# A Continuous Variable Optimization method for the Quadratic Assignment Problem

**author names withheld**

**Under Review for OPT 2024**

## Abstract

We present a novel continuous algorithm for solving the Quadratic Assignment Problem (QAP), leveraging auxiliary dynamics to enforce permutation constraints without the need for post-processing. This approach outperforms traditional continuous methods in terms of constraint enforcement and demonstrates faster convergence compared to branch-and-bound techniques. Despite the algorithm's effectiveness, the number of auxiliary variables currently scales cubically with the problem size, posing a limitation. However, our analysis suggests that associating auxiliary variables with correlators of clause functions could significantly improve efficiency. Additionally, the algorithm encounters challenges with local minima due to the geodesically non-convex QAP potential. We propose future research directions to address this issue, including alternative formulations of the potential landscape and strategies for escaping local minima.

**Keywords:** Non-convex optimization, Manifold optimization, Dynamical systems, Differential equations, Combinatorial Optimization, Chaos theory, Boolean Satisfyability, SAT, Quadratic Assignment problem, QAP, Traveling Salesmen problem, TSP, Continuous variable computing

## 1. Introduction

Combinatorial optimization is crucial for many real-world applications but is hindered by the exponential time complexity of state-of-the-art methods. As problem size increases, even advanced algorithms become computationally infeasible. While exact methods like branch and bound and heuristics such as Tabu search [4], genetic algorithms [3], and simulated annealing [7] have been developed, they also face scalability challenges. This work presents a continuous algorithm addressing one of the toughest combinatorial optimization challenges: the Quadratic Assignment Problem (QAP), a generalization of the Travelling Salesman Problem (TSP). By employing a continuous representation of the discrete problem, we harness analog variable computing to enhance scalability.

### 1.1. Continuous Variable Machines

Digital computing advancements are nearing a standstill due to physical limits on transistor size. Simultaneously, little progress has been made in efficiently solving classes of combinatorial constraint satisfaction and optimization problems. Consequently, alternative computational paradigms are emerging. Continuous variable algorithms may outperform digital systems in specific domains, with recent analog computing devices solving problems significantly faster than conventional methods [1, 5].However, benchmarking these continuous algorithms poses a unique challenge: unless a dedicated physical machine is built, the only way to verify their performance is through simulation on digital machines. This adds a significant constant and a polynomial overhead to the time complexity. Nevertheless, continuous variable machines hold potential for overcoming current computational bottlenecks in combinatorial optimization.

### 1.2. Problem Formulation

This work focuses on the Quadratic Assignment Problem (QAP), involving two sets: $N$ facilities and $N$ locations. Each location has a distance specified, and each facility has a weight or flow, indicating goods transported between facilities. The goal is to assign facilities to locations to minimize total cost, calculated as the sum of the products of flows and distances.

Mathematically, we aim to minimize:

$$\min_{P\in\mathcal{P}_n} \underbrace{\operatorname{tr}\left(A^T P^T B P + P^T C\right)}_{\equiv V_{\mathrm{QAP}}(P)}, \tag{1}$$

where $A$, $B$, and $C$ are constant $N \times N$ matrices. The constraint ensures $P$ is a permutation matrix, with the special case $C = 0$ reducing the QAP to the TSP.

A continuous variable solver is defined by the dynamical system:

$$\frac{d}{dt}P = f(P, t; A, B, C, \dots),$$

where $f$ is constructed to ensure fixed points correspond to solutions defined by matrices $A$, $B$, and $C$.

### 1.3. Related Work

As continuous variable computation gains traction, efforts to address QAP-like optimization problems have emerged. Adapting the discrete nature of permutation groups to continuous systems

poses challenges. One approach is to relax the solution space to $N \times N$ matrices, broadening the search space but introducing inefficiencies. Alternatively, restricting optimization to the orthogonal manifold $\mathcal{O}_N$ has been explored using Riemannian gradient descent, but this method struggles with convergence due to increased geodesic non-convexity and local minima.

Restricting dynamics to the convex hull of permutation matrices (the Birkhoff polytope) maintains combinatorial structure while benefiting from convexity, making optimization potentially more tractable.

## 2. Methodology

Our approach was to enforce the permutation constraint by reformulating the it as a boolean satisfyability problem and employing the dynamical system from [2] in addition to the gradient on the original objective function.

### 2.1. The SAT problem

Boolean SAT is a family of constraint satisfaction problems in which we are given a set of Boolean[1] variables $x_i \in \mathbb{B}$, where $i \in [N]$, and a set of constraints in the form of a logical expression (also called "formula"):

$$F(x_1, x_2, \ldots, x_N) = \bigwedge_{m=1}^{M} C_m = C_1 \wedge C_2 \wedge \cdots \wedge C_M,$$

where $\wedge$ is the logical *and* operation and $C_m$ is called the $m^{\text{th}}$ *clause*. Each clause consists of *literals* concatenated by $\vee$, the logical *or* operator: $C_m = \nu_{m_1} \vee \cdots \vee \nu_{m_k}$. A literal $\nu_i$ is a variable $x_i$ or its negation $\overline{x}_i$. The decision problem is to determine whether there is a configuration of the logical variables such that the constraint is satisfied. A convenient way to encode a particular problem instance (a particular choice of $\nu_i$) is to define a matrix

$$c_{im} = \begin{cases} 1 & \text{if } x_i \in C_m, \\ -1 & \text{if } \overline{x}_i \in C_m, \\ 0 & \text{otherwise.} \end{cases}$$

The state space of a SAT problem with $N$ variables is identical to the vertex set of the hypercube graph $\mathbf{Q}_N$, in which each vertex is identified by a string of $N$ bits. The chief ingredient that makes SAT problems hard is the exponential size of the state space (the configuration space $\mathbf{Q}_N$ has $2^N$ vertices).

### 2.2. Continuous satisfyability solver

The idea behind the algorithm defined in [2], is to break up the problem along the clauses, and define an objective function -called clause function- for each:

$$K_m = 2^{-k_m} \prod_{j=1}^{N} (1 - c_{mj} s_j(t)) \in [0, 1] \ . \tag{2}$$

1. The set is defind as $\mathbb{B} \equiv \{0, 1\}$ or FALSE/TRUE

Here $s_i$ is called a soft-spin, the embedded or relaxed continuous variable assotiated to the boolean variable $x_i$, and it can take values from $-1$ to $+1$. The clause function is a continuous relaxation of the boolean observable of the clause, it is one (true) if the function is zero, and one otherwise. To solve a SAT problem, all of these function have to be true at the same time, corresponding to the the minimum of the sum of some monotonically increasing function of these clause functions. If we were to define a gradient descent on such a potential, it would be guaranteed that the minima of that function corresponds to solution to the SAT problem. The problem is that this function is non-convex, and so such dynamics are bound to get stuck in local minima. To overcome this hurdle, non-local information has to be incorporated into the algorithm. This comes in the form of additional auxiliary variables that contain information about the previous evolution of the system: The SAT problem becomes a minimization problem of the potential

$$V_{\text{SAT}}(s,a) = \sum_m^M a_m K_m^2 = \sum_m^M K_m^2 \underbrace{\exp \int_0^t K_m^2(\tau) d\tau}_{=a_m(t)} = V_{\text{SAT}}\left[s(\tau \leq t)\right].$$

with the dynamics defined as:

$$\begin{aligned} \frac{ds_i}{dt} &= -\partial_{s_i} V_{\text{SAT}}(s,a)\,, \\ \frac{da_m}{dt} &= a_m K_m^2\,. \end{aligned}$$

It is proved in [2] that the only fixed points if this dynamical system are those that correspond to a valid solution to the SAT problem.

### 2.3. Reformulating the constraint

Per definition a permutation matrix contains one and only one number 1 in each of its rows and columns. The logical function $F$ that the elements of a permutation matrix have to satisfy is thus:

$$F(P_{11}, P_{12}, \ldots, P_{NN}) = \left( \bigwedge_{i=1}^N \bigvee_{j=1}^N \left( P_{ij} \wedge \bigwedge_{k \neq j} \neg P_{ik} \right) \right) \wedge \left( \bigwedge_{j=1}^N \bigvee_{i=1}^N \left( P_{ij} \wedge \bigwedge_{k \neq i} \neg P_{kj} \right) \right)\,.$$

Which can be formulated as a SAT problem in conjunctive normal form:

$$F(P) = \left( \bigvee_{i=1}^N P_{ij} \right) \wedge \left( \bigwedge_{j \neq k} (\neg P_{ij} \vee \neg P_{kj}) \right)\,. \tag{3}$$

Note that the least amount of clauses in this formulation is $M = 2N\left(1 + N\frac{(N-1)}{2}\right)$

### 2.4. Proposed algorithm

The Eucledian gradient on the relaxed (soft-spin) matrix elements:

$$\nabla V_{\text{QAP}}(P) = BPA + B^T PA + C.$$

However, this gradient is a continuous vector function defined over the whole eucledian embedding space, and does not respect any constraints. To enforce the permutation constraints we supplement the gradient descent by the SAT potential described in the previous section.

$$\frac{d}{dt}P_{ij} = -\beta\partial_{P_{ij}}V_{\text{QAP}}(P) - \partial_{P_{ij}}V_{\text{SAT}}(P,a) \equiv f(P,a)\,, \tag{4}$$

$$\frac{d}{dt}a_m = a_m K_m^2\,. \tag{5}$$

The two terms in the formulation have different long term behavior. The QAP gradient is always some polynomial in the elements of $P$ while the SAT potential behaves exponentially due to equation 5. This means, that eventually the constraint term will outgrow the QAP gradient and the dynamics will converge to a permutation matrix. The analog time until the dynamics converge to a candidate solution state (aTTS) mainly depends on the SAT dynamics, and the value of the parameter $\beta$.

The SAT dynamics approximately constrain the dynamics to the interior of the hypercube $\mathcal{H}_{N^2} = [-1,+1]^{\times N^2}$, but it might be worthwhile to restrict the dynamics to the embedded manifold of orthogonal matrices $\mathcal{O}(N)$. This can be achieved by projecting the vector-field $f$ from equation 4, down to $\mathcal{O}(N)$ via the generalized Lie bracket

$$\tilde{f}(P) = P\left\{f(P), P\right\}\,.$$

## 3. Simulations and Results

The algorithm was tested on a randomly generated dataset using the DifferentialEquations module of the Julia programming language and compared against a conventional optimizer, SCIP. The matrix formulation poses a significant challenge for both methods, even for smaller problems with $N = 10$, primarily due to the nature of the constraints. While digital methods that directly handle permutations (such as 2-opt heuristics) or brute-force techniques can solve problems of this size, they do not offer convergence guarantees. These simulations are intended as a proof of principle.

Due to the exponential enforcement of the constraints, the analog solver's time-to-solution remains constant as a function of problem size. However, when simulating the system on a digital machine, larger problems require increased memory allocation, which slows down the integration process. The simulation results suggest that the evolution of the solution gap follows a sub-power-law trend, in contrast to the behavior observed with branch-and-bound methods (see Fig. 2).

Simulations suggest, that restricting the dynamics to the embedded orthogonal manifold has both advantages and drawbacks. Operating on the manifold potentially allows a drastic reduction in the number of SAT constraints, as permutation matrices are the only orthogonal matrices with only non-negative entries. However, the potential described by equation 1 is geodesically non-convex, meaning the restriction introduces new local minima. As a result, the Riemannian gradient often gets trapped in local optima.

## 4. Conclusion and Future Work

We have demonstrated that the auxiliary dynamics effectively enforce the permutation constraint, representing a notable improvement over previous continuous approaches that either did not enforce

these constraints [6] or required post-processing. Additionally, we observed that the convergence of the gap is faster than with branch-and-bound methods.

A significant drawback of the current SAT constraints is the number of auxiliary variables, which scales cubically with the number of sites. However, the evolution of these auxiliary variables suggests that the SAT problem itself is relatively simple. This observation implies that the number of auxiliary variables could potentially be reduced, improving efficiency. One possible approach is to associate auxiliary variables with correlators of clause functions, rather than assigning them to each individual clause function.

In its current form, the algorithm does not address scenarios where the dynamics become trapped in local minima due to the geodesically non-convex nature of the QAP potential. Further research is needed to mitigate this issue, potentially by developing techniques designed to escape local minima or through alternative formulations of the potential landscape.

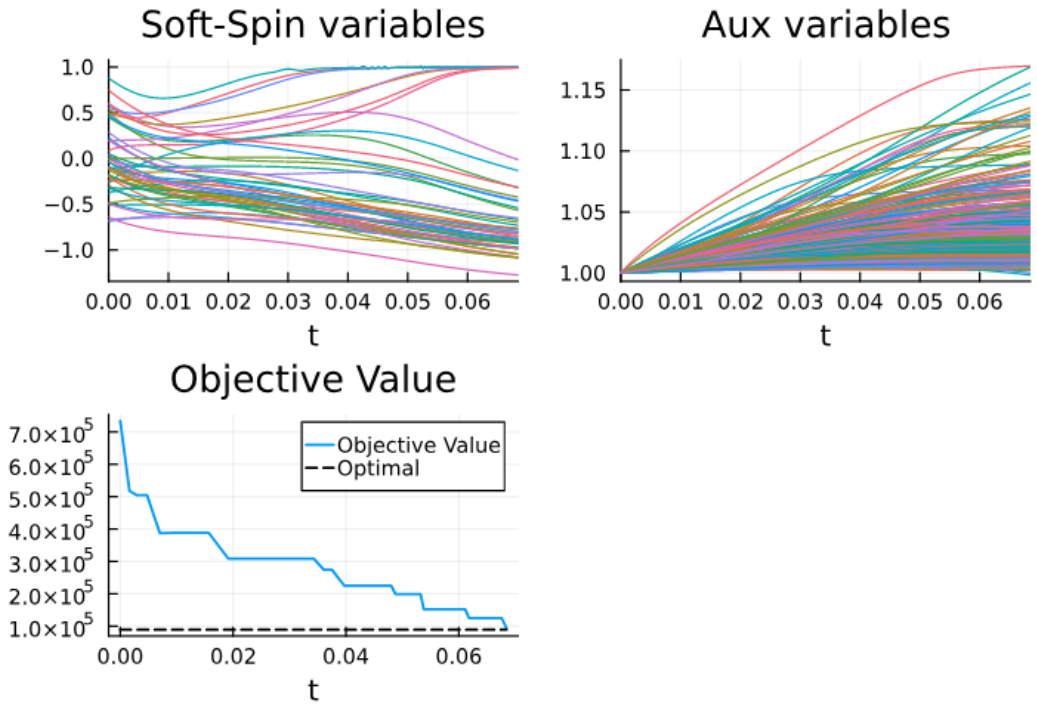


Figure 1: An example simulation of an $N = 7$ system finding a solution.

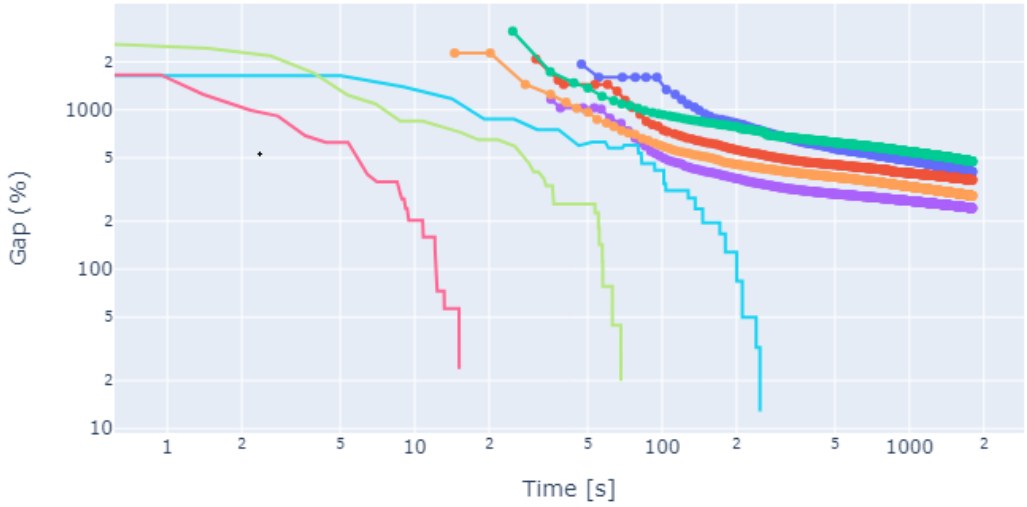


Figure 2: The evolution of the gap as a function of wall-clock time. Solid lines correspond to analog solver, dotted line correspond to branch and bound. For $N = 10$ sized problems.

## References

[1] Muya Chang, Xunzhao Yin, Zoltan Toroczkai, Xiaobo Hu, and Arijit Raychowdhury. An analog clock-free compute fabric base on continuous-time dynamical system for solving combinatorial

optimization problems. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, April 2022. doi: 10.1109/cicc53496.2022.9772850. URL https://doi.org/10.1109/cicc53496.2022.9772850.

[2] Mária Ercsey-Ravasz and Zoltán Toroczkai. Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nature Physics*, 7(12):966–970, October 2011. doi: 10.1038/nphys2105. URL https://doi.org/10.1038/nphys2105.

[3] Charles Fleurent, Jacques A Ferland, et al. Genetic hybrids for the quadratic assignment problem. *Quadratic assignment and related problems*, 16:173–187, 1993.

[4] Alfonsas Misevicius. An implementation of the iterated tabu search algorithm for the quadratic assignment problem. *OR spectrum*, 34(3):665–690, 2012.

[5] Edwin Ng, Tatsuhiro Onodera, Satoshi Kako, Peter L. McMahon, Hideo Mabuchi, and Yoshihisa Yamamoto. Efficient sampling of ground and low-energy ising spin configurations with a coherent ising machine. *Physical Review Research*, 4(1), January 2022. ISSN 2643-1564. doi: 10.1103/physrevresearch.4.013009. URL http://dx.doi.org/10.1103/PhysRevResearch.4.013009.

[6] Tuhin Sahai, Adrian Ziessler, Stefan Klus, and Michael Dellnitz. Continuous relaxations for the traveling salesman problem. *Nonlinear Dynamics*, 97(4):2003–2022, July 2019. ISSN 1573-269X. doi: 10.1007/s11071-019-05092-5. URL http://dx.doi.org/10.1007/s11071-019-05092-5.

[7] Mickey R Wilhelm and Thomas L Ward. Solving quadratic assignment problems by 'simulated annealing'. *IIE transactions*, 19(1):107–119, 1987.