Volume: 07 Issue: 03 Pages: 2784-2789 (2015) ISSN: 0975-0290

An Elitist Simulated Annealing Algorithm for Solving Multi Objective Optimization Problems in Internet of Things Design

***Subhamoy Chakraborti**
AVP – IT, Magma Fincorp Limited, India
Email: chakrabortisubhamoy@gmail.com
Sugata Sanyal
Member, School of Computing and Informatics' "Brain Trust"
University of Louisiana at Lafayette, USA
sanyals@gmail.com
*Corresponding author

ABSTRACT

Internet of Things (IoT) is going to introduce billions of data collection and computing nodes all over the world in next few years. IoT would be impacting daily life in many ways by virtue of more granular field-level data collection via those nodes and thus delivering faster actions. One of the key challenges in IoT design decision is resource constraintwhich often limits the space, battery capacity, computing power available in each of the nodes. This presents an optimization problem with multiple objectives, with competing objectives. This paper proposes an algorithm based on Simulated annealing. Simulated Annealing is inspired by the physical annealing process which leads to a gradual movement towards a solution set. This paper proposes to use a variant of this mechanism to solve multi-objective optimization problems in IoT space to come out with a set of solutions which are non-dominated from each other.

Keywords**: Internet of Things, IoT, Simulated Annealing, Multiobjective Optimization Problem**

Date of Submission: Nov 23, 2015 Date of Acceptance: Dec 25, 2015

1. INTRODUCTION

Due to the increasing demand of Internet of Things (IoT) Services and Applications in various areas, IoT is of major interest in research work. The application and service areas include smart cities, healthcare, transport, logistics, retail, safety and security etc.The IoT implementation requires many disciplines of computer science to work together. However designing the sensing and computing nodes are often tricky whilearchitecting an IoT setup. Due to the resource constraints, the size and form factor often put a limitation on the maximum computing capacity available in a node. These factors play roles in competitive ways, making the design decisions a complex one. This paper introduces a Simulated Annealing based elitist Multiobjective Optimization algorithm for coming out with a set of solutions in such scenarios.

2. DESIGN CHALLENGES IN INTERNET OF THINGS

The key technology components in IoT are as below:

a) Hardware platform: This includes the sensors and actuators. The sensors work as input components, sensing and collecting surrounding information while the actuators are mainly output components, altering the surrounding environment by controlling motors and other physical parts.
b) Network protocols:Due to the typical high volume deployment, the usual network protocol needs to be tweaked in IoT deployment. Variants of wireless network protocol are used for handling the nodes. The reliability and performance are the key factors in network design.
c) Application layer: The applications in IoT need to be designed keeping domain specific factors in consideration. Typical applications include smart grid, healthcare, automotive, smart city, industrial automation, environmental monitoring etc. The demand of the application layer also leads to customized operating systems [1].

The overall architecture requirement boils down to a large number of connected devices or nodes, which are autonomous in nature. These devices need to be fabricated at low cost and also should be having fewer issues in maintenance. Also the nodes often need to handle high data rate with low delay tolerance and require a long battery life along with security considerations [2], [3].The security features need to be lightweight also while allowing the protection of the devices via strong encryption and authentication mechanisms [4]as well as handling spectrum management efficiently [18].The nodes are generally small in size, powered by battery having limited source of energy. Since a large number of nodes are required for collecting data, the cost needs to be on the lower side. The low cost factor forces the size of the

Volume: 07 Issue: 03 Pages: 2784-2789 (2015) ISSN: 0975-0290

controlling and processing unit to be small. This in turn constraints the computing capacity. Due to these various factors, designing the nodes presents problems of optimizing multiple factors together. These factors are not linearly related. Rather often they are orthogonal in nature. For example, cost & size reduction puts a restriction on the computing capacity available in a node. Both size minimization and computing capacity maximization can't be done simultaneously without compromising the any of the factors. Similarly computing capacity of the units is limited by the power available due to the remote locations and battery sizes.Social projects like Financial Inclusion also requires the computing nodes to be available at remote rural locations, thus requiring sufficient computing power as well as high battery life [5], which are again competitive in nature. The diverse set of devices [6] often makes the choice even more difficult. Considering these challenges, we propose to handle the IoT node design decisions using Multiobjective Optimization algorithms and propose an algorithm on this line to arrive at the solution set.

3. MULTIOBJECTIVE OPTIMIZATION

As mentioned in the previous section, most of the real world problems deal with simultaneously optimizing two or more objectives, which are competitive in nature. Finding a single solution is difficult in such scenarios. We can get a solution by optimizing one objective, which may not be the best solution for the other objective. Instead of searching for one single solution, Multiobjective Optimization Problem (MOOP) solutions are generally proposed to present a set of solutions, which are optimal in the sense that none of them are better than the other if all the objectives are concerned.The set of solutions of an MOOP consists of all the decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another – these vectors are known as Pareto Optimal. The goal of any algorithm that intends to solve the MOOP should be to achieve the Pareto-optimal set effectively and efficiently.To solve the MOOP, evolutionary algorithms are often used. Simulated Annealing is one of such optimization techniques which are based on the principle of statistical mechanics. Evolutionary Algorithm has been natural choice [7] for solving complex MOOP. Though Simulated Annealing is used for point by point search, this paper proposes to use simulated annealing to solve MOOP problems in IoT space.
Evolutionary Algorithms (EA) are popular in search methods. They mimic the metaphor of natural biological evolutions [8]. EA works on a population and by applying the principle of survival of the fittest, better approximations are sought in each iteration. Simulated Annealing is one of the popular EA algorithms, which originates from the annealing procedure. The strength of SA comes from the gradual temperature reduction technique [9]. In this work, we propose an algorithm which can be used to find the Pareto set using Simulated Annealing.

3.1 ISSUES IN MULTIOBJECTIVE OPTIMIZATION PROBLEMS

An MOOP has more than one objective functions, which are to be optimized simultaneously. Like single objective problem, MOOP has a number of constraints which defines the feasible solution space. We can define MOOP as a vector function f that maps a tuple of m parameters (decision variables) to a tuple of n objectives. Formally:
Min/max $y = f(x) = (f_1(x), f_2(x), \ldots fn(x))$
Where $x = (x_1, x_2, \ldots x_n)$ in X
$y = (y_1, y_2, \ldots y_n)$ in Y
Where x is the decision vector, X is the parameter space, y is the objective vector and Y is the objective space.

3.2 DOMINANCE RELATION

Dominance relation is one of the key concepts in MOOP, where we find out if a solution set is better than the other. Mathematically, a is said to dominate b, if for all vector functions f_i, a has a higher or equal value than that of b and also there exists at least one vector function f_j for which a's value is strictly greater than that of b.

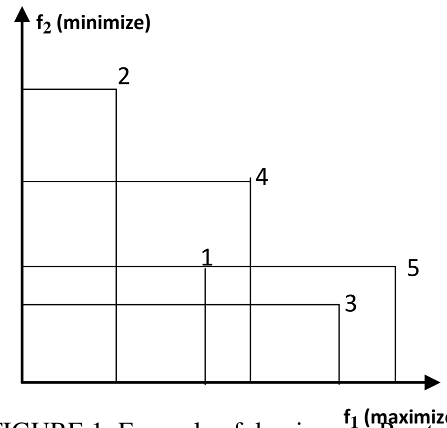

FIGURE 1: Example of dominance, Pareto optimality

We would analyze different pairs of solutions in Figure 1 and find out the dominance relationships.

- Solution 1 & 2: Solution 1 is better than solution 2 in both the objective functions f_1 and f_2. Hence Solution 1 dominates solution 2.
- Solution 1 & 5: Solution 5 is better than solution 1 in terms of f_1, and they have same value in f_2. Hence solution 5 dominates solution 1.

All decision vectors that are not dominated by any other decision vector of a given set are called *non-dominated* with regard to this set. The set of non-dominated solutions with respect to the entire parameter space constitute the *Pareto-optimal front* or the Pareto-optimal set. The goal of a Multiobjective optimization technique is to find the Pareto front efficiently and

Volume: 07 Issue: 03 Pages: 2784-2789 (2015) ISSN: 0975-0290

effectively. In the Figure 1, solution 5 is better than solution 3 in terms of objective function f_1, but solution 3 is better than solution 5 in terms of the objective function f_2. Thus these two solutions are non-dominated with respect to each other and the set of all such points constitute the Pareto optimal set.A solution *astrongly dominates* a solution *b* if solution *a* is strictly better than solution *b* in all objectives. In Figure 1, solution 5 doesn't strongly dominate solution 1 as it is not strictly better than solution 1 in terms of objective function f_1, though it *weakly dominates* solution 1. Among the set of solutions P, the weakly non-dominated set of solutions P' are those that are not strongly dominated by any other member of the set P.The *rank* of solution x_i in a population Q is said to be r_i if the solution is dominated by exactly r_i number of solutions in the population. The non-dominated solutions are of rank zero.

3.3 SIMULATED ANNEALING

Like other Evolutionary Algorithms, Simulated Annealing (SA) operates on a population of potential solutions applying the survival of the fittest mechanism to produce better approximations at each iteration. SA follows the Annealing process where a crystal is cooled down from the liquid to the solid phase. If the cooling is done slowly enough, the energy state of the crystal at the end will be very close to its minimum value. Based on current solution (x_1, x_2, … x_n), the functional value f (x_i) is calculated. A change in solution set is done via changing the temperature. With a small change in the solution set by temperature change, a new functional value f (x_j) is calculated. These functional values are represented in energy form. If the new energy value is less than or equal to the older energy value, the new solution set is accepted. If the new energy value is higher than the older one, it is not discarded straightaway. Rather it is accepted with a probability which is an exponential function of the energy difference, current temperature and Boltzmann's constant.

As defined in [10], this SA principle can be applied in search problems by converting the search space into strings, usually binary. These binary strings represent various states. Low energy state corresponds to near optimal solution. The energy corresponds to objective function. Temperature is the controlling parameter of the system. The primary objective of SA would be to find global minima of a cost function. The strength of SA comes from the fact that to find the global minima, it doesn't always go downhill, by try to go downhill most of the time [11]. A typical SA algorithm would look like below:

1. *Begin*
2. *Q = Initial random string*
3. *T = T_{max}*
4. *E(Q,T) = Calculated energy*
5. *While (T >= T_{min})*
6. *For i = 1 to k*
7. *Mutate (flip) a random position of Q to S*
8. *E (S, T) = New energy*
9. *Set Q ← S with probability 1/ (1 + exp -(E(Q,T) – E(S,T))/T*
10. *End for*
11. *T = rT*
12. *End While*
13. *Decode string Q to get the solution*
14. *End*

The initial solution is taken as a random binary string.At each iteration, one of the bits is flipped in random. The energy state of the new string is identified and accepted with a probability dependent on the energy difference and the temperature. The temperature is reduced using T = rT where 0 < r < 1. The temperature reduction schedule can be experimented to get the most even solution.

The reason why SA is not generally applied in MOOP is because SA usually finds one solution instead of a set of solutions. However the strength of SA comes from its good selection technique and annealing scheme through gradual temperature reduction technique. We would use that in conjunction with the probability of acceptance to define an elitist Multiobjective optimization algorithm in the next sections.

4. ELITIST MULTIOBJECTIVE SIMULATED ANNEALING

In this paper, we propose an Elitist Multiobjective Simulated Annealing algorithm. In this algorithm, whenever we get a solution that is non-dominated, we keep the solution, thus following the principle of elitism. We propose to use an archive to store the non-dominated solutions found so far. We also define two limits – one is the hard limit and the other is the soft limit. The algorithm goes on selecting new points in each iteration. When the count of solution points exceeds the soft limit, clustering is done to reduce the number of points to match the hard limit. The algorithm proposes to use single linkage clustering mechanism, where the distance between any two clusters is given by the value of the length of the shortest link between the two clusters [12]. If there are *t* points, then *t* clusters are assumed to be present at level *0*. The number of clusters keeps reducing at each level and ultimately there would be one cluster at the *(t-1)*th level. If the number of clusters *K* is known, then the process can be stopped when *K* clusters result in.

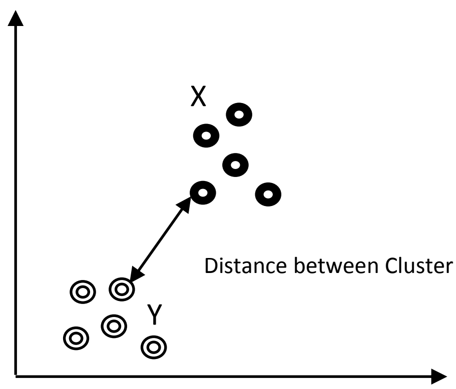

FIGURE 2: Example of Single Linkage Clustering

Volume: 07 Issue: 03 Pages: 2784-2789 (2015) ISSN: 0975-0290

4.1 ALGORITHM OF ELITIST MOSA

Initially a random solution set is generated. The solutions, which are not dominated by any other solutions, are stored in an archive. One of the points from archive is selected randomly and it is perturbed via flipping a random position of the binary string. Then the new string is evaluated and compared with original string. Depending on the dominance relation, the acceptance criterion is decided for the new point.

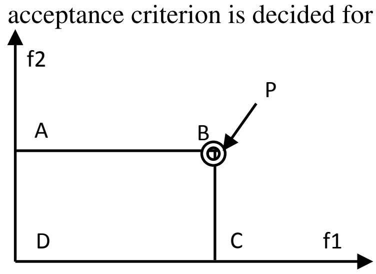

FIGURE 3: Definition of coverage

The coverage is defined by the area covered by the point in the objective space. In Figure 3 above, the area of the rectangle ABCD gives the coverage of the point P for a two objective problem, where both f_1 and f_2 are to be maximized.
All the possible cases arising out of the coverage difference of the old point and the new point are mentioned below:

1) The current point dominates the new point, but no other point dominates the new point. In this case, the new point is selected as the current point with the probability inversely proportional to the difference in coverage.

2) The new point is dominated by not only the current point, but also by k other points in the archive. In this case, we find the difference in coverage of each such point and the sum of it. The probability of selecting the new point as the current point is taken to be inversely proportional to the summation of coverage as calculated.

3) New point is not dominated by current point, but is dominated by k points in the archive where k is greater than or equal to 1. In that case we find the sum of the differences in coverage with respect to all those k points. The probability of selection of the new point as the current point is made to be inversely proportional to this sum.

4) The new point is not dominated by either the current point or by any other point in the archive. In this case the new point is on the same front as the archive. Here the new point is selected as the current point and added to the archive. At this time if the number of points in the archive exceeds the soft limit, clustering is performed to reduce the number of points to the hard limit.

5) New point dominates the current point, but k points in the archive dominate this new point. This would arise if the current point is not a member of the archive. We calculate the difference of coverage between the new point and the k points and select the point from the archive as the current point which corresponds to the minimum difference. This selection is done with the probability of the selection proportional to the delta of coverage.

4.2 METRICS FOR MEASURING THE SUCCESS OF THE ALGORITHM

While solving Multiobjective optimization problems, two primary functionalities need to be achieved by the algorithm [13]. They are as below:

a) The solution set should converge as close to the true Pareto optimal front

b) The solution set should be as diverse as possible

The metrics can be classified into three classes [13]: evaluation of closeness to Pareto optimal front, evaluation of diversity among non-dominated solutions and those which try to achieve both. Maximizing the number of solutions may be one more criterion.
Some of the popular metrics are discussed in this section which can be used in the particular problem set to find out the effectiveness of the algorithm.

a) Error ratio [14] finds the number of solutions that are not present on the Pareto optimal set

b) Set Coverage Metric [15] to calculate the proportion of solutions in an approximation set which are weakly dominated by another approximation set.

c) Generational distance [14] finds an average distance of the solutions from the Pareto optimal front.

d) Spacing [16] to find out the diversity of the solutions. Solutions with uniform spreading is preferred via this metric, however this doesn't take care of the extent of the spread.

e) Spread [17] takes care of Spacing by calculating the sum of distance between the extreme points on the Pareto optimal front. However it doesn't prefer an algorithm having equal distribution and width but more number of points.

Combined metrics to take care of the solution in the particular domain can be thought through to compare the algorithm with other available options.

Volume: 07 Issue: 03 Pages: 2784-2789 (2015) ISSN: 0975-0290

5. CONCLUSION

Multiobjective optimization techniques have been applied in various aspects of engineering and computer science as well as business problems. Internet of Things brings one more area, where solving optimization problem with competitive objectives is a key task in designing the architecture. The algorithm proposed in this paper can be tested and applied in IoT domain to come out with a solution set which optimizes all the competing objectives. The Pareto optimal front thus found can be used to come out with a design decision more conclusively.

REFERENCES

[1].Tuhin Borgohain, Uday Kumar, Sugata Sanyal, "*Survey of Operating Systems for the IoT Environment*", arXiv preprint arXiv:1504.02517, 2015

[2].Uday Kumar, Tuhin Borgohain, Sugata Sanyal, "*Comparative Analysis of Cryptography Library in IoT*", arXiv preprint arXiv: 1504.04306, 2015

[3].Tuhin Borgohain, Uday Kumar, Sugata Sanyal, "*Survey of Security and Privacy Issues of Internet of Things*", arXiv preprint arXiv: 1501.02211, 2015

[4].Bonetto, R., Bui, N. ; Lakkundi, V. ; Olivereau, A. ; Serbanati, A. ; Rossi, M. "*Secure communication for smart IoT objects*", IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012.

[5].Subhamoy Chakraborti, Sugata Sanyal, "*Heuristic Algorithm using Internet of Things and Mobility for solving demographic issues in Financial Inclusion projects*", IJANA, To be published in Mar-Apr, 2015

[6].IDC Smartphone OS Market Share, Q3 2014

[7].C. M. Fonseca and P. J. Fleming, "*An overview of evolutionary algorithms in Multiobjective optimization*", Evolutionary Computation, 3(1), 1995

[8].Tutorial on Evolutionary algorithm, http://geatbx.com/docu/algindex.html

[9].D. Nam and C. H. Park, "*Multiobjective Simulated Annealing: A comparative study to Evolutionary Algorithms*", International Journal on Fuzzy Systems, Vol. 2, no. 2, 2000

[10].S. Bandyopadhyay, S. K. Pal, U. Maulik, M. K. Pakhira, "*Clustering using Simulated Annealing with probabilistic redistribution*", International Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 2, 269-285, 2001.

[11].S Haykin, "*Neural Networks, a comprehensive foundation*", second edition, Addison Wesley Longman, 2001

[12].J. Tau, R. C. Gonzalez, "*Pattern Recognition Principles*", Addison Wesley, Reading, 1974

[13].K Deb, "*Multiobjective Optimization using Evolutionary Algorithms*", John Wiley & Sons, Ltd, 2001

[14].D. A. V Veldhuizen, "*Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations*", PhD Thesis, Dayton, OH: Air Force Institute of Technology, 1999

[15].E. Zitzler, "*Evolutionary Algorithms for Multiobjective Optimizations: Methods and Applications*", PhD thesis, Zurich, Switzerland, Swiss Federal Institute of Technology (ETH), Dissertation ETH No. 13398, 1999

[16].J. R. Schott, "*Fault Tolerant design using Single and Multi-Criteria Genetic Algorithms*", Master's Thesis, Boston, MA; Department of Aeronautics and Astronautics, MIT, 1995

[17].K Deb, S. Agarwal, A. Pratap, T. Meyarivan, "*A fast and elitist multi-objective genetic algorithm: NSGA-II*", Technical report 200001, Indian Institute of Technology, Kanpur, KanGAL, 2000

[18].S Sanyal, R Bhadauria, C Ghosh, "*Secure communication in cognitive radio networks*", Computers and Devices for Communication, 2009 http://www.tifr.res.in/~sanyal/papers/ CODEC-2009. pdf

Biographies and Photographs

Subhamoy Chakraborti is working as Associate Vice President at Magma Fincorp Limited, India where he heads Enterprise Mobility, Data Assurance and Cloud initiatives. He has 12 years of experience in Technology Management. Earlier he has worked with Oracle in ERP product development and various Mobile OEMs like Motorola, Toshiba, Fujitsu, Microsoft and Cisco in developing Mobile phone software at all levels of application stack. He has worked on making many phone programs successful for these OEMs while working for Product Engineering Services group at Wipro Technologies. He has keen interest in IoT, Mobility, Data Science and Cloud. His current focus includes building Enterprise Mobility ecosystem, Data driven decision making and Cloud Infrastructure. He has been working on defining the IT strategy and roadmap for his current organization with the CIO and

Volume: 07 Issue: 03 Pages: 2784-2789 (2015) ISSN: 0975-0290

Business heads. Subhamoy frequently contributes in CIO facing journals and forums.

Sugata Sanyal is presently a Member, "Brain Trust," an advisory group to faculty members at the School of Computing and Informatics, University of Louisiana at Lafayette's Ray P. Authement College of Sciences, USA; a Distinguished Scientific Consultant to the International Research Group: Study of Intelligence of Biological and Artificial Complex System, Bucharest, Romania; an honorary professor in the IIT, Guwahati and a Member, Senate, Indian Institute of Guwahati, India. Prof Sanyal has published many research papers in International and National Journals and Conferences worldwide: topics ranging from Internet of Things and associated Security issues, network security, intrusion detection system, computer architecture etc. He was with the Tata Institute of Fundamental Research from 1973 to 2012 and was acting as a Research Advisor to the Corporate Technology Office, Tata Consultancy Services, India from 2012 to 2015.