

# Solving Large MAX CUT Problems Using Cross Entropy & Parametric Minimum Cross Entropy

Boaz Kaminer, Technion

## Abstract:

Two stochastic optimization methods: Cross Entropy (CE) and Parametric Minimum Cross Entropy (PME) were tested against Large Max-Cut problems. The problems were taken from the "The DIMACS Library of Mixed Semidefinite-Quadratic-Linear Programs" challenge web site. The two methods achieved close and even better results than the Best Known Solution. In addition to results presentation, several implementation issues and convergence properties of the Cross Entropy will be discussed.

## 1. Introduction:

The Max-Cut problem, a well known NP-hard problem is considered a tutorial problem for the Cross-Entropy (CE) method [3]. Surprisingly the three dimensional Ising model Max-Cut problem presented in the DIMACS site was declared as a challenge for the CE method. This was the main motivation for the work presented in this paper. Two more motivations were: 1) Empirical evaluation of Costa et al. [4], Cross-Entropy convergence results. 2) Comparison between the new Parametric Minimum Cross Entropy method (PME) and the original more experienced Cross-Entropy Method.

As we shall see below, on one of the problems: "Torus Set g\_3\_8" (TSg38) both CE and PME methods achieved better results than Best Known Solution (BKS) [2]. On the second problem: "Torus Set pm\_3\_8\_50" (TSpm3850) both methods reached results up to 94.76% of the BKS.

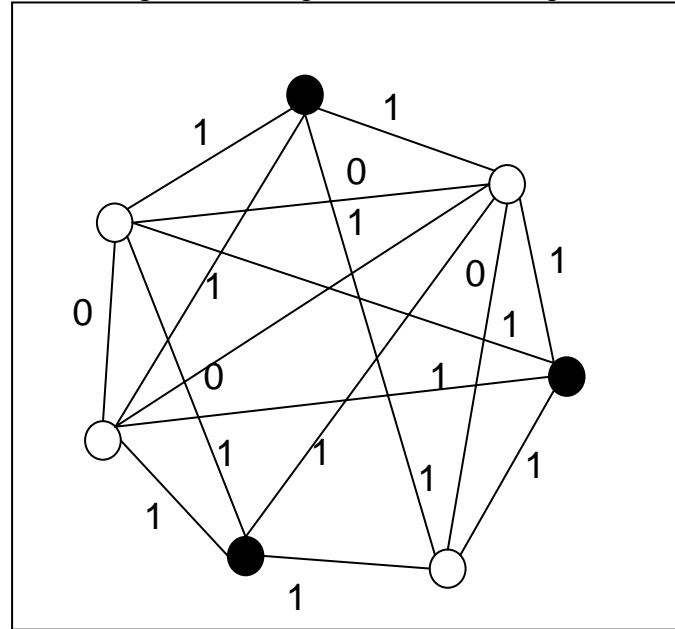
The rest of the paper is organized as follows: The second section describes the problems at hand: the Max-Cut Problem, in particular in the context of the Ising Model. The third part describes the methods used to solve the problems. Section four details results, the fifth part discusses convergence properties, and the six section summaries and concludes this paper.

### 2.1. The Max-Cut problem

Let  $G=(V,E)$  be an undirected edge-weighted graph. A cut  $C$  of  $G$  is any nontrivial subset of  $V$ . The weight of the cut is the sum of weights of edges crossing the cut. A Max-Cut is defined as a cut of  $G$  of maximum weight. Figure #1 gives a toy example of a graph which vertices are

distributed into two groups (black and white). The weights of the edges between the black and white edges have positive weights, thus this cut is considered to be maximal. Determining the MAXCUT of a graph is a known NP-hard problem.

Figure #1: Simple Max-Cut Example



## 2.2. The Three Dimensional Ising Model

Ising (1900-1998), a German physicist developed in 1925 in his PhD thesis a mathematical model in statistical mechanics which represent particles spins in a ferromagnetic material. In his model, each particle spin can be assigned two possible directions up (+1) or down (-1). The energy of the spins directions is computed through the following Hamiltonian:  $-\sum_{i \neq j} w_{ij} s_i s_j$ . When two adjacent particles have opposite spins they add to the total energy of the system and when their spins are similar they retract from the total energy of the system, according to the weight  $w_{ij}$  between the two particles.

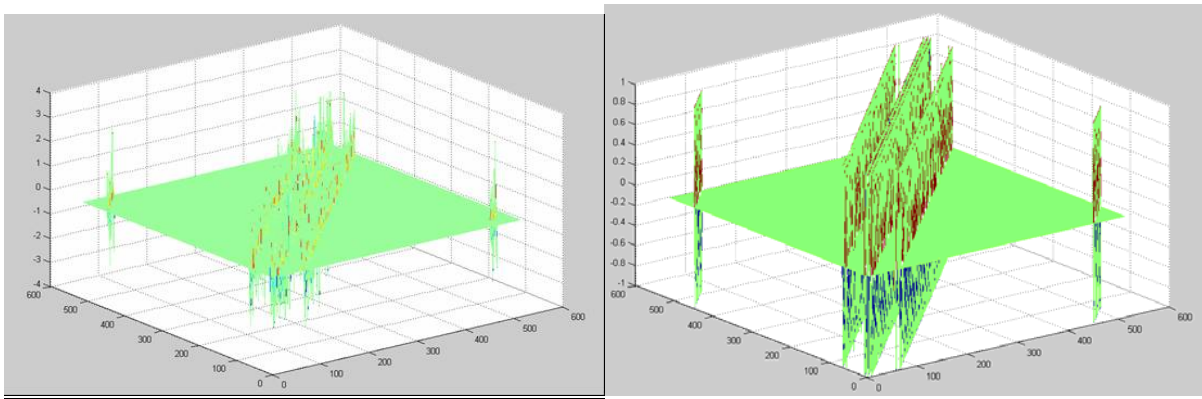
The Ising model is a rather simplified model of ferromagnetism, unlike for example the Potts model, or the Heisenberg model, which didn't assume any spin direction. However, the Ising model advantage is that it is solvable by computational methods. The Ising model in two dimensions, and in the absence of an external magnetic field, was analytically solved at the critical point in 1944 by Lars Onsager but the 3D Ising model resisted solution for decades and was finally proved to be an NP-Complete problem by Sorin Istrail in 2000 [1].

The three dimensional Ising model problem could be easily translated into a Max-Cut problem: The particles placed on a three dimensional grid, are presented by a graph in which the edges weights are presenting the weights ( $w_{ij}$ ) of the interactions between the adjacent particles.

Two large three dimensional Ising Model Max-Cut problems were taken from the "The DIMACS library of mixed semidefinite-quadratic-linear programs" problems challenge web site (URL: <http://dimacs.rutgers.edu/Challenges/Seventh/Instances/> ). In these problems the three dimensional grid is in a torus shape and the weights are randomly generated according to two types of distributions: Gaussian distribution and binary ( $\pm 1$ ) valued (based on  $p=0.5$  Bernoulli distribution). In this paper we will refer to the problems with  $8^3$  grid size. These problems data sets are represented by a graph of 512 nodes and 1536 edges. The best results were attained by Franceschini & Mannino using Simulated Annealing [2].

Both input data vectors were downloaded from the DIMACS site and processed into a matrix input for the Max - Cut solving methods. A representation of the input matrices for both problems appears in figure #2.

Figure #2: Representation of the input matrices for both max-cut problem



These problems were tested against two stochastic optimization methods: Cross Entropy (CE) and Parametric Minimum Cross Entropy (PME).

### 3.1. The Cross Entropy Method:

The Cross-Entropy (CE) method attributed to Reuven Rubinstein is a general Monte Carlo approach to combinatorial and continuous multi-extremal optimization and importance sampling. The method originated from the field of rare event simulation, where very small probabilities need to be accurately estimated, for example in network reliability analysis and queueing models. The CE method can be applied to static and noisy combinatorial optimization problems such as the Traveling Salesman Problem, the Quadratic Assignment Problem, the Max-Cut problem and the Buffer Allocation Problem, as well as continuous global optimization problems with many local extrema.

Generally the CE method consists of two phases:

1. Generate a random data sample (trajectories, vectors, etc.) according to a specified mechanism.
2. Update the parameters of the random mechanism based on the data to produce a "better" sample in the next iteration. This step involves minimizing the cross-entropy or Kullback-Leibler divergence.

The basic algorithm for optimization of Max-Cut problems is detailed in Table #1.

Table #1: Basic Cross Entropy Method Algorithm for Optimization

1. Define $v_0 = u$ . Set $t = 1$ (Iteration counter).
2. Generate a sample $X_1, \dots, X_N$ from the density $f(\cdot; v_{t-1})$ and compute the sample $(1 - \rho)$ quantile $\gamma_t$ of the performances according to $\gamma_t = S([ (1 - \rho)N ])$ .
3. Use the same sample $X_1, \dots, X_N$ and solve the stochastic program: $\max_v \frac{1}{N} \sum_{i=1}^N I_{\{S(x_i) \geq \gamma_t\}} \ln f(x_i; v)$ Denote the solution by $v_t$ .
3.1. For Bernoulli distribution (used for example for Max-Cut problems) the updating scheme is reduced to: $p_j = \frac{\sum_{i=1}^N I_{\{S(x_i) \geq \gamma_t\}} x_{ij}}{\sum_{i=1}^N I_{\{S(x_i) \geq \gamma_t\}}}$
4. Use smoothing scheme to update the PDF parameters: $v_t = \alpha v_t + (1 - \alpha) v_{t-1}$
4.1. For Bernoulli distribution (used for example for Max-Cut problems) the smoothing scheme is: $P_{j,t} = \alpha \left( \frac{\sum_{i=1}^N I_{\{S(x_i) \geq \gamma_t\}} x_{ij}}{\sum_{i=1}^N I_{\{S(x_i) \geq \gamma_t\}}} \right) + (1 - \alpha) P_{j,t-1}$
5. If stopping condition reached – Stop. Else return to 2.

Frequently used stopping conditions for the CE methods are: number of iterations with no improvement of the best score, convergence of the distribution functions into degenerated distribution and maximum number of iterations as safety margin.

### 3.2. The Parametric Minimum Cross-Entropy (PME) Method

The PME method is a parametric method to solve the well known Kullback Minimum Cross Entropy (MinxEnt) problem. This method was shown [5] to provide good results in both optimization and #P complete counting problems. Similar to Cross-Entropy (CE), the PME algorithms first casts the underlying counting problem into associate rare-event probability estimation, and then, solving the PME program, it finds the optimal parameters of the importance sampling distribution, to estimate efficiently the desired quantity.

The MinxEnt problem is formulated as:

$$(1) \quad \begin{aligned} \min_{\mathbf{p}} \{D(\mathbf{p}, \mathbf{u})\} &= \min_{\mathbf{p}} \left\{ \sum_{j=1}^m p_j \ln \frac{p_j}{u_j} \right\} \\ \text{s.t.} \\ E_{\mathbf{p}} S(\mathbf{X}) &= \gamma, \\ \sum_{j=1}^m p_j &= 1 \end{aligned}$$

The simulation based procedure to parametrically solve this problem computes the following equation (2), at each iteration, to extract the estimated value of the LaGrange coefficient:  $\hat{\lambda}_t$ :

$$(2) \quad \frac{\sum_{k=1}^N S(\mathbf{X}_k) \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)}{\sum_{k=1}^N \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)} = \hat{\gamma}_t$$

The estimated value of the LaGrange coefficient:  $\hat{\lambda}_t$  is used at each iteration to solve the following equation (3) used for optimization. Equation (3) updates the marginal distribution of the parameters in the probabilities vector, to converge into the optimal parameters.

$$(3) \quad \tilde{p}_{t,j} = \frac{\sum_{k=1}^N I_{\{X_{ki}=1\}} \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)}{\sum_{k=1}^N \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)}$$

### 3.3. Convergence Properties of the Cross-Entropy Method

The smoothing parameter used by both methods stands in the center of several convergence results for the CE method. Following we will present few of the main results from Costa et al. [4] paper which we introduced into the problem at hand to empirically test them on the problem convergence properties. The first property derived from Costa et al. [4] is the following: The optimal solution is generated eventually by the CE algorithm with probability 1 if the smoothing sequence  $\{\alpha_t\}_{t=1}^{\infty}$  satisfies the condition:  $\sum_{t=1}^{\infty} \prod_{m=1}^t (1 - \alpha_m)^n = \infty$ .

A slightly less strong result that was proved by Costa et al. [4] is the following: If the smoothing sequence is a constant, with  $\alpha_t = \alpha$ ,  $\alpha \in (0, 1]$ , and  $p_{0,i} \in (0, 1)$  for all  $i$ , then the sequence of probability mass functions  $f(x; p_t)$ ,  $t \geq 1$ , converges with probability 1 to a unit mass located at some (random) candidate  $x \in X$ . For sufficiently small smoothing parameter  $\alpha$  the following result was proved: The probability that the optimal solution is generated can be made arbitrarily close to 1 by selecting a sufficiently small value of  $\alpha$ .

Later in the results section we will show how the above properties were used to improve convergence of both methods and discuss their impact in the results discussion section.

### 3.4. Implementation Notes:

The Cross Entropy method implemented was adopted from Rubinstein & Kroese [3]. As detailed above the only parameters used by the Cross Entropy Method are the samples number (N) the elite-samples percentile ( $\rho$ ) and the smoothing parameter ( $\alpha$ ). Number of samples that was chosen initially was around two times the size of the problem (1000). For the more challenging problem larger sample number was taken up to five times and more the problem size (2500-5000). Several elite samples quantiles ( $\rho$ ) were tested. Usually this parameter was fitted to the other parameters to allow convergence. Three different smoothing schemes were tested:

- 1) Constant smoothing parameter: attempting smaller smoothing parameter ( $\alpha$ ) for the challenging problems.
- 2) Decreasing smoothing parameter at each iteration according to:  $\alpha = 1/(j/10 + 1)$  adopted from Costa et al. [4].
- 3) Smoothing augmented with small percent (0.01-0.02) of the initial (unified) probabilities vector.

A combination of several Stopping conditions was used:

- 1) Probabilities vector convergence into degenerated distribution.

- 2) Threshold number of iterations with no progress in maximum result reached.
- 3) Maximum number of iterations.

Parametric Minimum Cross Entropy (PME) implemented was adopted from Glynn et al. [5]. The Lagrange Coefficient estimator  $\lambda$  parameter was computed from the equation:

$$\frac{\sum_{k=1}^N S(\mathbf{X}_k) \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)}{\sum_{k=1}^N \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)} = \hat{\gamma}_t. \text{ The } \gamma_t \text{ values were taken from the elite sample quantile}$$

function value. MATLAB solvers (e.g. "fzero") were used to extract the  $\lambda$  value from the above equation. Several numerical adaptations were required to ensure that the value will not reach MATLAB limits. Using the  $\lambda$  value computed, the probabilities vector was updated according to

the following equation: 
$$\tilde{p}_{t,i} = \frac{\sum_{k=1}^N I_{\{X_{ki}=1\}} \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)}{\sum_{k=1}^N \exp(-S(\mathbf{X}_k) \hat{\lambda}_t)}.$$

#### 4.1. Results for the TSg38 Problem

The Best Known solution (BKS) for the above problem as presented at the DIMACS website [2] is 391.11654.

1. Chart #1 details few results achieved by CE for the TSg38 problem by various parameters used for CE.
2. Chart #2 details few results achieved by PME for the TSg38 problem by various parameters used for PME.

Chart #1: CE Results for the TSg38 problem

Run type.	Smoothing parameter ( $\alpha$ )	Elite Samples ( $\rho$ ), Samples number (N)	Solution Reached	Percentage from BKS	CPU Time (sec.)	Iterations
1	0.7	0.1, 1000	391.66438	100.14%	105.32	64
2	0.6	0.1, 1000	385.26243	98.5%	119.09	74
3	0.7	0.2, 1000	394.45717	100.85%	104.09	65
4	0.7	0.2, 1000	395.71476	101.18%	133.11	83
5	0.7	0.1, 2500	399.9528	102.26%	245.48	63

Chart #2 PME Results for the TSg38 problem

Run type.	Smoothing parameter ( $\alpha$ )	Elite Samples ( $\rho$ )	Solution Reached	Percentage from BKS	CPU Time (sec.)	Iterations
1	0.7	0.1	386.8223	98.9%	393.6	96
2	0.7	0.2	390.19732	99.76%	418.12	101
3	0.7	0.2	401.40524	102.63%	657.93	163
4	0.8	0.1	385.5784	98.58%	400.79	96
5	0.6	0.1	396.53643	101.39%	503.4	126
6	0.6	0.2	395.03798	101%	757.48	189
7	0.6	0.2	401.1107	102.55%	674.54	169

\*All results in Chart #2 were achieved using N=1000 samples.

It is noticeable that for the Gaussian distributed weights problem, both methods reached better than Best-Known Solution even with the basic tutorial values of the CE parameters. It is also noticeable that PME tends to reach better results than CE, although it converges is slower.

#### 4.2. Results for the TSpm3850 Problem

The Best Known solution (BKS) for the above problem as presented at the DIMACS website [2] is 458. The binary valued weights problem proved to be a more challenging problem for both heuristic methods. As is shown in the charts below, the basic parameters values reached bad solutions enforcing adjustments of the parameters to improve the convergence: enlarging the sample size, reducing the smoothing parameter ( $\alpha$ ) and even using a converging series of smoothing parameters as adopted from Costa et al. [4].

1. Chart #3 details few results achieved by CE for the TSpm3850 problem by various parameters used for CE.
2. Chart #4 details few results achieved by PME for the TSpm3850 problem by various parameters used for PME.

Chart #3: CE Results for the TSpm3850 problem

Run type.	Smoothing parameter ( $\alpha$ )	Elite quantile ( $\rho$ ), Samples number (N)	Solution Reached	Percentage from BKS	Iterations	CPU Time (sec.)
1	0.7	0.2 ,1000	398	86.9%	73	114.47
2	0.7	0.2 ,1000	410	89.52%	71	121.41
3	0.6	0.1 ,1000	406	88.65%	67	107.4
4	0.6	0.1 ,2500	410	89.52%	76	297.17
5	0.6	0.05 ,2500	422	92.14%	64	258.08
6	0.5	0.1 ,2500	404	88.2%	90	353.26
7	0.5	0.05 ,2500	412	89.96%	76	299.06
8	0.6	0.01 ,5000	412	89.96%	78	612.60
9	0.7	0.2 ,5000	420	91.7%	94	733.66
10	0.2	0.02 ,5000	408	89.08%	141	1102.1
11	$1/(j/10+1)^{1.1}$ *	0.02 ,5000	414	90.39%	208	6421.23
12	0.2, 0.01**	0.02 ,**5000	410	89.52%	593	5003.02
13	0.2, 0.02**	0.02 ,**5000	428	93.45%	1031	8672.7

\* At run #11 smoothing scheme used adaptive parameter value adopted from [4].

\*\* At runs 12 and 13 the third smoothing scheme was used (adding 1-2% of the initial probabilities vector). The number of samples was also increased iteratively similar to the Fully Adaptive CE (FACE) adopted from [3].

As is noticeable from chart #4 below, PME reached better results than CE, although converging more slowly.

Chart #4: PME Results for the TSpm3850 problem

Run type.	Smoothing parameter ( $\alpha$ )	Elite quantile ( $\rho$ ), Samples number (N)	Solution Reached	Percentage from BKS	Iterations	CPU Time (sec.)
1	0.7	0.1 ,2500	398	86.9%	108	437.81
2	0.6	0.2 ,2500	404	88.21%	95	377.74
3	0.5	0.1 ,2500	420	91.7%	133	559.24
4	0.4	0.1 ,5000	400	87.34%	155	1267.26
5	0.2	0.05 ,5000	416	90.82%	212	1698.04
6	0.1	0.02 ,5000	412	89.96%	345	2788.37
7	0.1	*0.02 ,5000	412	89.96%	548	2323.9
8	0.2, 0.02**	0.02, 5000**	420	91.7%	326	2646.87
9	0.2, 0.02**	0.02, 5000**	422	92.14%	465	3861.53
10	0.2, 0.02**	0.1, 5000**	430	93.89%	808	8569.6
11	0.4, 0.02**	0.1, 10000**	432	94.32%	616	13236.4
12	0.2, 0.02**	0.1, 10000**	434	94.76%	707	15772.04
13	0.2, 0.02**	0.1, 20000**	426	93.01%	664	21369.59

\* At run #7 elite samples quantile was increased (by one) at each iteration.

\*\* At runs 8 -13 the third smoothing scheme was used (adding 1-2% of the initial probabilities vector, percentage reduced by half iteratively every 200 iterations). The number of samples was also increased iteratively similar to the Fully Adaptive CE (FACE) adopted from [3].

### 5. Results Discussion – Convergence properties

The following empirical conclusions were derived from this Max-Cut problem concerning adjusting the few parameters used by CE to improve convergence and optimization results in challenging problems:

Enlarging the number of samples (N) almost always improve convergence of CE and optimization results. By theory, infinite number of samples will ensure a global solution. However, it degrades the powerful property of CE of swift convergence into an exhaustive search. In fact enlarging the number of samples to  $10^5$  enabled the best results achieved by CE for the more challenging problem, but it took a full day to reach these results. Another option to use the number of sample parameter (N) is similar to the Fully Automatic CE (FACE) algorithm presented in [3]. At any iteration with no improvement of the best result the number of samples

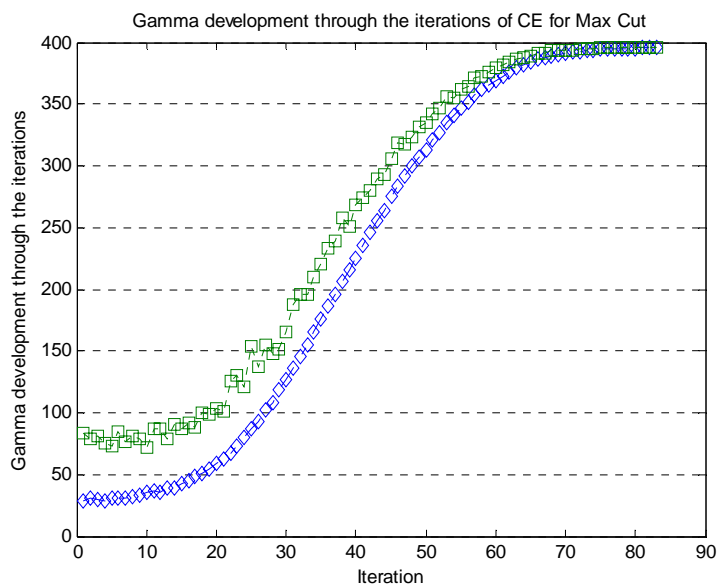
was increased. This not only enlarged the sample size but consequently enlarged the elite sample size, thus increasing the probability to sample better results.

Choosing the value of the elite percentile ( $\rho$ ) is a trade-off between the convergence rate and the probability to escape convergence into local optima. Very small percentile of elite samples ensures that only really the best results will influence the improvement of the probabilities vector in CE. Large percentile of elite samples might reduce the ability of the CE to converge into good results. However, as mentioned above, enlarging the elite samples percentile improves the probability of introducing new solutions into the convergence scheme of the CE.

The influence of the smoothing parameter ( $\alpha$ ) was detailed extensively by Costa et al. [4]. As shown in the results section above, smaller smoothing percentage sometimes enabled reaching better results. Using a small percentage of the initial probabilities vector also improved results. Using a large percentage of the initial probabilities vector could slow the convergence progress dramatically. Reducing the percentage every number of iterations accelerates the convergence. This property will be shown below in figure #5.

The convergence behavior of the CE method through the iterations is shown in figure #3 which represents a typical S shaped graph of the CE type methods.

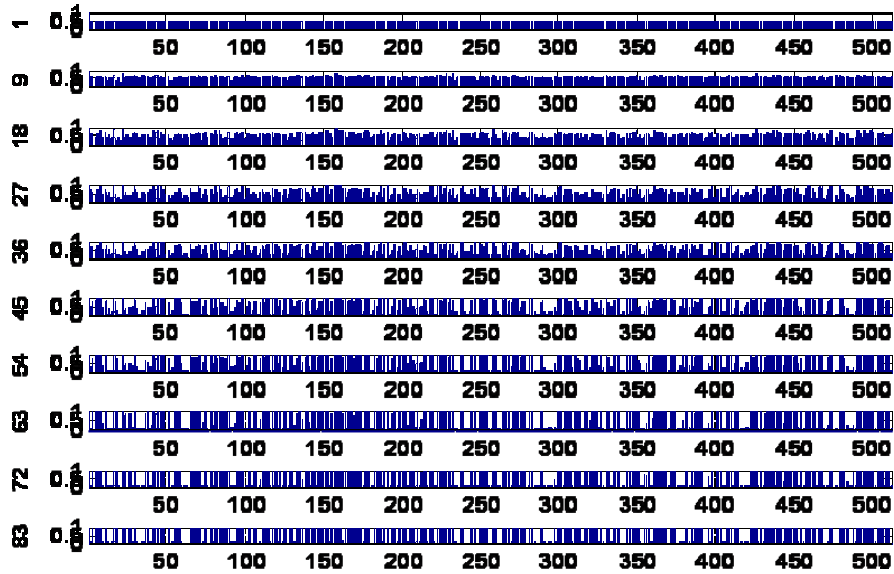
Figure #3: Typical Convergence Dynamics of CE through the Iterations for run #5 of CE for the TSg38 problem



The S shaped graph shown in figure #3 above represents slow progress at the first iterations. This behavior could have negative implications with the stopping rule which counts the number of iterations with no progress. As the process convergence towards a degenerate state of the probabilities vector, at the end of the run, a slow progress of the solution development is

typical. The degenerate status of the probabilities vector causes most of the samples to be similar and the gamma ( $\gamma$ ) results percentile value to converge closer to the best result value, as is noticeable from figure #3 above. Unfortunately this convergence property is true also in convergence into a local optimum. This convergence behavior is also noticeable in figure #4 below which gives example of the probabilities vector status through the iterations.

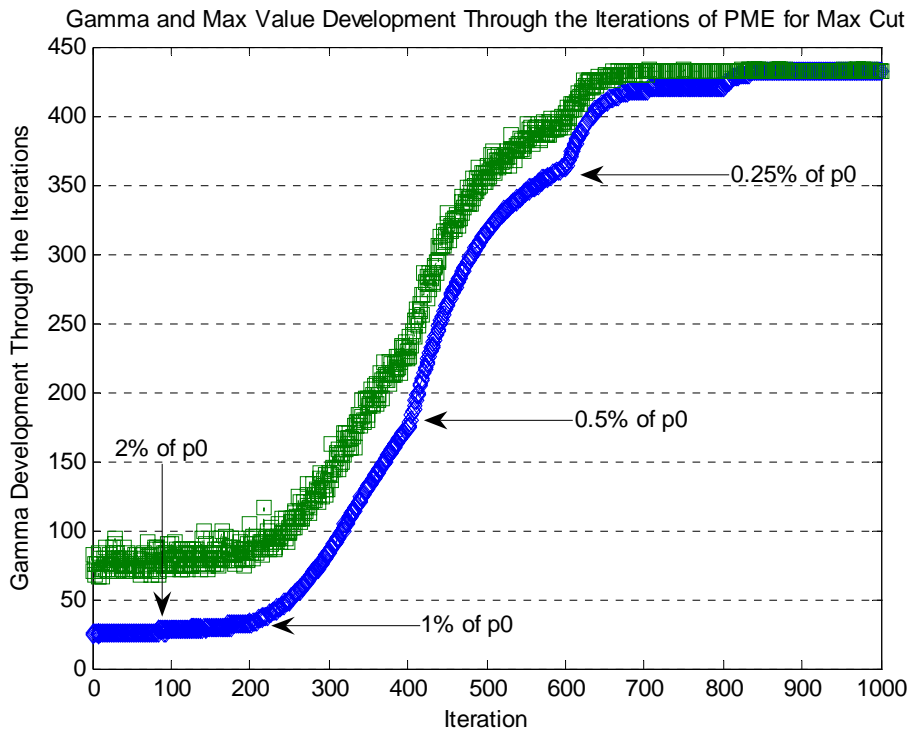
Figure #4: CE Convergence Development through the Probabilities Vector



It is noticeable in figure #4 above that toward the end of the run most of the probabilities degenerates into either zero or one. The resulting situation samples almost the same solutions in all the samples.

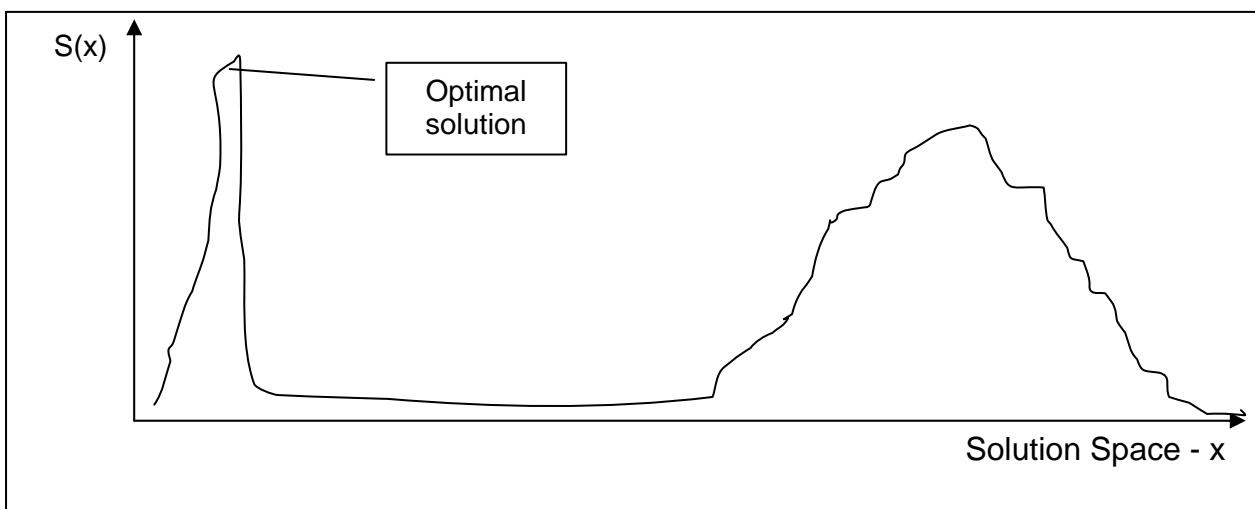
Figure #5 presents convergence properties of the PME method. It is noticeable that PME dynamic convergence behavior is quite similar to CE. In the run presented in figure #5 the use of small percentage of the initial probabilities was incorporated inside the smoothing scheme. Note that as expected, decreasing the percentage of using the initial probabilities vector increases convergence speed.

Figure #5: Typical Convergence Dynamics of PME using the third smoothing scheme, through the Iterations for run #12 of PME for the TSpm3850 problem



The TSpm3850 problem proved to be a real challenge for the CE and the PME method. Because of the statistic nature of the CE method, typical kinds of problems could prove challenging to CE. When the best results are found very far on the solution space from good results, the CE method tends to converge toward the good results and not the best ones. Figure #6 is a hand sketch example of challenging problems for the CE method.

Figure #6: Typical Example of a Challenging Problem for CE



The convergence of the CE and PME methods through the iterations increases the probability to continue to proceed in the same direction and reduces the probability to find

solutions in the other regions of the solution space. Thus for this type of problems, it could be difficult for CE to find optimal solution, due to the use of statistics results.

This challenge could generally be overcome by using the 3rd smoothing scheme: using a small percentage of the initial probabilities vector inside the smoothing scheme constitute a continuous positive probability to return to each state of the solution space, throughout the convergence process. A non-zero probability to return to every state turns a Markov chain to become irreducible. The Markov chain thus becomes ergodic, which similar to Simulated Annealing [6], ensures that the optimal solution could be found with probability equals one.

## 6. Summary

Two powerful heuristics optimization methods: CE and PME were tested against two challenging Max-Cut problems from the DIMACS web site. For the Torus Set g\_3\_8 problem, the CE method reached a solution which is better than the Best Known Solution (BKS) by 2.26% and PME reached a solution which is better than the BKS by 2.63%. For the Torus Set pm\_3\_8\_50 problem both methods reached solutions around 5-6% less than the BKS.

Comparison between the new PME method and CE yields that in general PME reaches better results than CE, however is a little slower. Several tools suggested for CE implementers facing difficult problems: Computing correctly number of samples required for problem size, Costa's smoothing parameters decrease, initial probabilities vector smoothing and selection of appropriate stopping conditions.

We consider these results to be most practical for CE method implementers since the same framework of solving Max-Cut problems using the CE method, is used in many other optimization problems through binary coding of the decision variables.

## Acknowledgments

The author wishes to thank Professor Reuven Rubinstein from the IE faculty in the Technion on his guidance and useful remarks. The author also wishes to thank Andrey Dolgin from the IE faculty in the Technion for his help on the PME method.

## References

- [1] Cipra B. A. **The Ising model is NP-complete.** In *SIAM News*, Vol. 33, No. 6.
- [2] Pataki G. and Schmieta S. **The DIMACS library of mixed semidefinite-quadratic-linear programs.** 2002. URL: <http://dimacs.rutgers.edu/Challenges/Seventh/Instances/lib.ps>
- [3] Rubinstein, Y. R. Kroese, D. P. **The Cross Entropy Method - A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning.** Springer, 2004. p. 30-201.
- [4] Costa A., Jones, O. D. and Kroese D. **Convergence Properties of the Cross-Entropy Method for Discrete Optimization.** *Operations Research Letters*, 2007.
- [5] Glynn, P. W., Dolgin A., Kroese D. P., and Rubinstein Y. R. **Parametric Minimum Cross - Entropy Method for Counting the Number of Satisfiability Assignments.** 2007. URL: <http://iew3.technion.ac.il/Home/Users/ierrr01/minxent-glynn.pdf>
- [6] Bertsimas D. and Tsitsiklis J. **Simulated Annealing.** In *Statistical Science*. Vol. 8, No. 1. 1993. pp. 10-15.