Finally, the solution is given by our solver in the 0.001 second (time of resolution). The value for each variable is: $y_1 = v_1$, $y_2 = v_3$ and $y_3 = v_7$ such that this solution is coded in the following way: $SHOES = Cordovans$, $SHIRT = green$ and $SLACKS = gray$. We show that, the number of violated constraints is 1 because the tuple $(Cordovans, green)$ violate the constraint $C(SHOES, SHIRT)$ (remark 3).

### 4.2 Numerical results

In order to show the practical interest of the proposed approach, the experiments are achieved to solve some typical problems of different natures. These series represent a large spectrum of instances [22]. These experiments are effectuated in personal computer with a 2.79 GHz processor and 512 MB RAM. The performance has been measured in terms of the CPU time per second.

This solver is modeled by the Unified Modeling Language method and implemented by java language. The starting points are chosen in such a way that the points corresponding to the most constrained variables are the most favored in the processing. This choice allows the continuous Hopfield network to handle in a first order the variables most constrained. This process lead to an assignment of values to variables such that the number of satisfied constraints is maximized. This is guaranteed by a good affectation to the variables most constrained. Therefore, the maximum of constraints in the problem are satisfied. Finally, the starting points are randomly generated by the following expression:

$$x_{ir} = 0.8 + 0.19\frac{NVC_i}{NVMC} + 10^{-2}U$$

Where $NVC_i$ is the number of participation for each variable $y_i$ in the set of constraints *C*, $NVMC = \max\left\{NVC_i / i \in \{1,...,n\}\right\}$ representing the more present variable in the constraints of the Max-CSP problem, and *U* is a random variable in the interval [-0.5, 0.5]. Recall that n is the number of variables. Based on a series of experiments, $\alpha$ and $\varepsilon$ are determined by the following values:

$$\alpha = \frac{1}{n}, \ \varepsilon = 10^{-4}$$

In this experimentation, some instances as benchmarks for the first round of the 2008 Max-CSPs [22] solver competition are used to test this algorithm (See Table 1).

A statistical study was represented in order to examine the quality of our approach. This study is based on the computation of performance operators. In fact, the quality of solutions obtained by our approach was evaluated in terms of performance report:

$$\rho = \frac{\text{number of violated constraints obtained by this approach}}{\text{best number of violated constraints obtained by the best solver}}$$

Among these operators performance, we have (Table 1):

- Ratio mode: the ratio between the most repetitive (mode) number of violated constraints (the optimal value obtained by CHN) in the number of run and the least number of violated constraints (best results existing in the literature) obtained by the best solver, Ratio mean: the ratio between the average number of violated constraints in a number of run and the best results existing in the benchmarks obtained by other solver,
- Ratio minimum : the ratio of the smaller number of violated constraints (better results obtained by our solver) and the best result existing in the literature obtained by other solver,
- Mean of CPU time: the average time consumed to obtain the solution in a number of run.

We show that, for each instance, if Ratio minimum is less than 1 then the proposed approach gives the best results than the others, i.e., gives a solution that violates fewer constraints compared to the best existing solver. Also Ratio minimum is superior to 1 then the proposed approach cannot give the results better than the others, i.e., gives a solution that violates more constraints compared to the best existing solver. Moreover, if Ratio minimum is equals to 1, in this case the given results is similar to the best solver existing in benchmarks (Table 1).

In comparison with other Max-CSPs solvers, the time of resolution obtained by our software is better than others; it does not exceed 1 second for the most of instances. Generally, our solver is very successful, it happens to get the solution to the maximal constraint satisfaction problems in a minimum time than the author solvers of Max-CSP [22] (See Table 1). For example, instances "kbtree-9-7-3-5-90-08", "cnf-2-40-1800-067529", "maxcut-30-400-5", "maxcut-30-340-5", etc, the proposed solver gives a solution that violates fewer constraints compared to the best existing solver.

| Name of instances | number of constraint | Best result obtained on benchmark | Best CPU time obtained on benchmark | Number of run | Ratio Minimum | Ratio mean | Ratio mode | Mean of iterations | mean of CPU Time |
|---|---|---|---|---|---|---|---|---|---|
| geom-40-2 | 78 | 22 | 0.03s | 200 | 1.05 | 1.36 | 1.27 | 58.76 | 0.006s |
| geom-30a-4 | 81 | 4 | 0.323 s | 200 | 1.25 | 3.26 | 2.75 | 192.29 | 0.046s |
| vcsp-25-10-25-87-44 | 75 | 32 | 5.198s | 200 | 1.38 | 1.99 | 1.78 | 254.42 | 0.212s |
| vcsp-25-10-21-85-33 | 63 | 19 | 0.219 s | 200 | 1.58 | 2.81 | 2.42 | 261.8 | 0.255s |
| vcsp-25-10-25-87-36 | 75 | 29 | 1.587 s | 200 | 1.21 | 2.71 | 1.86 | 244.96 | 0.209s |
| scenw-6-sub4-20 | 477 | 23 | 0.132 s | 100 | 1.83 | 4.91 | 2.39 | 326.1 | 1.023s |
| scenw-6-sub2 | 353 | 22 | 0.374s | 100 | 1.32 | 3.96 | 1.86 | 326.31 | 2.466s |
| scenw-6-sub2 | 353 | 22 | 0.374s | 100 | 1.32 | 3.96 | 1.86 | 326.31 | 2.466s |
| kbtree-9-7-3-5-90-08 | 189 | 123 | 0.311 s | 100 | 0.93 | 1.44 | 1.24 | 235.43 | 0.076s |
| kbtree-9-7-3-5-60-01 | 189 | 54 | 0.335s | 100 | 1.17 | 2.1 | 1.48 | 214.25 | 0.071s |
| kbtree-9-7-3-5-90-11 | 189 | 123 | 0.338s | 100 | 0.98 | 1.52 | 1.24 | 213.16 | 0.068s |
| kbtree-9-7-3-5-80-50 | 189 | 97 | 0.846 s | 100 | 1.03 | 1.73 | 1.26 | 189.65 | 0.062s |
| kbtree-9-7-3-5-70-50 | 189 | 72 | 1.522s | 100 | 1.07 | 1.84 | 1.46 | 217.16 | 0.069s |
| kbtree-9-5-3-5-90-45 | 255 | 163 | 2.297s | 100 | 0.95 | 1.6 | 1.21 | 232.88 | 0.133s |
| kbtree-9-5-3-5-90-10 | 255 | 167 | 3.17s | 100 | 1 | 1.66 | 1.26 | 183.89 | 0.097s |
| kbtree-9-5-3-5-80-47 | 255 | 129 | 4.969s | 100 | 1.09 | 1.74 | 1.34 | 226.37 | 0.121s |
| kbtree-9-5-3-5-80-08 | 255 | 134 | 5.931s | 100 | 1.13 | 1.51 | 1.31 | 266.14 | 0.141s |
| kbtree-9-2-3-5-90-29 | 309 | 196 | 461.111s | 100 | 1.06 | 1.45 | 1.2 | 271.75 | 0.302s |
| kbtree-9-2-3-5-80-50 | 309 | 153 | 1143.39s | 100 | 1.11 | 1.86 | 1.28 | 236.32 | 0.263s |
| cnf-2-80-300-186945 | 286 | 25 | 1.032s | 100 | 1.04 | 1.3 | 1.28 | 35.83 | 0.015s |
| cnf-2-40-1800-067529 | 667 | 246 | 1.367s | 50 | 0.97 | 1.07 | 1.06 | 30.68 | 0.003s |
| cnf-2-40-1500-890427 | 640 | 210 | 1.417s | 50 | 1.01 | 1.1 | 1.06 | 28.7 | 0.002s |
| cnf-2-40-1600-616123 | 663 | 227 | 1.856s | 50 | 0.98 | 1.06 | 1.04 | 27.5 | 0.002s |
| cnf-2-40-2500-147426 | 685 | 307 | 2.139s | 50 | 0.99 | 1.06 | 1.05 | 35.94 | 0.003s |
| cnf-2-40-2600-873121 | 670 | 297 | 3.418s | 50 | 0.97 | 1.09 | 1.05 | 42.22 | 0.004s |
| cnf-2-40-2400-421728 | 681 | 291 | 3.742s | 50 | 1 | 1.1 | 1.06 | 36.94 | 0.003s |
| cnf-2-40-2400-421727 | 683 | 284 | 5.117s | 50 | 1.02 | 1.12 | 1.07 | 41 | 0.003s |
| cnf-2-80-800-815450 | 710 | 102 | 163.366s | 50 | 1.23 | 1.3 | 1.28 | 19.72 | 0.010s |
| cnf-2-80-800-815444 | 717 | 107 | 391.728s | 50 | 1.13 | 1.25 | 1.22 | 29.26 | 0.013s |
| cnf-2-80-1100-992546 | 946 | 160 | 979.48s | 50 | 1.08 | 1.11 | 1.11 | 22.08 | 0.007s |
| cnf-2-80-1200-718241 | 989 | 176 | 1261.0s | 50 | 1.05 | 1.13 | 1.14 | 30.28 | 0.010s |
| maxcut-30-400-5 | 400 | 179 | 6.597s | 100 | 0.97 | 1.12 | 1.06 | 72.73 | 0.003s |
| maxcut-40-480-6 | 480 | 193 | 231.657s | 100 | 1.07 | 1.2 | 1.17 | 37.73 | 0.005s |
| maxcut-30-340-5 | 340 | 142 | 3228.99s | 100 | 0.99 | 1.19 | 1.12 | 40.86 | 0.004s |
| maxcut-30-370-5 | 370 | 160 | 17.851s | 100 | 1.01 | 1.22 | 1.09 | 48.85 | 0.002s |
| maxcut-40-420-5 | 420 | 161 | 20.163s | 100 | 1.06 | 1.23 | 1.2 | 40.17 | 0.004s |
| maxcut-40-440-6 | 440 | 173 | 29.374 s | 100 | 1.07 | 1.24 | 1.14 | 42.58 | 0.004s |
| maxcut-40-480-8 | 480 | 192 | 136.187 s | 100 | 1.03 | 1.22 | 1.17 | 43.99 | 0.004s |
| maxcut-40-540-3 | 540 | 225 | 717.983s | 50 | 1.07 | 1.17 | 1.12 | 44.1 | 0.004s |
| maxcut-40-520-1 | 520 | 210 | 272.491s | 50 | 1.07 | 1.27 | 1.17 | 33.84 | 0.003s |
| maxcut-40-520-10 | 520 | 213 | 340.681s | 50 | 1.08 | 1.16 | 1.14 | 60.08 | 0.005s |
| maxcut-50-580-6 | 580 | 219 | 940.592s | 50 | 1.11 | 1.29 | 1.19 | 34.54 | 0.004s |
| maxcut-40-580-1 | 580 | 241 | 161.684s | 50 | 1.05 | 1.22 | 1.14 | 52.26 | 0.005s |
| maxcut-50-560-10 | 560 | 212 | 176.564s | 50 | 1.07 | 1.24 | 1.21 | 48.58 | 0.011s |
| maxcut-60-580-2 | 580 | 207 | 2121.51s | 50 | 1.15 | 1.26 | 1.27 | 31.54 | 0.007s |
| c-fat200-2 | 16865 | 176 | 1.217s | 25 | 1.22 | 1.22 | 1.22 | 366.36 | 0.963s |
| c-fat500-2 | 116111 | 474 | 283.354s | 25 | 1.24 | 1.24 | 1.25 | 374.12 | 4.906s |
| c-fat500-10 | 78623 | 374 | 1101.83s | 25 | 1.32 | 1.33 | 1.33 | 410.08 | 5.386s |
| c-fat500-5 | 102059 | 436 | 1151.8s | 25 | 1.38 | 1.38 | 1.38 | 369.2 | 4.899s |

Table 1: Computational results of the typical Max-CSP instances

However, in some cases as "vcsp-25-10-21-85-33", "kbtree-9-7-3-5-60-01", "maxcut-50-580-6", "c-fat500-10", etc, our approach violate plus the constraints more than author solvers. Finally, we can conclude that the best results are obtained by this approach.

## 5. Conclusion

In this paper, we have proposed a new approach for solving binary maximal constraint satisfaction problems. The interesting steps of this approach are: proposing the new model of maximal constraint satisfaction problem as a 0-1 quadratic program subject to linear constraints and using the continuous Hopfield network to solve this problem. The most interesting propriety of this approach is used to give the solution of the binary Max-CSP. It is also interesting to note that this method can be used with a non-binary Max-CSP after converting the latter into a binary Max-CSP [14]. The experimental results show that our method can find a good optimal solution in short time compared to other solvers. Future directions of this research are reducing the architecture of Hopfield neural network and applying this approach to get a good solution of real world problems such as warehouse location problem, max-clique and max-cut.

*References:*


[1] M. Charikar, and al., Near-optimal algorithms for maximum constraint satisfaction problems, *Journal ACM Transactions on Algorithms (TALG) TALG Homepage archive*, Vol. 5, No. 3, 2009, pp. 62 – 68.

[2] H. M. El-bakry and N. Mastorakis, A Modified Hopfield Neural Network for Perfect Calculation of Magnetic Resonance Spectroscopy, *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS*, Vol. 5, No. 12, 2008, pp. 1654-1666.

[3] M. Ettaouill, and C. Loqman, A New Optimization Model for Solving the Constraint Satisfaction Problem, *Journal of Advanced Research in Computer Science*, Vol. 1, No. 1, 2009, pp. 13-31.

[4] M. Ettaouil, C. Loqman, K. Haddouch. Job Shop Scheduling Problem solved by the continuous Hopfield networks, *Journal of Advanced Research in Computer Science (JARCS)*, Vol. 2, No. 1, 2010, pp. 31 - 47.

[5] M. Ettaouill, and C. Loqman, Constraint Satisfaction Problems Solved by Semidefinite Relaxations, WSEAS TRANSACTIONS on COMPUTERS, Vol. 7, No. 7, 2008, pp. 951-961.

[6] M. Ettaouil, K. Elmoutaouakil, Y.Ghanou. The continuous Hopfield networks (CHN) for the placement of the electronic circuits problem, WSEAS Transactions on Computer, Vol. 8, No. 12, 2009, pp. 1865-1874.

[7] D. J. Evansi, and M. N. Sulaiman, Solving optimisation problems using neucomp-a neural network compiler, *International Journal of Computer Mathematics*, Vol. 62, No. 1, 1996, pp. 1-21.

[8] E. Freuder, and R Wallace, Partial Constraint Satisfaction, *Artificial Intelligence*, Vol. 58, 1992, pp. 21-70.

[9] M.R Garey, and D.S Johnson, A Guide to the Theory of NP-Completeness, *Computers and Intractability, New York: W.H. Freeman and Company*, 1979.

[10] R. M. Haralick et G. L. Elliott, Increasing tree search efficiency for constraint satisfaction problems, *Artificial Intelligence*, Vol. 14, 1980, pp. 263-313.

[11] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 79, 1982, pp. 2554-2558.

[12] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-states neurons. *proceedings of the National academy of sciences of the USA 81*, 1984, pp. 3088-3092.

[13] J.J. Hopfield, D.W. Tank, Neural computation of decisions in optimization problems, *Biological Cybernetics*, Vol. 52, 1985, pp.1-25.

[14] N. Mamoulis, and K. Stergiou, Solving non-binary CSPs using the Hidden Variables Encoding, *In Proceedings of CP'*, 2001.

[15] V. Kumar, Algorithms for Constraint Satisfaction Problems, T A Survey, *AI Magazine*, Vol. 13, No. 1, 1992, pp. 32-44.

[16] M. Azlinah, Y. Marina, A. M. Itaza, M. Sofianita and A. R. Shuzlina, Constraint Satisfaction Problem Using Modified Branch and Bound Algorithm, *WSEAS*

*TRANSACTIONS on COMPUTERS*, Vol. 7, No. 1, 2008, pp. 1-7.

[17] Richard J. Wallace, Enhancements of Branch and Bound Methods for the Maximal Constraint Satisfaction Problem, *Proc. of AAAI*, 1996, pp.188-196.

[18] P.M. Talavàn and J. Yànez, A continuous Hpfield network equilibrium points algorithm. *computers and operations research*, Vol. 32, 2005, pp. 2179-2196.

[19] Talavàn and J. Yànez, The generalized quadratic knapsack problem. A neuronal network approach, *Neural Networks*, Vol. 19, 2006, pp. 416-428.

[20] Thiongan, and al., An Adapted Step Size Algorithm for a 0-1 Biknapsack Lagrangean Dual, *Annals of Operations Research*, Vol. 139, No. 1, 2005, pp. 353-373.

[21] E. Tsang, *A Foundations of Constraint Satisfaction*, Academic Press, 1993.

[22] MAX-CSP Competition: available results, Available at *http://www.cril.univ-artois.fr/CPAI08/results/results.php?idev=16, 20.*