Design index-based hedging

Bundled loss property and hybrid genetic algorithm

Frank Xuyan Wang
Validus Research Inc.
Waterloo, ON N2J1R1 Canada
frank.wang@validusresearch.com

***Abstract*—For index-based hedging design, the scatter plot of the hedging contract losses versus the losses to be hedged is generally used to visualize and quantify basis risk. While studying this scatter plot, which does not cluster along the diagonal as desired, a "bundled loss" phenomenon is found. In a setting where both the hedging and the hedged contracts have 100,000 years of simulated losses, this shows that if we need to hedge one loss in a year for the hedged contract, we may need to pay for other losses in other years in the hedging contract, which are unnecessary and unwanted. The reason is that the index used in the hedging may have identical loss values in different years while the hedged contract may not. This finding is a guiding principle for forming the risk measures and solution frameworks. To solve the problem so formed, a hybrid multi-parent and orthogonal crossover genetic algorithm, GA-MPC-OX, is used and pertinent adjustments are studied. For a problem with hundreds of dimensions, using eleven parents seems best, while a problem with tens of dimensions would prefer nine parents. Depending on the dimensions, relevant best strategies of the orthogonal crossover are also suggested by experimental results. To combat the stagnation of the algorithm, the perturbation by Lévy stable distribution is studied. This reveals possible effective parameters and forms. Numerical comparison with other algorithms is also conducted that confirms its competence for the hedging problem.**

Keywords*—hedging problem; genetic algorithm; multi-parent crossover; orthogonal crossover; Lévy stable distribution

I. Introduction

In the reinsurance industry, we frequently need to mimic a client company's losses by an insurance industry loss index, which are functions of the collective losses from all insurance companies across all geography, peril, and line of business. The latter is used to construct index-based hedging contracts for the client .

If the index loss is an accurate approximation of the client's actual loss, we should naturally expect their scatter plot closely clustered along the diagonal, their expected losses around the same, and their empirical CDF and PDF plots not far apart. More specifically, we would want the risk as given by quantiles of the loss differences distribution for various probabilities, or by probabilities of these differences above given losses on condition that the client's losses greater than a list of thresholds, are within some expected limit, since these differences, especially where the losses from the client contract are above the losses from the index-based contract, are the residual risk of un-hedged losses. The expected loss of the index-based contract is a key determinant of the cost of the hedging. So we first attempt to quantify the effectiveness of the hedge. Next, we introduce methods to optimally balance the effectiveness and the cost of hedging.

The accompanying mathematical problem is finding the forms of the function and the objective value function we should use for the index, as well as which algorithm or problem-related adjustment we should adopt for solving the hedging problem.

In a previous study of the insurance-linked securities portfolio optimization [1], we found a domain specific property that many of the candidate contracts are either best or worst and their contribution should be kept constant, and a hybrid multi-parent crossover, orthogonal crossover genetic algorithm and catfish algorithm, GA-MPC-OX, which can utilize said property, and through numerical comparison studies, established its superiority. For the hedging problem, we also found a "bundled-loss" property, which worked as a guiding principle in forming our solution framework, and in selecting and evolving algorithms for solving it. We will then check its efficiency by comparing to results from using such algorithms as the Firefly Algorithm, Bat Algorithm, Cuckoo Algorithm, Flower Algorithm [6], and Wind Algorithm [5].

II. Hedging Problem Solution

A. Bundled-loss

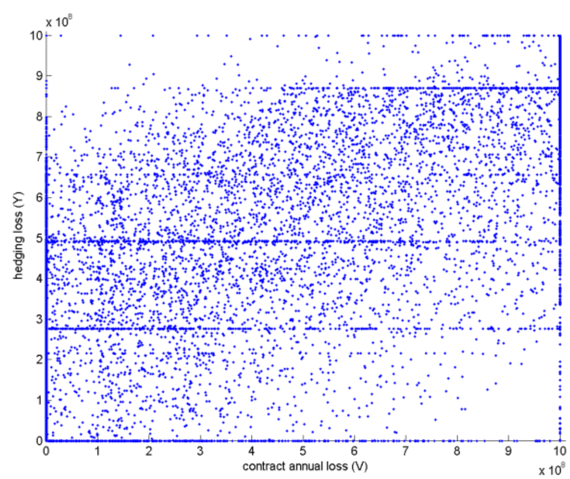

Fig. 1. Why does the scatter plot not cluster along the diagonal? The x-axis is the contract annual loss, and the y-axis is the hedging annual loss, in the same year, for a simulated set of 100,000 years.

In our experiment, the scatter plot of index loss vs. client loss never clusters along the diagonal within a narrow band. We always see the points spread out horizontally, such as in Fig.1.

A close examination of the points reveals a “bundled loss” phenomenon in the hedging problem, and this discovery gives us empirical rules on how to address these problems.

The “bundled loss” principle can be explained most clearly in the top end of the to-be hedged client contract loss (called V), which occupies more than 30% of the non-zero loss years; and all values are the same number, 10^9. To hedge the 10^9 loss of one of these years of V with a to-be constructed industry loss portfolio, called the index or the hedging contract, the portfolio needs to have a loss of 10^9 for that year. At the same time, the portfolio (called Y, we will not differentiate between the portfolio and the index formed from it) will have many other years that have the same loss or almost the same loss as that year, possibly in the years where V have zero losses because V have zero losses in 90% of the years. The additional losses of Y are the “bundled loss”, for hedging the loss of the needed year of V, and will be the additional cost in the expected payoff of Y. Because of these bundled losses, we will see horizontal scatter in the V-Y scatter plot: same Y but different V (the special case mentioned of V=0 is in the y-axis condensation shown in our results).

To overcome these bundled losses, we need to improve the industry portfolio discrimination power: the most ideal case is Y will have different losses for different years (and we can use an unlimited number of functions of Y to cut it into slim slices, which have values 0 outside of the narrow domains). To achieve these we can use two approaches:

- Using more industry loss contracts, such as by Country and by Cresta contracts; when that is not available, we can split the industry losses by risk group, and use them as our universe.
- Using multiple portfolios rather than just one to form Y, so that each portfolio may address different portions of the V loss range (In our study we used 10 portfolios so our variable dimension is 480 plus the 10 weights of the portfolios); this strategy shifts the difficulty of the problem to that of the capability of the algorithm for finding high dimensional optimal solutions.

Since the scatter plot will always have some spill out due to the bundled loss, we should not rely on it alone for deciding whether we get the good solution; we should also use additional means for comparing or checking the results. Other than the CDF plot of V-Y on condition that V greater than some given threshold losses and the PDF plot of V and Y, we can also look at various numerical criteria.

When it is hard to discern from the scatter plot which solution is better, we can check the risk represented by 0.96, 0.99, 0.996 quantiles of abs(V-Y), we used abs since the positive part means un-hedged loss and the negative part means over-hedged loss that is the additional cost, the count of years with abs(V-Y) falling into ten different sub intervals of the range 0 to 10^9 with each band a length of 10^8, and the cost represented by E(Y), and vice versa. For example, for two solutions with similar risk (and scatter plot), the solution with smaller cost E(Y) would be better. If two solutions have similar risk and cost, then the one with higher conditional CDF plot is better.

If we want the count of years when abs(V-Y) near the higher end of 10^9 to be small, the count of years when abs(V-Y) near the lower end of 0 will be large due to the bundled loss. To get an overall number for this trade off, we need a weighting scheme to sum up these counts. We can use this number as our objective, which tested to be more robust than the p-norm or any other forms, considering that a few years' losses may be outliers out of the simulation data generating process.

The 1-norm, 2-norm, or other higher p-norm appear to be more sensitive to outliers in the data, and will produce solutions with bouncing Y PDF while the original V PDF is very smooth and slowly-changing. Various weighting schemes are tested with some exponential function the best.

Using the count of years abs(V-Y) falling to different intervals in objective value function increased the stability but also may not differentiate solutions accurately. Two solutions with objective values so constructed and differing by a few percent may not have the property that the better solution will have the better objective value, especially when the function forms used are different. We then need to consider and compare all the different measures discussed above.

As for the forms or the formulas that define Y, we get these empirical principles: the best form is mimicking the V payoff function, i.e., using event limit, annual limit and big annual aggregate deductible (better than any piecewise linear, piecewise constant, power, and other highly nonlinear functions such as combing min and max functions); when designing the multiple portfolios, each portfolio is better to cover a different loss range and has tens of times less non-zero loss years than V.

There is also another empirical finding of giving different weighting to $V>Y$ and $V<Y$ losses: if we consider the un-hedged $V>Y$ portion as worse than the over-hedged portion where $V<Y$, and want to give a relative large factor to it, we then can find that we will reduce the un-hedged conditional probability such as $1-F(V-Y<=4e8|V>7.5e8)$. At the same time it will increase the cost as given by E(Y). The relationship between the factor and the probability is almost linear. For example, when using a factor of 1.1, we get a probability of 1-0.81 and when we use a factor of 1.35, we get a probability of 1-0.85.

When the objective values of solutions only differ by a small percentage, their risk measures and costs, as well as the scatter plots, CDF and PDF plots will be similar. It seems that a decrease in some portions of the risk curve will be offset by increases in other portions. However, to reveal some emerging pattern in the solution form, the tiny difference matters. Only the best solution has a low enough noise to show the true figure.

The search for good Y form and a good algorithm is a reciprocal process. We do not want the noise in the algorithm to affect the decision about which form is better, and we want the adjustment of the algorithm to be pertinent to the objective function form. So we first fix several forms of the Y and test the algorithm; the settings that are constantly better are adopted, and with the new algorithm, we test more Y forms using the criteria of various plots and abs(V-Y) counts. This process is then repeated.

B. GA-MPC-OX adjusted

In [1], the GA-MPC-OX algorithm, which performed better than any of the other algorithm tested for the portfolio optimization problem, such as PSwarm, MOEAD, ENSMOEAD, DyHF, CMODE, ICDE, PSO-DE, DSA, DECC-G, CoDE, ETLBO, OXDE, MBA, IRM-MEDA, TLBO, MMEA, RM-MEDA, ABC, IABC, NBIPOPaCMA, SHADE_CEC2013, DRMA-LSCh-CMA, and iCMAES-ILS, [10]-[30], is proposed. This prompts us to adopt it to the hedging problem.

Numerical experiments showed that the GA-MPC-OX can become stagnant easily, so we adjusted the number of parents used in its crossover operator. We found that the strategy of using nine parents generating nine children is best in a problem with 59 free-to-change variables, followed by eleven parents, and then by six, seven, or fourteen parents. These numbers seem related to the dimension, for example, for problem with 490 dimensions, using eleven parents is best.

The Catfish Algorithm from [3] as used in [1] is akin to the dominance property of the portfolio optimization problem, and may not work in our hedging problem. Replacing it with the original normal perturbation operator from [4] produced better results.

The original interpolation method for orthogonal crossover in [2] tested better than [1]'s shortcut of table lookup method. So we followed the original method. However, instead of using Catfish Algorithm's method of taking candidates from the lower half, we took those from the upper half of the candidates pool. For the levels used, we tested methods of using increasing, decreasing, or equal probabilities of selecting a level from a number of levels. The three best strategies are using three levels with equal probabilities, always taking three levels, and using six levels with equal probabilities. But for higher dimensions, the last strategy seems best, followed by the first and then the second; adding randomness in selecting levels appears more effective.

With the four combinations of parent and levels numbers when each selected the best two, for 58 dimensional problems, the precedence is, (9,3),(9,6),(11,3), and (11,6). For 590 dimensional problems the order is reversed.

Five other algorithms are used to solve the same problem as ours, the WDO [5], Firefly Algorithm, Bat Algorithm, Cuckoo Search Algorithm, and Flower Pollination Algorithm [6]. In one form of the objective function, which is a weighted sum of the counts of the differences in loss belonging to different intervals, so that the smaller the objective value, the better the hedging should be, our algorithm finds the objective value after 100,000 function evaluations of 100,394, and after 1.4 million function evaluations of 64,169. The WDO gets the objective value after 100,000 function evaluations of 262,557, 2.61 times our number, and is stagnant after 40,000 function evaluations.

For another form of the objective function, our algorithm finds the objective value after 150,000 function evaluations of 109,551, after 200,000 function evaluations of 101,110, after 500,000 function evaluations of 85,781, and after one million function evaluations of 83,648. The Firefly Algorithm finds the objective value after 200,000 function evaluations of 372,697, stagnant after 150,000 function evaluations. The Bat Algorithm gets the objective value after 160,000 function evaluations of 327,017. The Cuckoo Search Algorithm gets the objective value after one million function evaluations of 130,405, 55.9% larger than that of 83,648. The Flower Pollination Algorithm gets the objective value after 500,000 function evaluations of 126,181, 47% larger than that of 85,781.

These comparisons may not show the superiority of our algorithm due to their example implementation, but they do show the importance of fine adjusting of the strategies and parameters used. Dr. Yang [6] emphasized the benefit of Lévy stable distribution, so we will try applying it in our algorithm.

C. Gauss or Lévy

The normal perturbation in the original GA-MPC algorithm is of the form 0.5U+0.25U*N, where U is the uniform distribution in (0,1) and N is the standard normal distribution. The Lévy flight perturbation Dr. Yang used is of the form 0.01N*S(1.5,0)*(x-best), where S(1.5,0) is the Lévy alpha-stable distribution (http://en.wikipedia.org/wiki/Stable_distribution) with stability parameter 1.5, skewness parameter 0, scale parameter 1, and location parameter 0.

We performed three runs and saw one run using Lévy flight perturbation obtained better results than when using normal perturbation while the other two runs were worse. It seems the Lévy flight has effect but it is not trivial to harness its power, or it is purely caused by chance and more due to the randomly selected initial population. We tested on the following additional forms of the perturbation: U-0.5+0.25U*N, U-0.5+0.25U*tan(π*(U-0.5)), U-0.5+0.25U*S(0.5,0), 0.5U+0.25U*tan(π*(U-0.5)), 0.5U+0.25U*S(0.5,0), 0.5U+0.25U*S(α,0), α*N*S(0.5,1), α*(U-0.5)*S(0.5,1), and α*N*S(0.5,0), using the stable distribution code from [7], since it is faster than the other two implementations [8] and [9]. The test results are collected in Table I.

TABLE I. EFFECTS OF PERTURBATION FORMS

Perturbation Form	Objective Value
0.5U+0.25U*N	58369.3129228756[a]
0.01N*S(1.5,0)*(x-best)	58588.2094140444
U-0.5+0.25U*N	58626.7825481966
U-0.5+0.25U*tan(π*(U-0.5))	59253.6826758084

Perturbation Form	Objective Value
U-0.5+0.25U*S(0.5,0), 1st run	58417.5946349593
U-0.5+0.25U*S(0.5,0), 2nd run	58836.9978332785
U-0.5+0.25U*S(0.5,0), 3rd run	58903.4202994694
0.5U+0.25U*tan(π*(U-0.5))	58454.9400481151
0.5U+0.25U*S(0.01,0)	58533.4400598268
0.5U+0.25U*S(0.05,0)	58365.6438802964
0.5U+0.25U*S(0.1,0)	58487.5490576022
0.5U+0.25U*S(0.3,0)	58846.7701952340
0.5U+0.25U*S(0.5,0)	58437.9835642726
0.5U+0.25U*S(0.7,0)	58921.7782284960
0.5U+0.25U*S(0.9,0)	58595.5569928767
0.5U+0.25U*S(1.01,0)	58721.6837996683
0.5U+0.25U*S(1.05,0)	58356.6284969962
0.5U+0.25U*S(1.1,0)	58396.0031511394
0.5U+0.25U*S(1.3,0)	58446.7420494971
0.5U+0.25U*S(1.5,0)	59143.4785428433
0.5U+0.25U*S(1.7,0)	58577.2452540111
0.5U+0.25U*S(1.9,0)	58334.9363537724
0.5U+0.25U*S(1.95,0)	58390.6540507207
0.5U+0.25U*S(1.99,0)	58798.0866233763
0.01*N*S(0.5,1)	58623.7008885335
0.01*(U-0.5)*S(0.5,1)	59379.2371618059
0.01*N*S(0.5,0)	58489.2841485367
0.05*N*S(0.5,0)	58782.6471115213
0.075*N*S(0.5,0)	58468.6686062425
0.1*N*S(0.5,0)	58412.1454721504
0.15*N*S(0.5,0)	58472.1776203138
0.2*N*S(0.5,0)	58399.2953524066
0.25*N*S(0.5,0)	58465.769182205
0.3*N*S(0.5,0)	58874.2030665987
0.35*N*S(0.5,0)	58653.8876928435
0.4*N*S(0.5,0)	59082.81743882
0.45*N*S(0.5,0)	58556.6515104805
0.5*N*S(0.5,0)	59047.7318800064
1*N*S(0.5,0)	59020.9476755368
0.1*N*S(0.05,0)	58442.6398141161

Perturbation Form	Objective Value
0.1*N*S(1.05,0)	58436.1462185919
0.1*N*S(1.5,0)	59154.7577594512
0.1*N*S(1.9,0)	59028.6114845251

a. 58 dimensional problem, used 9-parent MPC and 3-levels orthogonal crossover operators.

Out of all the tested cases, the Gauss or normal distribution used by the original GA-MPC is at the higher quantile end, outperformed only by three cases that used general Lévy alpha-stable distribution for which the stability parameter α is near the Gauss end 2 or the Cauchy end 1, or near 0: 1.9, 1.05, and 0.05. The middle point of (1,2) 1.5 was the worst for that range, but 0.5 was the second best for the interval (0,1). Adding the symmetry perturbation term U-0.5 is not as good as adding the shifted-up term 0.5U, for the hedging problem: using more weights would match more losses with added costs. It may also be possible that our cases are mainly stochastic noises, and more experiments are needed for a definite conclusion.

III. CONCLUSION

For the hedging problem, a bundled-loss property is found, which explained why the scatter plot is always blurred and cannot be used for fine selection of the solution, except when the algorithms used are too inefficient or solutions found are deviated too much from each other. When we cannot distinguish two solutions by their scatter plot, we can still differentiate between them by other means such as using weighted counts of their differences for objective value, conditional probability plots, and etc. This property also guided us in adjusting the hybrid multi-parent, orthogonal crossover genetic algorithm GA-MPC-OX for the hedging problem, which tested far better than several example algorithms which may not have been fine-tuned or problem-tuned for performance. The normal perturbation used in the GA-MPC-OX generally performed well, but can be surpassed by some parameter Lévy alpha-stable distribution in some tests. Studies suggest some parameter values are effective, but a perturbation scheme that always performs better still requires more research.

ACKNOWLEDGMENT

F.X.Wang thanks Dr. Lixin Zeng for forming the hedging problem framework and guiding the research direction through questions, comments, suggestions, and expectations. All this work was done to fill in the details of that framework, answer the aforementioned qquestions, and explain the observed discrepancy.

REFERENCES

[1] Frank Xuyan Wang, “Relay Optimization Method,” May 2014. Available: http://www.optimization-online.org/DB_FILE/2014/05/4345.pdf

[2] Y. Wang, Z. Cai, and Q. Zhang, “Enhancing the search ability of differential evolution through orthogonal crossover,” Information Sciences, vol. 185, no. 1, pp. 153-177, 2012.OXDE.

[3] Li-Yeh Chuang, Sheng-Wei Tsai, Cheng-Hong Yang, “Chaotic catfish particle swarm optimization for solving global numerical optimization problems,” Applied Mathematics and Computation, 217 (2011) pp6900–6916.

[4] S. M. Elsayed, R. A. Sarker, D. L. Essam, “GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems,” in Proc. IEEE Congr. Evol. Comput. (CEC), 2011, pp. 1034–1040.

[5] Z. Bayraktar, M. Komurcu, J. A. Bossard and D. H. Werner, "The Wind Driven Optimization Technique and its Application in Electromagnetics," IEEE Transactions on Antennas and Propagation, Volume 61, Issue 5, pages 2745 - 2757, May 2013. Available: http://wdo.cloudturkiye.com/wdo_matlab_03.m

[6] Xin-She Yang, Nature-Inspired Metaheuristic Algorithms, Second Edition, Luniver Press, (2010). Available: http://www.mathworks.com/matlabcentral/fileexchange/authors/119376

[7] Rafal Weron, "STABLERND: MATLAB function to generate random numbers from the stable distribution," (2010.04.26). Available: http://ideas.repec.org/c/boc/bocode/m429003.html

[8] J. Huston McCulloch, "Stable Random Number Generator," (12/18/96). Available: http://www.econ.ohio-state.edu/jhm/programs/STABRND.M

[9] Mark Veillette, "STBL: Alpha stable distributions for MATLAB," (16 Jul 2012). Available: http://www.mathworks.com/matlabcentral/fileexchange/37514-stbl--alpha-stable-distributions-for-matlab

[10] A. I. F. Vaz and L. N. Vicente, “A particle swarm pattern search method for bound constrained global optimization,” *Journal of Global Optimization*, 39 (2007) 197-219. Available: http://www.norg.uminho.pt/aivaz/pswarm/

[11] Q. Zhang and H. Li, “MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition,” *IEEE Trans. on Evolutionary Computation*, vol.11, no. 6, pp712-731 2007.

[12] Q. Zhang, W. Liu, and H. Li, “The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances,” Working Report CES-491, School of CS & EE, University of Essex, 02/2009.

[13] S. Z. Zhao, P. N. Suganthan, and Q. Zhang, “MOEA/D with an Ensemble of Neighbourhood Sizes,” *IEEE Trans on Evolutionary Computation*, (TEC) 16(3):442-446 (2012).

[14] Y. Wang and Z. Cai, “A dynamic hybrid framework for constrained evolutionary optimization,” in *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.42, no.1,2012, pp. 203-217.

[15] Y. Wang and Z. Cai, “Combining multiobjective optimization with differential evolution to solve constrained optimization problems,” in *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, 2012, pp. 117-134.CMODE

[16] G. Jia, Y. Wang, Z. Cai, and Y. Jin, “An improved (μ+λ)-constrained differential evolution for constrained optimization,” in *Information Sciences*, vol. 222, 2013, pp. 302-322.ICDE

[17] H. Liu, Z. Cai, and Y. Wang, “Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization,” *Applied Soft Computing*, vol. 10, no. 2, pp. 629-640, 2010.PSO-DE

[18] P. Civicioglu, "Transforming Geocentric Cartesian Coordinates to Geodetic Coordinates by Using Differential Search Algorithm," *Computers and Geosciences*, 46, 229-247, 2012. Available: http://www.pinarcivicioglu.com/ds.html

[19] Zhenyu Yang, Ke Tang and Xin Yao, "Large Scale Evolutionary Optimization Using Cooperative Coevolution," *Information Sciences*, 178(15):2985-2999, 2008.DECC-G

[20] Zhenyu Yang, Ke Tang and Xin Yao, "Self-adaptive Differential Evolution with Neighborhood Search," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC2008)*, Hongkong, China, 2008. SaNSDE

[21] Y. Wang, Z. Cai, and Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55-66, 2011.CoDE

[22] R. V. Rao, V. Patel, “An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems,” *International Journal of Industrial Engineering Computations*, Volume 3, Issue 4, 2012, 535-560. Available: https://sites.google.com/site/tlborao/

[23] Y. Wang, Z. Cai, and Q. Zhang, “Enhancing the search ability of differential evolution through orthogonal crossover,” *Information Sciences*, vol. 185, no. 1, pp. 153-177, 2012.OXDE

[24] Ali Sadollah, Ardeshir Bahreininejad, Hadi Eskandar, and Mohd Hamdi, “Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems,” *Applied Soft Computing*, Volume 13, Issue 5, May 2013, pp2592–2612.MBA

[25] Y. Wang, J. Xiang, and Z. Cai, “A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator,” *Applied Soft Computing*, vol. 12, no. 11, pp. 3526-3538, 2012.IRM-MEDA

[26] A. Zhou, Q. Zhang and Y. Jin, “Approximating the Set of Pareto Optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm,” *IEEE Trans on Evolutionary Computation*, vol. 13, no. 5, pp1167-1189, 2009.MMEA

[27] Qingfu Zhang, Aimin Zhou and Yaochu Jin, “RM-MEDA: A Regularity Model Based Multiobjective Estimation of Distribution Algorithm,” in *IEEE Trans. on Evolutionary Computation*, vol. 12, no. 1, pp41-63, 2008.

[28] D. Karaboga, B. Basturk, “A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm,” *Journal of Global Optimization*, Volume:39, Issue:3,pp:459-171, November 2007. Available: http://www.mathworks.com/matlabcentral/fileexchange/27125-solution-to-economic-dispatch-by-artificial-bee-colony-algorithm/content/ABC-eld/runABC.m

[29] P. W. Tsai, J. S. Pan, B. Y. Liao, and S. C. Chu, “Enhanced artificial bee colony optimization,” *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5081–5092, 2009. IABC

[30] Special Session & Competition on Real-Parameter Single Objective Optimization at CEC-2013, Cancun, Mexico 21-23 June 2013. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm