & M & M & M & M \\ M & 0 & 1 & M & 1 \\ M & 1 & 0 & M & 1 \\ M & M & M & 0 & M \\ M & 1 & 1 & M & 0 \end{bmatrix} = 10$$

Clause $Q_{A_i,B}$ satisfied when the lb is reached : $\exists\sigma \; Q_{A_i,B}(\sigma) = \text{lb}(A_i)$

$Q_{A,B}$ satisfied when all clauses are satisfied : $\exists\sigma \; Q_{A,B}(\sigma) = m \; \text{lb}(A_i)$

Example of satisfiability : $m = 2$, and $m_1 = 2$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 3 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 1 & 3 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 3 & 4 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & M & M & 1 \\ 1 & 0 & M & M & 1 \\ M & M & 0 & M & M \\ M & M & M & 0 & M \\ 1 & 1 & M & M & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & 1 & 1 & M & M \\ 1 & 0 & 1 & M & M \\ 1 & 1 & 0 & M & M \\ M & M & M & 0 & M \\ M & M & M & M & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & M & 1 \\ 1 & 0 & 1 & M & 1 \\ 1 & 1 & 0 & M & M \\ M & M & M & 0 & M \\ 1 & 1 & M & M & 0 \end{bmatrix}$$

$$B^\sigma = \begin{bmatrix} 0 & M & M & M & M \\ M & 0 & 1 & M & 1 \\ M & 1 & 0 & M & 1 \\ M & M & M & 0 & M \\ M & 1 & 1 & M & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & M & M & M & M \\ M & 0 & M & M & M \\ M & M & 0 & 1 & 1 \\ M & M & 1 & 0 & 1 \\ M & M & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & M & M & M & M \\ M & 0 & 1 & 1 & 1 \\ M & 1 & 0 & 1 & 1 \\ M & 1 & 1 & 0 & 1 \\ M & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 2 & 3 & 1 \end{pmatrix} \quad Q_{A,B}(\sigma) = 20, \; Q_{A,B} \text{ is satisfiable}$$

Random QAP-SAT instances

QAP-k-SAT

All clauses have the same size k

For each clause (A-clauses, and B-clauses),
- Select randomly and independently k different variables
- Use $A_{(3)}$, and $B_{(3)}$ to complete the clause indexed by the var.

Complete matrix B values $d > 1$ s. t. proportions follow $p_d = p_1^d$

$V_A = \{2, 3, 5\}$, $V_B = \{1, 2, 5\}$

$$A^{(3)} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix} \quad B^{(3)} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Python code (generator), instances, data :

`https://gitlab.com/verel/qap-sat`

Experimental setup

Instance generation

	Description	Values
n	Problem dimension	$\{8, 9, \ldots, 17\}$ $\{18, 19\}$
k	Size of clause	3
m_1	Num. of B-clauses	$\{3, 6, 9, \ldots, 27\}$ $\{3, 9, 15, \ldots, 57\}$
m	Num. of A-clauses	$\{1, 2, 3, \ldots, 40\}$ $\{1, 3, 9, 15, \ldots, 57, 63\}$

50 instances for each parameter triplet (n, m_1, m)

Branch & Bound algorithm by Fujii *et al.* [7]

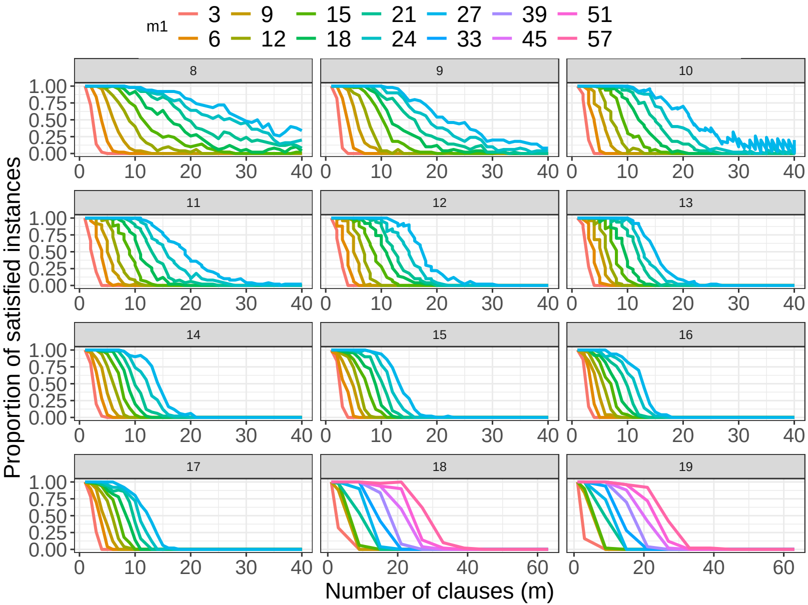
Lagrangian doubly non-negative relaxation and Newton-bracketing MATLAB code available.

Notice that : full enumeration is possible for $n \leq 13$

Tabu search

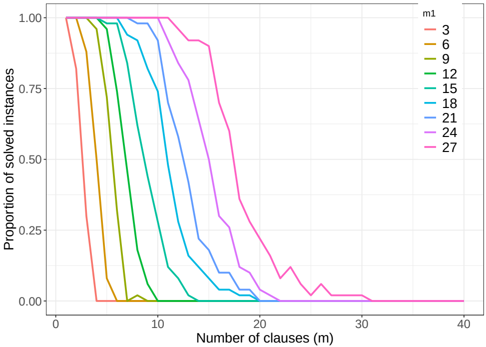
Baseline "classical" Robust Tabu Search of Taillard

Proportion of satisfiable instances



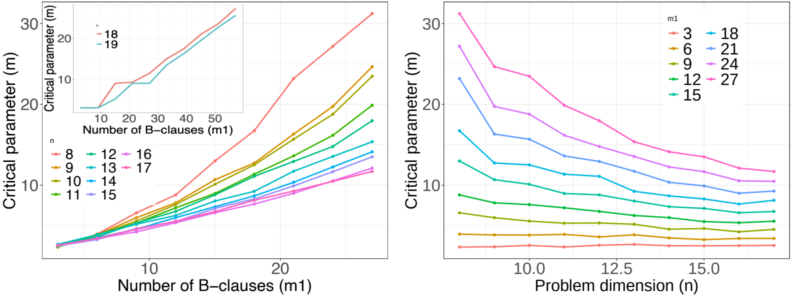
Proportion of satisfiable instances

$n = 12$



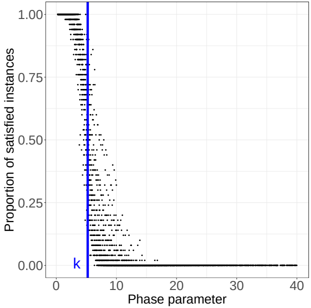
- When m_1 is fixed, fast drop "around" $m \approx m_1$
- Same shape for every problem dim. n
- Faster drop when n is larger

Phase transition parameter



- Critical parameter m_c of logistic model, estimated by logit regression (high R^2 values > 0.9)
- Regression of m_1 when n is given : $m_c = \beta_0 + \beta_1 m_1 + \epsilon$
R^2 over 0.97 Slope β_1 decreases with n

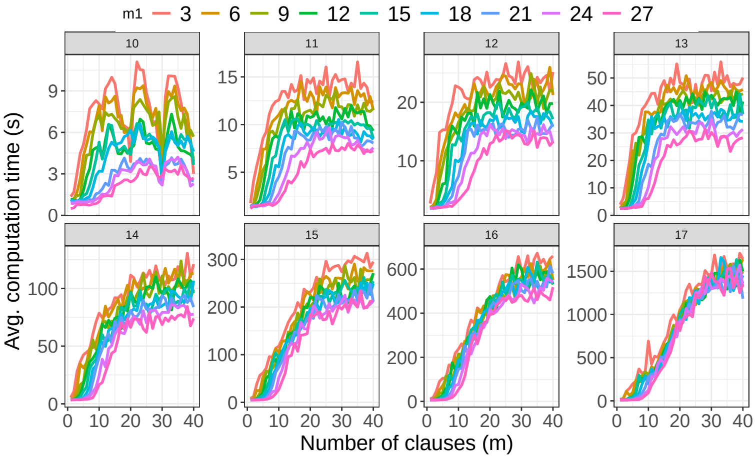
Phase transition parameter



$$m_c = k n^{\alpha_1} m_1^{\alpha_2} + \epsilon$$

Adding a scaling factor n^{α_1} (such as TSP)
Estimation using $\log(m_c)$, $R^2 = 0.947$ ($R^2 = 0.898$ without log)
$\alpha_1 = -0.75999$: negative $\in [-1/\sqrt{n}$, and $-1/n]$
$\alpha_2 = 0.90365$: close to 1. $\log(k) = 1.65453$
Hypothesise on phase trans. param. : $m \; n^{-\alpha_1} m_1^{-\alpha_2}$

B & B computation time

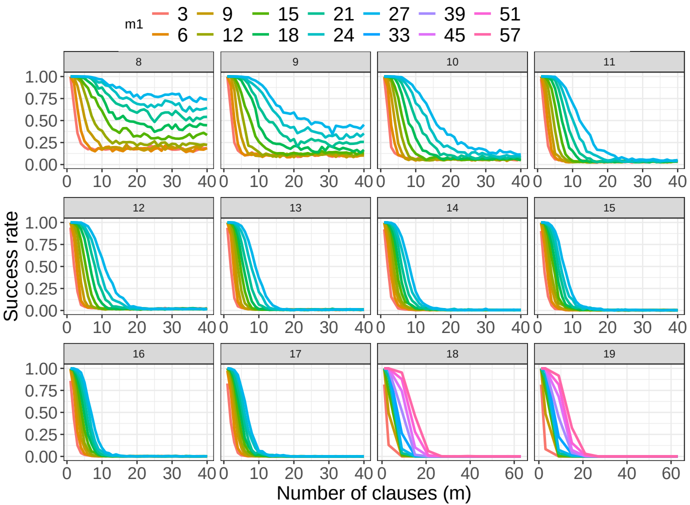


Sigmoid regression model : $t(m) = \frac{L}{1+e^{-r(m-m_t)}}$

$L \approx \gamma(2.043 + 0.476(n-8))$ max. value, r rate, and m_t inflexion

Median value regressions of $R^2 = 0.969$

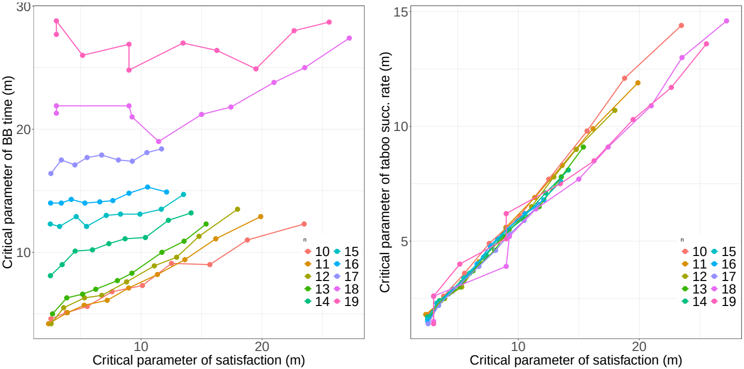
Tabu search success rate



Analysis using sigmoid regression model :

High regression quality again, follow the shape of m_c

Critical parameter m_c *v.s.* B & B, and Tabu critical param.



High linear correlation between m_c, and critical param. of algorithm
Correlation for tabu search : seems not depend of problem dim. n

Conclusions

Summary

- Propose new QAP benchmark :
  Difficulty related to the link between A, and B matrices
- Show phase transition across the instances
- Hypothesis of phase transition parameter model

Perspectives

- Large instances
- Compare QAP-SAT with QAPLib,
  decompose real-world instances into "clauses"
- Fitness landscape analysis, theoretical investigation
  QAP-2-SAT (graph matching), etc.
- Different k, clauses, relax the satisfiability condition, etc.

$$A^{(3)} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & ? & ? \\ ? & 0 & ? \\ ? & ? & 0 \end{bmatrix} \quad B^{(3)} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

Rainer E Burkard, Stefan E Karisch, and Franz Rendl.
Qaplib–a quadratic assignment problem library.
Journal of Global optimization, 10 :391–403, 1997.

Francisco Chicano, Gabriel Luque, and Enrique Alba.
Elementary landscape decomposition of the quadratic assignment problem.
In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 1425–1432, 2010.

Clayton W Commander.
A survey of the quadratic assignment problem, with applications.
2005.

Fabio Daolio, Sébastien Verel, Gabriela Ochoa, and Marco Tomassini.
Local optima networks of the quadratic assignment problem.
In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.

Zvi Drezner, Peter M Hahn, and Éeric D Taillard.
Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods.
Annals of Operations research, 139 :65–94, 2005.

Anne Elorza, Leticia Hernando, and Jose A Lozano.
Characterizing permutation-based combinatorial optimization problems in fourier space.
Evolutionary Computation, pages 1–39, 2022.

Koichi Fujii, Naoki Ito, Sunyoung Kim, Masakazu Kojima, Yuji Shinano, and Kim-Chuan Toh.
Solving challenging large scale qaps.
arXiv preprint arXiv :2101.09629, 2021.

Risi Kondor.
A fourier space algorithm for solving quadratic assignment problems.

In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete algorithms*, pages 1017–1028. SIAM, 2010.

Peter Merz and Bernd Freisleben.
Fitness landscape analysis and memetic algorithms for the quadratic assignment problem.
IEEE transactions on evolutionary computation, 4(4) :337–352, 2000.

Erik Pitzer, Andreas Beham, and Michael Affenzeller.
Automatic algorithm selection for the quadratic assignment problem using fitness landscape analysis.
In *Evolutionary Computation in Combinatorial Optimization : 13th European Conference, EvoCOP 2013, Vienna, Austria, April 3-5, 2013. Proceedings 13*, pages 109–120. Springer, 2013.

Kate A Smith-Miles.

Towards insightful algorithm selection for optimisation using meta-learning concepts.
In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 4118–4124. ieee, 2008.

Thomas Stützle and Susana Fernandes.
New benchmark instances for the qap and the experimental analysis of algorithms.
In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 199–209. Springer, 2004.

Mohammad-H Tayarani-N and Adam Prügel-Bennett.
Quadratic assignment problem : a landscape analysis.
Evolutionary Intelligence, 8 :165–184, 2015.

Thomas E Vollmann and Elwood S Buffa.
The facilities layout problem in perspective.
Management Science, 12(10) :B–450, 1966.