CHN and Swap Heuristic to Solve the Maximum Independent Set Problem

Bouhouch Adil[1], **Loqman Chakir**[2], **and El Qadi Abderrahime**[3]
[1] Team TIM, High School of Technology, CEDoc-SFA faculty of sciences, Moulay Ismail University, Meknes, Morocco
[2]department of informatics, Sciences Faculty, Dhar Mehraz, Sidi Mohammed Ben Abdellah University,Fez, Morocco
[3]LASTIMI, High School of Technology - Mohammed V University of Rabat, Morocco

Article Info

Article history:
Received: Jan 9, 2017
Revised: Jun 14, 2017
Accepted: Jul 3, 2017

Keyword:
Max-Stable problem
Independent set
CHN
Local search
Combinatory problems
Graph

ABSTRACT

We describe a new approach to solve the problem to find the maximum independent set in a given Graph, known also as Max-Stable set problem (MSSP). In this paper, we show how Max-Stable problem can be reformulated into a linear problem under quadratic constraints, and then we resolve the QP result by a hybrid approach based Continuous Hopfeild Neural Network (CHN) and Local Search. In a manner that the solution given by the CHN will be the starting point of the local search. The new approach showed a good performance than the original one which executes a suite of CHN runs, at each execution a new leaner constraint is added into the resolved model. To prove the efficiency of our approach, we present some computational experiments of solving random generated problem and typical MSSP instances of real life problem.

Corresponding Author:
Name Bouhouch Adil
Affiliation Team TIM, High School of Technology Meknes
Address High School of Technology Meknes Moulay Ismail university Meknes Morocco
Phone +212664786422
Email bouhouch.adil@gmail.com

1. INTRODUCTION

The Max-Stable Set Problem (MSSP) is a problem that attempts to find the largest independent set at a given graph. All the nodes included in the independent set must respect the condition that they are not pairwise connected by an arc. Max-table is largely applied in many areas: case-based reasoning [1], computer vision [2], scheduling, ... Max-Stableis a Strong NP-Hard problem while it's hard to be approximate. Therefore, solving MSSP in polynomial time for arbitrary graph case is unlikely. For arbitrary graph there are many exact algorithm which enumerating all cliques and select the one with the maximal cardinality. To our knowledge Harary [3] was the first one who introduced in the literature an exact method. Loukakis [4] generated all maximal independent sets lexicographically introduced by a depth-first enumerative algorithm. Their study includes a comparison against Regneri [5] and Tsukiyama [6] algorithmes. The theoretical superior efficiency of their algorithm is also reinforced by computational results, moreover there method was largely faster than that proposed by Tsukiyama [6] and that introduced by Bron [7]. After two years, Loukakis [8] imported additional change to their previous work [4] and improve it by three speed-up. In the years 1988s, Johnson [9] introduced an exact approach which determines all maximal stable sets in lexicographic order. The approach get each independent set by times complexity of order $O(n^3)$. Chiba [10] proposed an algorithm to the maximal cliques on the order $O(a(G)m\mu)$ of times complexity -where a(G) is the arboricity of graph G- , this is over the time complexity of [6]. Based on the Born [7] work, Tomita [11] proposed an improved variant having complexity equal to $O(3^{n/3})$. Intuitively, it seems that exacts approaches are better to solve Max-stable problem. They consist to enumerate all possible independent set and then select the one which have the maximum of cardinality [12]. However, the analysis of the complexity discouraged this idea for large graph. In this case heuristic methods proved to be an efficient alternative away to solve large problem instances. The second categories contains Meta-heuristic methods which explore the search space, local search methods and hybrid methods. In [13] show how to build a good hybrid strategies. Recently, many approach based neural network was developed to solve combinatorial and hard optimisation problem. The authors of [14]investigate Continuous Hopfield Neural Network (CHN) to solve largess instances of Max-Stable problems. The mean idea is to execute the CHN many times. The role of first run is to find a valid initial solution of MSSP by CHN and a quadratic reformulation. In the second step the cardinality of the solution given by first run is added as linear constraint to the resolution model, then they run CHN again. The second step can be repeated until to have no solution improvement. In this paper we propose new hybrid approach, in order to take advantage of he faster convergence of CHN in the one hand. In the second hand, the Local Search which look through neighborhods to find the best one. To solve MSSP by LS, There are many successful heuristic [15, 16, 17, 18, 19, 20] . The common among them is The start with a random solution and improve it regularly by very simple operations like deletions of nodes which don't meet the adjacent condition , insertions of new nodes or swaps (case when current node succeed by its neighbors).

This paper is structured as follows: In section 2. we present the resolution model bested CHN which is divided into many steps:we introduce the continuous Hopfeild network, next we give the reformulation of maximum stable set problem as a 0-1 quadratic program, then we build an adapted energy function of CHN. Section 3. is devoted to improve the solution by LS. Experimental results are presented in the last section.

2. CONTINUOUS HOPFEILD NEURAL NETWORK TO SOLVE MSSP

In this section we present an overview of [14] approach which implement CHN to solve a quadratic model of MSSP.

2.1. The continuous Hopfield neural network

CHN is a fully connected neural Model with one layer. It was Introduced by Hopfield in the year 1980 to solve combinatorial problems. Further, Hopfield [21] proposed an energy functions to solve many optimization problems as linear programming problems, analog to digital conversion, graph coloring problem,TSP, processing image and . The evolution of CHN dynamic is controlled by The following differential equation:

$$\frac{dy}{dt} = -\frac{x}{\tau} + T\,x + i^b \tag{1}$$

where

x : vector of neurons input

y : vector of output

T : the Matrix of weight between each neurones pairs

Neurons output is governed by function:

$$x_i = g(y_i) = \frac{1}{2}(1 + \tanh(\frac{y_i}{u_0})) \qquad with \qquad u_0 > 0 \quad and \quad i = 1, ..., n$$

The g function is bounded ($g(u) \in [0, 1]$) and u_0 is a parameter to estimate the gain (or slope) of the activation function. The limit point of the CHN u^e by this differential system exists such that $u(t) = u^e$ $\forall t \geq t_e$ (and $t_e \geq 0$), this point is called an equilibrium point The energy of CHN is a Lyapunov function defined as:

$$E(x) = -\frac{1}{2}x^t T x - (i^b)^t x + \sum_{i=1}^{n} \frac{1}{\tau_i} \int_0^{x_i} g^{-1}(v)dv$$

If T is symmetric then the equilibrium points existence is guaranteed. So, any combinatorial problems formulate as the following expression can be solved by The CHN:

$$E(x) = -\frac{1}{2}x^t T x - (i^b)^t x \tag{2}$$

The matrix T is called also matrix of weights connections and just inhibitory connection is allowed on this symmetric matrix. $E(x)$ evaluate at the hypercube $[0,1]^n$ and converge to the corners of this n-dimensional space. To solve a combinatorial problem by CHN we need just to associate the energy function of CHN with the objective function of problem to be minimized, so that the minimum of $E(x)$ coincide with the combinatorial problem solution. Implicitly, the inputs of network outputs represent the problem solution. To show how we can mapping a combinatorial problem to be associated with it neural model we take the assignment problem. It is a easier and direct model to be mapped. So, we consider the following model of assignment problem with n variables m linear constraint:

$$(QP) \qquad \begin{cases} Min & \frac{1}{2}x^tQx + q^tx \\ Subjectto & \\ & Ax = b \\ & x_i \in \{0,1\} \qquad i = 1, ...n \end{cases}$$

Furthermore, we he need to define the following set to solve this QP:

- $H \equiv \{x \in [0,1]^n\}$: the Hamming hypercube
- $H_c \equiv \{x \in \{0,1\}^n\}$: the corners of the Hamming hypercube.
- $H_f \equiv \{x \in H_c : Ax = b\}$: feasible solution set.

To map the QP bellow, we must respect some conditions so that local minimums of the QP to be associated with the CHN limit. The energy can also be defined by two terms:

$$E(x) = E^0(x) + E^R(x) \qquad \forall x \in H$$

Where:

- the term $E^0(x)$ is associated with the problem objective function.
- the quadratic function $E^R(x)$ have two objectives. The first one, is to penalizes the connections which violated at last one constraint. The second one, concerns to guarantees that the CHN converge to a valid solution. To perform a good mapping, this function must be constant in FxH and ding a well choice.

For this problem an adapted generalized function is proposed in [22]:

$$E(x) = \frac{\alpha}{2}x^tQx + \frac{1}{2}(Ax)^t\phi(Ax) + x^t diag(\gamma)(1-x) + \beta^t Ax \qquad \forall x \in H \tag{3}$$

With parameters $\alpha \in \mathbb{R}^+, \gamma \in \mathbb{R}^n, \beta \in \mathbb{R}^N$ and a $m \times m$ matrix ϕ. The goal of this work is to solve the maximum cardinality of the independent set by improving the previous approach based on the CHN proposed by [14]. The main idea of last work is to convert the MSSP as CHN energy function. Before mapping problem used a quadratic 0-1 model to represent the MSSP. In the section 2.2., we describe the reformulation of MSSP, then we present the solving approach in the section 2.3..

2.2. Formulation of the Maximum Stable Set Problem

Let $G = (V, E)$ an undirected graph with V a set of n nodes and E the set of m edges. An independent set of a graph G is a set of nodes S with the property that any node nodes in S is not connected by a direct edges to others nodes of S. The MSSP consist to determine the independent set which have the cardinality maximum $\alpha(G)$.

So, the objective function to maximize is the number of nodes which are pairwise independent, rather we reformulate the connection penalty by a quadratic constraint which represent direct edge connection. To perform the reformulation, we define the binary variables x_i such that:

$$x_i = \begin{cases} 1 & \text{if} \quad v_i \in S \\ 0 & \text{Otherwise} \end{cases}$$

Let $S \subset V$ be a stable set of nodes. For each node v_i of the graph G, we have the following relations:

- To be valid S must don't contains two adjacent node. So, for two node v_i and v_j in S: there corresponding neurones x_i and x_j we have: let p_ij a connection penalty between v_i and v_j.

$$(v_i, v_j) \in S \Longrightarrow x_i = 1 and x_j = 1$$

$$\begin{cases} (v_i, v_j) \in E \Longrightarrow p_{ij}x_ix_j = 0 \Longrightarrow p_{ij} = 0 \\ (v_i, v_j) \notin E \Longrightarrow p_{ij}x_ix_j = 1 \Longrightarrow p_{ij} = 1 \end{cases}$$

Now we can define the quantity:

$$P(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}x_ix_j \tag{4}$$

Referring to all connections constraints imposed in 2.2.. We have $P(h) = 0$ when all nodes in S are pairwise not adjacent. With

$$p_{ij} = \begin{cases} 1 & \text{if} \quad (v_i, v_j) \in E \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

- The objective function to be minimised is :

$$f(x) = -\sum_{i=1}^{n} x_i$$

Finally, the QP of the MSSP problem can be formulated as:

$$(QP) \quad \begin{cases} Min & f(x) = -\sum_{i=1}^{n} x_i \\ Subject\ to & \\ & P(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}x_ix_j = 0 \\ & x \in \{0,1\}^n \end{cases}$$

After the reformulation of MSSP problem as a quadratic 0-1 programming, it can be solved by any adapted approach by minimizing the linear function under quadratic constraints (QP), such as interior point, semidefinite relaxations [23] or lagrangian relaxations[24]. In this paper we are interested in a very different approach based on CHN [14]. In the last approach authors propose to solve the MSSP into two phases. First, solving the QP of MSSP by CHN. We note by $\varGamma$ be the value found by CHN. Second, changing QP model formulation by adding a new constraint to the objective function such as:

$$(NQP) \quad \begin{cases} Min & f(x) = -\sum_{i=1}^{n} x_i \\ Subject\ to & \\ & P(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}x_ix_j = 0 \\ & \sum_{i=1}^{n} x_i \geq \varGamma \\ & x \in \{0,1\}^n \end{cases}$$

Then they solve the new quadratic problem(NQP) by a second CHN associated to this new reformulation (CHN^2) . In this work we propose to replace the second run CHN^2 by the local search heuristic. On other words, the solution given by the first run on the QP will be the starting point of the local search.

2.3. A continuous Hopfield network to solve MSSP

Now The MSSP was reformulate as QP, the next step is to find an adapted energy function , we choose the same one introduced by [14]:

$$E(x) = -\alpha \sum_{i=1}^{n} x_i + \frac{1}{2}\phi \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}x_ix_j + \gamma \sum_{i=1}^{n} x_i(1-x_i) \tag{6}$$

The advantage of this energy expression is more associated with the objective function, it's relaxed by the quadratic constraint.

by corresponding the Equation (2) with Equation (6), we deduce the weights and thresholds:

$$\left\{ \begin{array}{c} T_{i,j} = -\phi p_{ij} + 2\delta_{i,j}\gamma \\ i_i^b = \alpha - \gamma \end{array} \right. \tag{7}$$

With δ_{ij} represent the Kronecker symbol.

$$\delta_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{array} \right.$$

Matrix p is given by Equation (5), the values of parameters ϕ, γ and α are critical to reach the CHN equilibrium points into a valid solution of MSSP.

The study of partial derivatives of the generalized energy function led to control those parameters to guide the convergence to a feasible solution:

$$\frac{\partial E(x)}{\partial x_i} = E_i(x) = -\alpha + \phi \sum_{j=1}^{n} b_{ij}x_j + \gamma(1-2x_i)$$

To determine the parameters-setting, Talavan [25] used the hyperplane method analyse to study $\partial E(x)$.

This procedure consist of partition the Hamming hypercube H by a hyperplane containing all feasible solutions so that two properties must be respected by the CHN evolution: first, adding any solution not belonging to this hyperplane, second, dropping out any infeasible solution belonging to the hyperplane.

Also, some conditions are imposed to determine these parameters-setting easily:

$$\phi > 0, \quad \gamma \geq 0$$

- To minimize the objective function, we fixed the following constraint:

$$\alpha > 0$$

- To escape the stability of the interior points $x \in H - H_C$, the next constraint is necessary:

$$T_{i,i} = 2\gamma \geq 0$$

While MSSP model contains just one constraint then we have:

$$H_C - H_F = \{x \in H_C / h(x) > 0\}$$

Let $x \in H_C - H_F$, so for two adjacents nodes x_i and x_j are in the stable set S, then $x_i = x_j = 1$ and therefore the activation of x_i will be discouraged if $E_i^0(x) \geq \epsilon$ where $\epsilon > 0$.

Based to this study the parameter settings are restricted by the following condition:

$$-\alpha + \phi - \gamma \geq \epsilon$$

A feasible solution can be reached if the following conditions are respected:

$$\left\{ \begin{array}{l} \alpha > 0, \quad \phi > 0, \quad \gamma \geq 0 \\ -\alpha + \phi - \gamma = \epsilon \end{array} \right. \tag{8}$$

The question then is about making good choice of these parameter under the condition bellow to calculate the weights and thresholds of the CHN.,The advantage of resolving MSSP by Continuous dynamic of Hopfield neural network is that CHN are very fast but always converge to a local minima, in [14] authors insert some linear constraint to perturb the network weights and consequently escaping from all local minima less than threshold Γ. This method appear efficient, but the control of CHN convergence become difficult. .

3. PROPOSED APPROACH

Hopfeild neural network converge faster, but it converges closely to the nearest local minima of the starting point. So many time the network gives a no good solution. To overcome this weakness many solution was proposed to escape from local minima. In [14], authors propose to add a linear constraint which restrict a lower bound of the minimum cardinality of the solution, as shown in [14], the insertion of new constraint to the QP leads to improve the solution, but in contrast of obtained numerical results, we remark that network failed to give a valid solution many time. To improve this meta-heuristic approach, we propose to use a hybridization with other heuristic. It seems beneficial to combine CHN with an adapted local search. In [16] authors prove that swap(1,2) give a good performance. Algorithm (1) show the proposed local search based Swap heuristic. As described, this process by replacing each node n in the the given initial solution by two nodes u and v. The chosen u and v must be neighbors of n. Therefore, the solution is improved at each time by adding one node. This last search is down in linear times. This is possible by adding a collection of neighborhods set $V(n)$ which store for each graph node the set of it adjacent nodes. The Local search work directly on the solution given by the CHN. We note that to check if a node n of the graph is in the solution, we verify if its associated CHN node is active. Also, to reduce the time to find the neighbors of the candidate node which can be replaced, we maintaining a stack To_visit of valid candidate to visit. After examine each node, we removed it from the candidate list, and go to the next until the list To_visit become empty. Although, we added to the candidates list To_visit any new insert node in the solution S.

In this work, we implement the direct and simple local search based on swap heuristic. For this, to replace the current candidate node n, we consider just the first pair of node u, v founded, which meet conditions. Nevertheless, we can add more improvement to this LS, like sorting V(n) by number adjacent neighbor in the solution [26] or plateau search [16].

```
Function $Swap\_1\_2$( S,T : ) : Solution
    In :

        • $To\_visit$ : Stack of candidate nodes

        • $V(i)$     : Set of neighborhood of the $node_i$

    Out : Improved Solution
    $To\_visit = S$
    While $(To\_visit\ is\ not\ empty)$ do
        $n = pop(To\_visit)$
        For $(u, v) \in V(n) \times V(n)$ do
            $Have\_Adjacent\_in\_S = false$
            If $(u \notin S\ and\ u \notin S\ and\ T(u, v) = -\phi\ )$ then
                If $(is\_marked(u) = false$ and $is\_marked(v) = false)$ then
                    For $(a \neq n) \in S$ do
                        If $(T(u, a) = -\phi\ or\ T(v, a) = -\phi\ )$ then
                            $Have\_Adjacent\_in\_S = true$
                            Break
                        end If
                    end For
                    If $(Have\_Adjacent\_in\_S = false\ )$ then
                        replace n by u and v
                        add u and v $To\_visit$
                        Break
                    end If
                end If
            end If

        end For
        mark(n)
    done

End
```

Algorithm 1. Local search algorithm

Naturally, CHN attempted to minimize the number conflicted constraints, therefore, some times network can stabilise with not null Energy $E \neq 0$, and gives an invalid solution. To overcome this problem, we add a local heuristic called Remove Process. It delete nodes from the solution until to being valid.

```
Sort S by number of Adjacent
For each $v \in S$ do
    if($\exists u \in S$ / $(u, v) \in E$) delate v
end For
```

Algorithm 2. Delate Process

4. NUMERICAL RESULT

To show the efficiency of our approach, we present a computational results. First, we run the solver on the random indirected generated graphs. The used generator return a $G_{n,p}$ random graph, also known as an Erdõs-Rényi graph or a binomial graph [27], where n the number of nodes and the graph density which can be determinate by the probability for edge creation p. So we generated five classes classified by the number of graph nodes from 100 to 500, and for each class, the density p varied from 10% to 100%. For a given p we generate random 100 instances. Figure 1 plot the CPU time tacked by CHN, CHN^2 -the improved variant [14]- and CHNSwap. The comparison is down under fixing the density $D = 50\%$ and varying the numbers of nodes. Practically, the curves of CHN and CHNSwap are superposed.

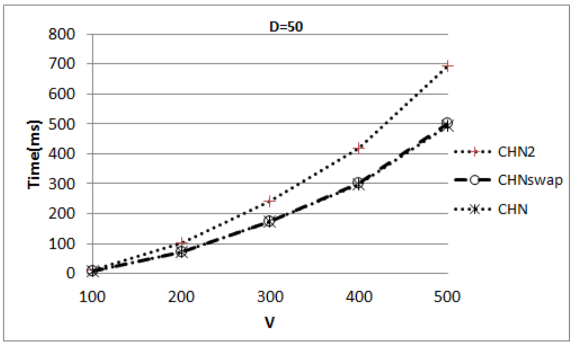

Figure 1. the CPU time as a function of number of nodes.

Figure 2 give the evolution of times of CHNSwap approach as function of the graph density (D), respectively, for classes V=100, V=300 and V=500. We note that left vertical axis is the times performed by CHN and the right for LS. It is clear that the problem difficulty increases as the number of nodes increases. The same remark can be seen at all tested graph classes.
To study the solution quality we run our approach on selected instance from the benchmark DIMACS[28].

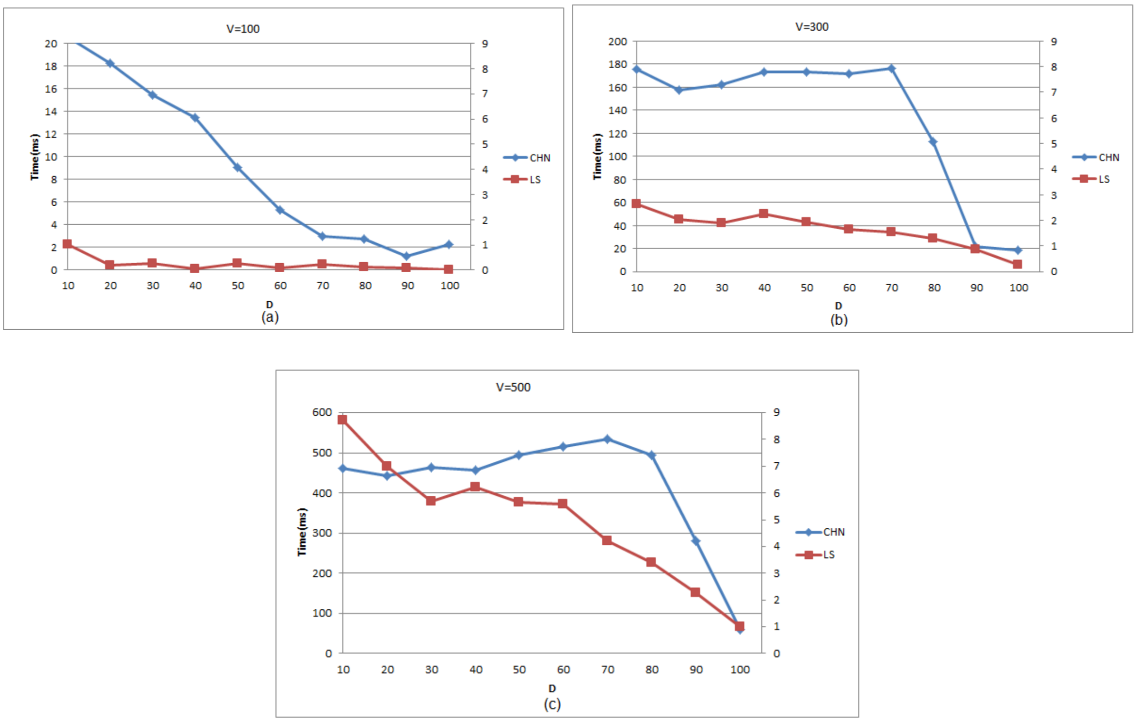

Figure 2. the CPU times tacked by CHN and LS in CHNSwap (a)Class of graph containing 100 nodes (b)Class of graph containing 300 nodes (c)Class of graph containing 500 nodes

				CHN			CHNSwap		
Graphs	V	E	$\alpha(G)$	Min	Mean	Best	Min	Mean	Best
brock200_2.clq	200	9876	12	8	8	8	9	9	9
brock200_4.clq	200	13089	17	11	11	11	13	13,99	14
C250.9.clq	250	27984	24	34	34	34	37	37,69	38
c-fat200-1.clq	200	1534	58	12	12	12	12	12	12
c-fat200-2.clq	200	3235	44	23	24	24	24	24	24
c-fat200-5.clq	200	8473	30	58	58	58	58	58	58
DSJC125.9.col	125	6961	44	30	30,08	31	32	32	32
frb30-15-1.clq	450	83198	55	20	20	20	24	24	24
gen200_p0.9_44.clq	200	17910	44	31	31	31	35	35,77	36
hamming6-4.clq	64	704	4	2	2,42	3	4	4	4
johnson16-2-4.clq	120	5460	8	6	6,35	7	8	8	8
johnson8-2-4.clq	28	210	14	2	4	6	4	4	4
johnson8-4-4.clq	70	1855	11	8	8,91	9	10	13,27	14
keller4.clq	171	9435	126	11	6	5	7	7	7
MANN_a27.clq	378	70551	126	117	117	117	117	117,98	121
MANN_a9.clq	45	918	16	12	12	12	15	15,76	16
p_hat300-1.clq	300	10933	8	6	6	6	6	6,51	8
p_hat300-2.clq	300	21928	25	22	22	22	24	24,99	25
p_hat300-3.clq	300	33390	36	29	30,89	31	33	33,97	34
san200_0.7_1.clq	200	13930	18	15	15	15	15	15	16
san200_0.9_3.clq	200	17910	44	31	31	31	33	33	33
san400_0.9_1.clq	400	71820	100	39	44,15	46	52	52,98	53

Table 1. Comparison of solution quality between CHN and CHNSwap over DIMACS

Graphs	E	V	CHN(ms)	Swap(ms)
johnson8-2-4.clq	210	28	2,12	0,01
MANN_a9.clq	918	45	3,18	0,06
hamming6-4.clq	704	64	1,62	0,18
johnson8-4-4.clq	1855	70	9,53	0,12
johnson16-2-4.clq	5460	120	25,65	0,32
DSJC125.9.col	6961	125	7,14	0,32
C125.9.clq	6963	125	9,88	0,08
C125.9.clq	6963	125	9,44	0,06
keller4.clq	9435	171	41,06	0,31
c-fat200-1.clq	1534	200	77,86	1,17
c-fat200-2.clq	3235	200	61,01	0,95
c-fat200-5.clq	8473	200	34,64	0,61
brock200_2.clq	9876	200	76,04	0,72
brock200_4.clq	13089	200	65,54	0,4
san200_0.7_1.clq	13930	200	16,09	0,5
gen200_p0.9_44.clq	17910	200	4,58	0,44
san200_0.9_3.clq	17910	200	2,5	0,68
C250.9.clq	27984	250	8,92	0,71
p_hat300-1.clq	10933	300	163,3	2,38
p_hat300-2.clq	21928	300	159,36	2,43
p_hat300-3.clq	33390	300	139,28	1,07
MANN_a27.clq	70551	378	160,81	0,82
san400_0.9_1.clq	71820	400	115,3	1,82
frb30-15-1.clq	83198	450	362,35	2,31

Table 2. Times performance

The efficiency of the proposed approach is reinforced by the experimental results. Table 1 shows that our method gives good result then used CHN alone. Table 2 show the time added by the improvement process with local heuristics against the time needed by CHN. It's clear that improvement is down with adding non-significant time.

Our approach is very powerful from a theoretical point of view. This approach gives a good performance compared with the improved approach CHN^2 proposed by [14] which crush often with no valid solution.

5. CONCLUSION

In the last decade several areas have applied MSSP problem and Max-Clique problem. So, many heuristic were developed. In this sense we directed our contribution. In this work, a new approach has been proposed based CHN and swap local heuristic to solve the maximal cardinality of independent sets. The MSSP is solved into three step: First, it is reformulate as quadratic problem. Second, the QP is associated with CHN energy function, then we stabilise CHN to converge at the solution. Third, the local search process performed by starting from the solution given in the second step. It same difficult to integrating the local heuristic in the main loop of CHN stabilisation for this we have chosen the collaborative hybridisation between CHN and LS. We executed a series of experiments in order to prove the efficiency regarding the time complexity and the solution of quality. The time allowed to LS is very small so that we can assert that our approach is able to find the maximal independent set for very large graph. Furthermore, the rate of success to give a valid solution is up to 100% in our approach. We note that this approach can be used to solve the Max-Clique problem special for perfect graph by solving the graph dual.

REFERENCES

[1] I. Iglezakis, "The conflict graph for maintaining casebased reasoning systems," in *International Conference on Case-Based Reasoning*. Springer, 2001, pp. 263–275.

[2] D. H. Ballard and C. M. Brown, "Computer vision, article, 4 pages prentice-hall," *Englewood Cliffs, New Jersey, believed to be published more than one year prior to the filing date of the present application*, 1982.

[3] F. Harary and I. C. Ross, "A procedure for clique detection using the group matrix," *Sociometry*, vol. 20, no. 3, pp. 205–215, 1957.

[4] E. Loukakis and C. Tsouros, "A depth first search algorithm to generate the family of maximal independent sets of a graph lexicographically," *Computing*, vol. 27, no. 4, pp. 349–366, 1981.

[5] M. Regneri, "Finding all cliques of an undirected graph," in *SeminarCurrent Trends in IE WS Jun*, 2007.

[6] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, "A new algorithm for generating all the maximal independent sets," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 505–517, 1977.

[7] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[8] E. Loukakis, "A new backtracking algorithm for generating the family of maximal independent sets of a graph," *Computers & Mathematics with Applications*, vol. 9, no. 4, pp. 583–589, 1983.

[9] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, "On generating all maximal independent sets," *Information Processing Letters*, vol. 27, no. 3, pp. 119–123, 1988.

[10] N. Chiba and T. Nishizeki, "Arboricity and subgraph listing algorithms," *SIAM Journal on Computing*, vol. 14, no. 1, pp. 210–223, 1985.

[11] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for finding all the cliques," Technical Report TR-C5, UEC, Tech. Rep., 1988.

[12] C. Mannino and A. Sassano, "An exact algorithm for the maximum stable set problem," *Computational Optimization and Applications*, vol. 3, no. 3, pp. 243–258, 1994.

[13] Q. Duan, T. W. Liao, and H. Yi, "A comparative study of different local search application strategies in hybrid metaheuristics," *Applied Soft Computing*, vol. 13, no. 3, pp. 1464–1477, 2013.

[14] M. Ettaouil, C. Loqman, and K. Elmoutaouakil, "Improved optimal competitive hopfield network for the maximum stable set problem," *International Journal on Computer Science & Engineering*, vol. 2, no. 6, pp. 2071–2077, 2010.

[15] R. Battiti and M. Protasi, "Reactive local search for the maximum clique problem 1," *Algorithmica*, vol. 29, no. 4, pp. 610–637, 2001.

[16] A. Grosso, M. Locatelli, and F. Della Croce, "Combining swaps and node weights in an adaptive greedy approach for the maximum clique problem," *Journal of Heuristics*, vol. 10, no. 2, pp. 135–152, 2004.

[17] P. Hansen, N. Mladenović, and D. Urošević, "Variable neighborhood search for the maximum clique," *Discrete Applied Mathematics*, vol. 145, no. 1, pp. 117–125, 2004.

[18] K. Katayama, A. Hamamoto, and H. Narihisa, "An effective local search for the maximum clique problem," *Information Processing Letters*, vol. 95, no. 5, pp. 503–511, 2005.

[19] W. Pullan and H. H. Hoos, "Dynamic local search for the maximum clique problem," *Journal of Artificial Intelligence Research*, vol. 25, pp. 159–185, 2006.

[20] S. Richter, M. Helmert, and C. Gretton, "A stochastic local search approach to vertex cover," in *Annual Conference on Artificial Intelligence*. Springer, 2007, pp. 412–426.

[21] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[22] M. Ettaouil, C. Loqman, K. Haddouch, and Y. Hami, "Maximal constraint satisfaction problems solved by continuous hopfield networks." *WSEAS Transactions on Computers*, vol. 12, no. 2, pp. 29–40, 2013.

[23] M. Ettaouil and C. Loqman, "Constraint satisfaction problems solved by semidefinite relaxations," *WSEAS Transactions on Computers*, vol. 7, no. 7, pp. 951–961, 2008.

[24] B. Thiongane, A. Nagih, and G. Plateau, "An adapted step size algorithm for a 0-1 biknapsack lagrangean dual," *Annals of Operations Research*, vol. 139, no. 1, pp. 353–373, 2005.

[25] P. M. Talaván and J. Yáñez, "The generalized quadratic knapsack problem. a neuronal network approach," *Neural networks*, vol. 19, no. 4, pp. 416–428, 2006.

[26] D. V. Andrade, M. G. Resende, and R. F. Werneck, "Fast local search for the maximum independent set problem," *Journal of Heuristics*, vol. 18, no. 4, pp. 525–547, 2012.

[27] S. Choudum, "A simple proof of the erdos-gallai theorem on graph sequences," *Bulletin of the Australian Mathematical Society*, vol. 33, no. 01, pp. 67–70, 1986.

[28] "The second dimacs implementation challenge," ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/, 1992.