International Journal of Hybrid Information Technology
Vol.8, No.6 (2015), pp.17-24
http://dx.doi.org/10.14257/ijhit.2015.8.6.02

Multi-criteria Decision Making Using Genetic Algorithmic Approach in Computer Simulation Models

Chandrashekhar Meshram[1] and Shyam Sundar Agrawal[2]

[1]Department of Mathematics R.T.M. Nagpur University, Nagpur, India
Department of Mathematics
[2]Vikash College of Engineering for Women, Vikash Enclave, Barhaguda Bargarh
cs_meshram@rediffmail.com, shyampkace@rediffmail.com

Abstract

In this paper, we described a genetic algorithm that allows finding the optimal configuration for a stochastic discrete events simulator when multiple performance measures have to be considered simultaneously. This type of algorithmic approach provide particularly interesting when the decision making authority is shared by multiple decision makers with conflicting priorities. The optimal solutions found with this algorithm typically represent a middle ground solution that may be acceptable to all the involved parties. The multi-criteria approach relies on an interval based variant of the Promethee method, which is combined with a feasibility score to obtain the ranking of the chromosomes within a certain generation of the genetic algorithm.

Keywords*--Multi-criteria optimization, Decision making, Genetic algorithms and Computer simulation models*

1. Introduction

The idea of evolutionary computing was introduced in 1960 by Rechenberg in his work Evolutionary Strategies. Genetic algorithms are computerized search and optimization algorithms based on the mechanics of natural selection. Prof. Holland of University of Michigan, Ann Arbor, envisaged the concept of these algorithms in the mid-sixties and published his seminal work (Holland, 1975) [4]. Thereafter, a number of students and other researcher have contributed to the development of this field.

The most of Genetic algorithms studies are available through some books by Davis [2], Goldberg [3], Holland [4], and Deb [5, 6] and through a number of conference and proceedings. The first application towards structural engineering was carried by Wang and Wang [11, 12] and Wang and Poh [13]. They applied genetic algorithm to the optimization of a ten-member plane trues.

Raj, *et al.*, [1] discussed the ranking multi-criterion river basin planning alternatives using fuzzy numbers. Reeves [7] showed the use of genetic algorithms for the operations researcher. Triantaphyllou [8], Velton and Steward [9], Lock and Pastijin [10] were discussed the Multicriteria Decision Making Methods and its application. Yeh and Deng [14] digened an algorithm for fuzzy multicriteria decision making. Yang and Soh [15] have been design intrigrated genetic programming by using fuzzy logics. Robert and Fuller [16] studied the resent development of fuzzy multiple criteria decision making. Pai and Rajasekaran [17] discussed anout genetic algorithm for neural network and fuzzu logics. Genetic algorithms have applied in biology, computer science, image processing, pattern recognition, physical science, social science, and neural networks.

The genetic algorithms are good at taking larger, potentially huge, search spaces and navigating them looking for optimal combinations of things and solution which we might not find in a life time. The genetic algorithms are very different from most of the

ISSN: 1738-9968 IJHIT

optimization method. Genetic algorithms need design space to be converted into genetic space. So, Genetic algorithms work with a coding of variables. The advantage of working with a coding of variable space is that coding discretizes the search space even though the function may be continuous. A more striking difference between genetic algorithms and most of the traditional optimization method is that Genetic Algorithm uses a population of points at one time in contrast to the single point approach by traditional optimization method. This means that Genetic Algorithms processes a number of designs at the same time. As we have seen earlier, to improve the search direction in traditional optimization methods, transition rules are used and they are deterministic in nature but Genetic Algorithms uses randomized operators. Random operators improve the search space in an adaptive manner.

While the optimization of simulation model is extensively studied in the literature, it is almost exclusively done from a single response point of view. In reality however, one of ten encounters problems where the assessment of the behavior of a system depends on multiple performance measures. In these cases, the solution that is perceived as the "best' solution will often turn out to be a compromise solution, which may differ significantly from the optimal solutions that would be found when following a single response approach. Different system configurations will typically improve some performance measures while deteriorating others. The selection of the best candidates (system configuration) among a finite set of alternatives assessed for a finite set of criteria (performance measures) is a typical multicriteria decision making problems.

Our contribution: As outlined in the above, we combine a classical MCDM-method with a genetic algorithm, which enables us to use the algorithm for the selection of the best systems configurations in a combinatorial optimization problem, where the set of candidates, although finite, is prohibitively large. Our aim is to find the best feasible configuration (according to the Promethee MCDM-method) for a stochastic discrete event simulator when the number of possible configurations is prohibitively large (*i.e.,* evaluating all the candidates is impossible within the allocated time-budget).

Organization: The remainder of this paper is organized as follows: Section 2 descried optimization problem, Promethee and Promethee-i MCDM-Methods and Genetic Algorithm respectively. Section 3 descried the simulation model and the performance measures Conclusion is given in the final Section 4.

2. Optimization Problem

Every configuration of the simulator can be defined by a vector $\alpha = (x_1, x_2, ..., x_m)$ where x_i represents the setting for the ith input parameters of the simulator. The n corresponding performance measures are defined by the vector $\beta = (y_1, y_2, ..., y_n)$ where $\beta = f(\alpha)$. We consider the form of the function f to be unknown; the values of the various performance measures are estimated using simulation. A configuration is considered feasible if a given set of constraints of the form $g(\alpha, \beta) \geq c$ are simultaneously satisfied. Note that feasibility of a configuration can in general only be verified after the execution of the simulations, as the feasibility constraints are functions of the performance measures.

We define the best configuration α_{OPT} as the feasible configuration that would be selected by the Promethee MCDM-method if we applied this method on the set of all possible feasible configurations.

2.1 The Promethee and Promethee-i MCDM-Methods

The Promethee MCDM-method is based on the assessment of finite number n candidates (configurations) on k criteria (performance measures). For each criterion a pair wise comparison (difference of assessment) of candidates 'a' and 'b' translated on the interval [0, 1] into a preference indicator $P_j(a,b)$. One of six different types of preference indicator functions can be selected. These $P_j(a,b)$ are aggregated over the set of all criteria by:

$$\pi(a,b) = \sum_j \omega_j P_j(a,b)$$

with the ω_j in [0, 1] being the normalized weight of criterion j. Then we calculate for each candidate 'a' the strength $\phi^+(a)$ and the weakness $\phi^-(a)$:

$$\phi^+(a) = \frac{1}{n-1}.\sum_x \pi(a,x)$$

$$\phi^-(a) = \frac{1}{n-1}.\sum_x \pi(x,a)$$

Finally we calculate the net dominance $\phi(a) = \phi^+(a) - \phi^-(a)$.

The Best alternative is the one with the highest net dominance

The Promethee-i methods is an interval based variant of this that can be used when the assessments are not crisp, but are defined by intervals. Now, we have executed m replications of a stochastic discrete event simulation, and then we obtain for each alternative configuration m assessments for each criterion (performance measure). These m assessments can be represented 'a' by an interval $[a^l, a^u]$, which we can take either as the interquartile intervals, or as a confidence interval on the mean. All the arithmetic of Promethee is now extended, keeping intervals all along the calculations, by means of the following:

$$[a^l, a^u] + [b^l, b^u] = [a^l + b^l, a^u + b^u]$$

$$[a^l, a^u] - [b^l, b^u] = [a^l - b^u, a^u - b^l]$$

We obtain consecutively:

$$P_j(a,b) = [P_j^{\,l}(a,b), P_j^{\,u}(a,b)]$$

$$\pi(a,b) = [\pi^l(a,b), \pi^u(a,b)]$$

where $\pi^l(a,b) = \sum_j \omega_j P^l{}_j(a,b)$ and $\pi^u(a,b) = \sum_j \omega_j P^u{}_j(a,b)$

Then, we calculate $\phi^+(a) = [\phi^{+l}(a), \phi^{+u}(a)]$ and $\phi^-(a) = [\phi^{-l}(a), \phi^{-u}(a)]$

where $\phi^{+l}(a) = \frac{1}{n-1}.\sum_x \pi^l(a,x)$ and $\phi^{+u}(a) = \frac{1}{n-1}.\sum_x \pi^u(a,x)$

$$\phi^{-l}(a) = \frac{1}{n-1}.\sum_x \pi^l(x,a) \text{ and } \phi^{-u}(a) = \frac{1}{n-1}.\sum_x \pi^u(x,a)$$

and Finally, $\phi(a) = \phi^+(a) - \phi^-(a) = [\phi^l(a), \phi^u(a)]$

In addition the original Promethee method is applied by taking into account all the worst bounds of the assessment intervals $[a^l, a^u]$ and another time by taking all the best bounds of these assessment intervals $[a^l, a^u]$ for all candidates on all criteria. This yields for each candidate another interval $[\phi^{l'}(a), \phi^{u'}(a)]$. Finally this Promethee-i procedures returns a trapezoidal fuzzy number $[\phi^{l}(a), \phi^{l'}(a), \phi^{u'}(a), \phi^{u}(a)]$ for each candidate a. On these trapezoidal fuzzy numbers, we apply the Yager operator Ψ and the best candidate corresponds to the highest value for this Yager operator Ψ .

2.2 Genetic Algorithm

The Pseudo code of our genetic algorithm is given by:

Program Begin
Generate Random First Generation of Chromosomes;
While Stopping Criterion not reached
  Begin
    Current Chromosome = First Chromosome of Current Generation;
    Repeat
      Initialize Simulation Model with configuration represented by current chromosome;
      Run m Replications of the Simulations;
      Save Performance Measures;
      Current Chromosome = Next chromosome of Current generation;
    Until End of Generation reached Rank configurations of current Generation with Promethee-I;
    Modify Ranking to promote Feasible Configurations;
    Attribute fitness to Chromosomes;
    Generate New Current Generations;
  End
Program End

2.1.1 Chromosome Representation: The chromosomes in the algorithm are strings of bits, whose problem-dependent size varies depending on the number and value range of the input parameters. Every possible configuration $\alpha \in A$ has to be mapped into exactly one chromosome. The first Generation is generated randomly.

2.1.2 Stopping Criterion: The stopping criterion can be defined either as a maximum number of generations or as a maximum number of non-improving iterations.

2.1.3 The Simulation: The replication length and the number of replications used are problem dependent and require some experience. Our Simulation models were implemented in ARENA, and we make use of common random numbers to reduce the variability of the difference between the performance measures for two configurations.

2.1.4 Promethee-i Ranking: All configurations in the current generations are ranked using the Promethee-i method based on the interquartile intervals for the elements of β , the vector of performance measures.

2.1.5 Promotion of Feasible Configurations: This procedure verifies whether the performance measures of the configuration fulfill the set of I constraints $g_i(\alpha, \beta) \geq c_i$. Every constraint has an associated weight. We define the feasibility score a configuration as the sum of the weights of all the satisfied constraints. The final ranking of the configuration is now calculated as follows: all configurations are ranked by decreasing feasibility score. All ties are broken in favor of the configuration with the highest

Promethee-i ranking. If no constraints were imposed, then the Promethee-i ranking remains unmodified.

2.1.6 Attribute Fitness: The chromosomes are assigned a fitness value that is a (scalable) linear function of their final ranking.

2.1.7 Creation of the Next Generations: The algorithm makes the use of elitism, cross over and mutation. Whenever a new child chromosome has been generated, we verify whether it is unique within the current generation. If an identical chromosome already exists, we generate another child and replace the duplicate chromosome.

3. The Simulation Model and the Performance Measures

The stochastic discrete-event simulator used is a model of an incident management process of a call centre. The figure below gives a schematic overview of the process-flow of the model. The incidents are initiated by the customers of the call centre.

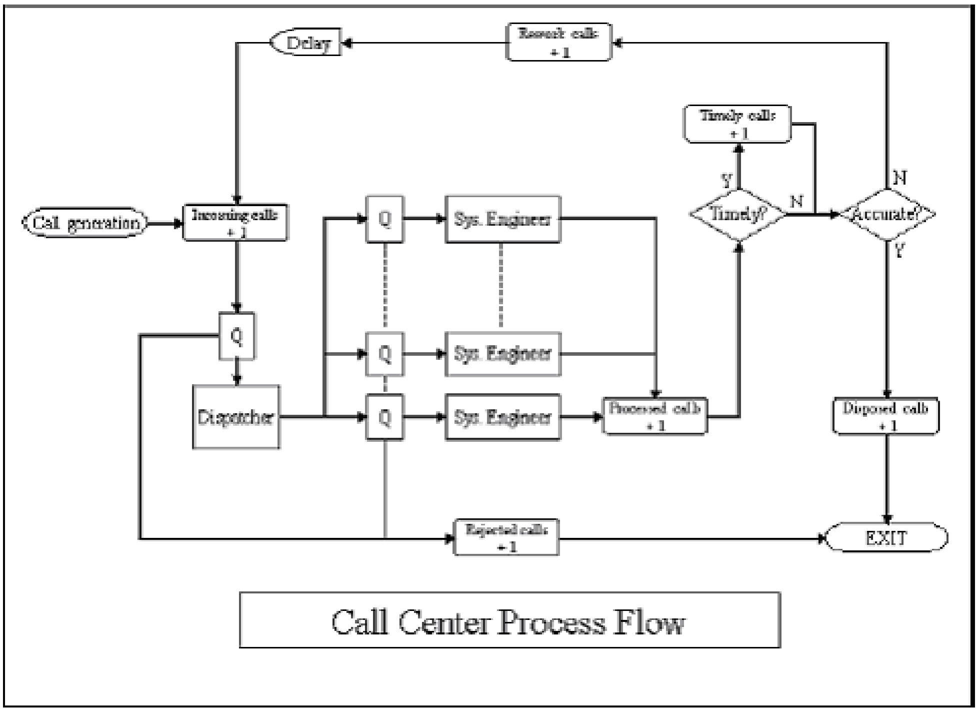

Figure 3.1. Simulation Model

These incidents are represented by the calls that arrive at the centre. These incoming calls follow a stochastic arrival pattern. The calls are subdivided into categories and subcategories, depending on the area of expertise required by the customer. Each category has a specified probability of occurrence, while the subcategories within a certain category are assumed to be equiprobable.

The resources in the model are the dispatcher(s) and the system engineers. Each resource has its own weekly working schedule, an hourly cost (based on the number of skills known) and a FIFO queue associated with it. Incoming calls will wait in the FIFO queue if the resource is busy. A call will be rejected (and leaves the system immediately) if the time spent waiting in a FIFO queue exceeds a certain fixed threshold. Every system engineers has its own areas of expertise, which are specified in the skills matrix. Every line in the matrix represents a subcategory, while every column represents system engineers.

The process flow used in the model is as follows: Every incoming call must pass through a dispatcher. The dispatcher will route the call to a system engineer whose area of expertise covers the category and subcategory of the call. If multiple system engineers are eligible, the dispatcher will route the call to the resource with the shortest queue. Ties are broken in favor of the resource located the most to the left in the skill matrix. The discipline time (the time needed by the dispatcher to decide on the routing of the call) follows a stochastic distribution.

The processing time (the time the system engineer needs to handle a call) follows a stochastic distribution, regardless of the subcategory. For every processed call, there is a fixed probability that the assistance provided. These customers will call back after a stochastic delay. These subsequent rework calls will results in decrease in the performance of the call centre. If the customer is satisfied with the assistance provided, the call is disposed and leaves the system.

We use four performance measures: waiting time in queues, resource utilization or productivity, service level and system cost. Waiting time in queues and resource utilization are average values obtained from standard ARENA statistics. Service level is expressed as the percentage of arriving calls which are finally disposed after a successful handling by the available resources (and as a consequence were not ejected from the system). He cost of a system engineer depends on his degree of polyvalence (number of skills). The overall system cost is a stochastic entity due to the fact that the resources continue to work at the end of their daily schedule until all call waiting in their queue at the end of the working day have been processed.

4. Conclusion

In this study, we described a genetic algorithm that allows us to find the optimal configuration for a stochastic discrete events simulator when multiple performance measures have to be considered simultaneously. This algorithm proved particularly interesting, when the decision making authority is shared by multiple decision makers with conflicting priorities and optimal solutions found middle ground solution that may be acceptable to all the involved parties. The multi-criteria approach relies on an interval based variant of the Prome-thee method, which is combined with a feasibility score to obtain the ranking of the chromosomes within a certain generation of the genetic algorithm.

References

[1] P. A. Raj and D. N. Kumar, “Ranking Multicriterion river basin planning alternatives using fuzzy numbers”, Fuzzy Sets & Systems, vol. 100, **(1998)**, pp. 89-99.
[2] L. Davis, “(ed.) Handbook of Genetic Algorithms”, Van Nostrand Reinhold, New York, **(1991)**.
[3] D. E. Goldberg, “Genetic Algorithm in Search”, Optimization and Machine Learning, Addison-Wesley, Reading, MA, **(1989)**.
[4] J. H. Holland, “Adaptation in Natural and Artificial Systems”, Ann Arbor. University of Michigan Press, **(1975)**.
[5] K. Deb, “An Introduction to Genetic Algorithm”, Sadhana, vol. 24, **(1999)**, pp. 293-315.
[6] K. Deb, D. K. Pratihar and A. Ghosh, “Learning to Avoid Moving Obstacles Optimally for Mobile Robots using Genetic Fuzzy Approach”, Parallel Problem solving for Nature, (Eds.), vol. 5, **(1998)**, pp. 583-592.
[7] C. R. Reeves, “Genetics Algorithms for the Operations researcher”, INFORMS Journal of Computing, vol. 9, no. 3, **(1997)**, pp. 231-250.
[8] E. Triantaphyllou, “Multicriteria Decision Making Methods”, A comparative study, Kluwer Academic Publishers, Dordrecht/Boston/London, **(2000)**.
[9] V. Velton and T. J. Steward, “Multiple criteria Decision Analysis-An Integrated Approach”, Kluwer Academic Publisher, Boston/Dordrecht/London, **(2002)**.

[10] V. R. Lock and H. Pastijin, “Perfromance Measures in Simulating Incident Management Processes of a Call Centre”, Orbel, Belgium, vol. 17, **(2003)** January 23-24.
[11] G. Y. Wang and W. Q. Wang, “Fuzzy Optimum Design of Structures”, Engg. Optimization, Reading, UK, vol. 8, no. 4, **(1985a)**, pp. 291-300.
[12] G. Y. Wang and W. Q. Wang, “Fuzzy Optimum Design of Asesmic Structures”, Earthquake and Structural Design, vol. 13, no. 6, **(1985b)**, pp. 827-837.
[13] W. Wang and K. L. Poh, “Fuzzy MCDM based on confidence analysis”, Fuzzy Economic Review, vol. VIII, no. 2, **(2003)**, pp. 25-28.
[14] C. H. Yeh and H. Deng, “An Algorithm for fuzzy multicriteria decision making”, Proceedings of IEEE International Conference on Intelligent Processing System, **(1997)**, pp. 1564-1568.
[15] Y. Yang and C. K. Soh, “Fuzzy Logic Integrated Genetic Programming for Optimization and Design”, AASCE, Journal of Computing in Civil Engg, vol. 14, no. 4, **(2000)**, pp. 249-254.
[16] C. C. Robert and R. Fuller, “Fuzzy multiple criteria decision making; recent developments”, Fuzzy Sets and Systems, vol. 78, **(1996)**, pp. 139-153.
[17] G. A. V. Pai and S. Rajasekaran, “Neural Networks”, Fuzzy Logic and Genetic Algorithms, Prentice Hall of India Pvt. Ltd. **(2005)**.

Authors

Chandrashekhar Meshram, He received the M.Sc and M.Phil degrees, from Pandit Ravishankar Shukla University, Raipur (C.G.) in 2007 and 2008, respectively and PhD from R.T.M. Nagpur University, Nagpur (M.S.) India. Presently he is teaching as an Assistant Professor in Department of Applied Mathematics, Gyan Ganga Institute of Technology and Sciences, Jabalpur (M.P.), India. His research interested in the field of Cryptography and its Application, Boundary value problem, Statistics, Raga (Music and Statistics), Neural Network , Ad hoc Network, Number theory, Environmental chemistry, Mathematical modeling, Thermo elasticity, Solid Mechanics and Fixed point theorem. He is a member of International Association of Engineers (IAENG), Hong Kong, World Academy of Science, Engineering and Technology (WASET), New Zealand , Computer Science Teachers Association (CSTA), USA, Association for Computing Machinery (ACM), USA, International Association of Computer Science and Information Technology(IACSIT), Singapore, European Association for Theoretical Computer Science (EATCS), Greece, International Association of Railway Operations Research (IAROR), Netherland, International Association for Pattern Recognition (IAPR), New York , International Federation for Information Processing (IFIP) ,Austria, Association for the Advancement of Computing in Education (AACE),USA, International Mathematical Union (IMU) Berlin, Germany, European Alliance for Innovation (EAI), International Linear Algebra Society (ILAS) Haifa, Israel, Science and Engineering Institute (SCIEI), Machine Intelligence Research Labs (MIR Labs) , USA, Society: Intelligent Systems, KES International Association, United Kingdom, Universal Association of Computer and Electronics Engineers (UACEE), The Society of Digital Information and Wireless Communications (SDIWC) and Life –time member of Internet Society (ISOC),USA ,Indian Mathematical Society , Cryptology Research Society of India and Ramanujan Mathematical Society of India (RMS) and editor in chief of IJRRWC, UK and managing editor of IJCMST, India. He is regular reviewer of thirty International Journals and International Conferences.

Shyam Sundar Agrawal, is working as an Associate Professor in the Department of Applied Mathematics in Vikash College of Engineering For Women, Vikash Enclave, Barhaguda Bargarh, India. He received the M.Sc (Maths) and Ph.D Degree from Sambalpur University, Orissa, India in 1997 and 2008, respectively. He is doing his research interest includes Decision Making under Fuzzy Logic, Combinatorics and Cryptography. He is a member of IMS, ISTE India and International Association of Engineers. Computer Science Teachers Association (CSTA, ACM), USA and International Association of Computer Science and Information Technology (IACSIT), Singapore.