Computational Intelligence in Call Admission Control

Jozef Goril', Ľubomír Doboš, Anton Čižmár
Dept. of Electronics and Multimedia Telecommunications
Technical University in Košice
Park Komenského 13, 041 20 Košice
Slovak Republic

joziboy@szm.sk, lubomir.dobos@tuke.sk, anton.cizmar@tuke.sk

Abstract

The paper discuss call admission control problem from computational point of view. Employment of various mathematical approaches to solving the call admission control problem is described, namely the usage of Markov decision process, neural nets, fuzzy logic and genetic algorithms. All these approaches are explained as used with guard channels policy.

1. Introduction

Part of the CAC policy will have to deal with handoff calls. Handoff calls to a cell from neighbouring cells should be considered to have higher priority then new call arrivals and must be dealt with immediatelly. This is because an abrupt, premature termination of an on-going conversation will definitely upset the caller more than a rejection of the call in the first place. Guard channels policy can be used to solve this problem; it reserves a subset of channels allocated to a given cell for handoff calls. Clearly, increasing the number of guard channels will reduce the handoff blocking rate and at the same time, it may increase the new call blocking rate. It is therefore very important to choose the right number of guard channels so that handoff blocking rate is guaranteed to be under the desired treshold and the sacrifice to new call blocking rate is kept at minimum. In this case, for any given desired treshold of handoff blocking probability, the minimum possible number of guard channels being kept at each base station in a wireless network must be determined.

The principle idea is as follows. The total number of available channels (denoted by C) will be divided into two parts: One part (denoted by C_A) is used for handling admitted calls and the other part (denoted by C_H) is reserved for handling handoff calls. In this case, $C=C_A+C_H$ and C_H indicates the number of guard channels. A new call request will be granted for admission if the total number of on-going calls (including handoff calls from other cells) is less than the treshold C_A. A handoff call request will be granted for admission if the total number of on-going calls in the cell is less than the total capacity C.

The number of guard channels has been considered to be one of the key design parameters which have tremendous effects on the performance of wireless networks [1][3][4]. This number at each base station should be determined through optimizing certain performance goal with service quality constraints. When a base station experiences high handoff call blocking rate, it will increase the number of guard channels until the handoff blocking rate drops below its treshold. When a base station does not get to use a significant portion of the guard channels over a period of time, it can gradually decrease the number of guard channels until most of the guard channels are used frequently.

2. A Markov Decision Approach

It is commonly known that users arrivals to the system are random. Therefore, arrival characteristics (average number of incoming users per time unit, average time of user abidance in system, etc.) are interesting. Typical phenomenon in communication network is that if there are none free links, the user is ommited; i.e. creating of users queue is not assumed. For simplicity, just one user-type can be considered, e.g. just one CBR class. Bulk service system labelled as M/M/n with zero queue length is suitable to describe network with such characteristics.

Stochastic system of M/M/n type is characterised as follows: users come into system in sigle, independently of each other and independently of maintenance behaviour. It is assumed, that new call arrivals match a Poisson process with mean λ and channel holding times meet exponentional distribution with mean $1/\mu$. There are n independent and equivalent service units in disposal to incoming users. System does not assume creating of waiting users queue, therefore if system is overloaded, the user is ommited. Such Markov chain can be described by system of probability equations. Poisson process is homogeneous Markov process which can be solved using system of Kolmogor differential equations. Its solution represents steady-state probability vector, which enables to derive probability of new call blocking and/or probability of handoff call drop-out [1][2][3].

The Markov model difficulty exponentialy grows with adding new service classes. The model of complex network is very complicated and cumbersome to compute [6][7][10]. Therefore, not a little labour is dedicated after simplified models and control techniques.

3. A Neural Nets Approach

In the following, consider a number M of different connection types to be served by the network. Further assume, that the entire information about the number of all connections being multiplexed is available. The system state seen by the network will be denoted by $X=\{n_1,n_2,\ldots,n_M\}$, with n_i is the number of active connections of type i being in the system. From mathematical point of view, the main CAC function can be represented by a mapping of the system state X to a decision vector Z defined by $Z=\{z_1,z_2,\ldots,z_M\}$, where $z_i=1$ stands for the acceptance decision of a connection establishment request of type i and $z_i=0$ for the rejection case. The CAC is thus reduced to the implementation of a mapping $f\colon X \rightarrow Z = f(X)$ according to the predefined quality of service of the network.

The mapping f can further be simplified by using the state $X^*=\{n_1,n_2,\ldots,n_i+1,\ldots,n_M\}$, i.e. the system state just after accepting the connection request of type i. The decision vector is reduced to

$$Z^*=\{z_i\}=\begin{cases} 1 & \text{connection } i \text{ should be accepted} \\ 0 & \text{connection } i \text{ should be rejected} \end{cases}$$

and CAC mapping to $f^*\colon\ X^* \rightarrow Z^* = f^*(X^*)$.

As discussed above, the connection admission control function can be interpreted as a mapping of e.g. the state vector X into the acceptance vector Z. This functional mapping divides the M-dimensional state space into two regions: the accept region and the reject region. In other words, the CAC problem can be formulated like a pattern recognition problem: upon recognition of the load pattern X, a yes/no decision has to be made to accept/reject the connection request [8].

Feed-forward neural net structure with back-propagation learning is the most commonly used for CAC purposes. It contains an input layer, one (or more) hidden layer and an output layer, connected in a fully meshed, feed-forward manner. A typical neural net for CAC with 3 service classes is shown at Fig. 1.

There are two operation modes: learning mode and recall mode. In the learning phase, pairs of input/output vectors are presented to the net. It computes its own output vector according to the equations mentioned above and compares the computed output vector with the presented output vector. The comparison results in an error vector, which will be used to change the weight matrices according to a learning rule. The learning phase will end if all input/output pairs to be learned have been presented and the total error is lower then a predefined threshold. After the completion of the learning phase, the information about the input/output pairs, which represents a mapping, can be seen as stored in the weight matrices.

The typical characteristic of neural net is the fact that it always can find the solution and it is always "the nearest" one. But how close is this "the nearest" one strongly depends on quality of learning phase. The main disadvantage of this mechanism is the difficulty to generate a significant number of good- and bad-patterns for the neural net to learn. The CAC performance of neural net is thus strongly dependent on the statistical significance of load patterns during the learning phase.

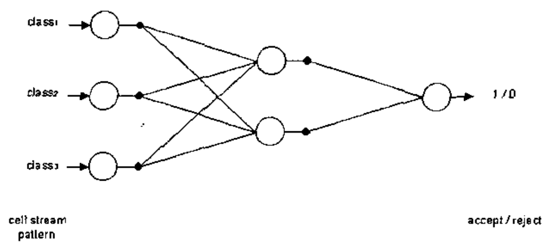

Fig. 1 A typical neural network for CAC with 3 service classes

4. A Fuzzy Approach

Fuzzy control denotes the field in control engineering in which fuzzy set theory and fuzzy inference are used to derive control laws. The concept of a fuzzy set is an extension of the concept of an ordinary set, called a crisp set. As it is known, for a crisp set B, an element b_I either belongs to B (number „1" is used to denote this relationship) or not (number „0"). This condition can also be expressed by the mapping b_I from B to the binary space $\{0,1\}$, which is called the „characteristic function" of B. A fuzzy set $F(B)$, on the other hand, is defined by a „membership function" that can assume any real number in the closed interval $<0,1>$.

Any value between 1 and 0 can express the grade of membership for which an element belongs to this fuzzy set. The concept of fuzzy sets makes it possible to use fuzzy inference. In the method of fuzzy inference, the knowledge of an expert in a field of application is expressed as a set of „IF-THEN" rules, leading to algorithms describing what action should be taken based on currently observed information.

Fuzzy controllers are the application of fuzzy sets and fuzzy inference in control theory. Their operation is very often divided into three phases, called fuzzification, inference, and defuzification. Let X be the finite set of the numerical values for the controller inputs. *Fuzzification* is a procedure that produces a fuzzy subset from the measurement of system input. That is, it is a mapping from the set X to a set of linguistic values. *Inference* is an interface that produces a new fuzzy subset from the result of the fuzzification (the linguistic values) using the set of rules. The result of inference is a fuzzy subset which can be called the fuzzy control action. For example: IF the number of lossed handoff calls is high, THEN the number of guard channels being added is high. *Defuzzification* is a procedure that produces a crisp output from the result of the inference. It means that it says, how many channels will be added exactly (e.g. 4 channels).

It was shown, that a fuzzy logic controller can provide algorithms which convert the linguistic control strategies based on intuition, heuristic learnings and expert knowledge into an automatic control strategy for connection admission control and bandwith management purposes [5][9]. In particular, the methodology of fuzzy logic controller appears to be very useful when the processes are too complex for analysis by conventional quantitative technique, and when the available information cannot be interpreted correctly and with certainty. It offers instruments for building efficient, simple, and robust algorithms, which can be simply self-adaptable to various traffic rate and QoS requirements.

5. A Genetic Approach

Consider system that can be described by a finite-dimensional Markovian state vector with a finite number of states. Define policy as simply mapping of the system state to a decision rule, without considering of system constraints. It means, when the system is full, the policy can admit new call even though the system will block all arrivals.

For a system described by a finite state Markov chain, the call admission problem can be formulated as a Markov decision process (MDP). The MDP model assumes a Markov system and a set of state-dependent controls. A policy is simply a specification of controls. Fixing a policy fixes the state transition matrix as well as a set of rewards that depend on both, system state and state transitions.

Genetic algorithms (GAs) combine the rules of natural selection and genetics to form a search algorithm for optimization problems. GAs are fairly different from the traditional search algorithms, in the sense that their ability to search for the optimum is not confided to any particular type of constraint. They only require the natural parameter set of the problem to be coded as a finite length string over some finite alphabet. The fact that GAs exploit coding similarities in a very general way results in overcoming most of the limitations of the functions to be optimized like continuity, derivative existence, unimodality etc.

With many optimization methods, the decision space on a point by point basis can be evaluated, i.e. the solution to the problem is searched by moving from one point to the other [14]. This method can often result in locating false peaks in a multimodal space. GAs on the other hand, work with a number of points (a *community*) simultaneously which reduces the chance of finding local optima.

The initial community is a collection of coded individuals called *organisms*. Each organism contains group of characters called *genes* and corresponds to a solution candidate for the optimization problem; i.e., for call admission, the genetic code specifies the structure of the policy. The objective function of the optimization problem is called the fitness measure.

There are four basic steps in a simple GA: selection, crossover, reaping and mutation.

A typical two parents - two children GA, shown in Fig. 2 shows these four basic steps. First a number of parents are chosen with the *selection* process to enter the mating pool. Then the *crossover* takes place on all the parents that are randomly paired. The offsprings of the crossover are the new entries to the population and are allowed to *mutate* before entering the community. The algorithm is allowed to run for a number of generations repeating these basic steps.

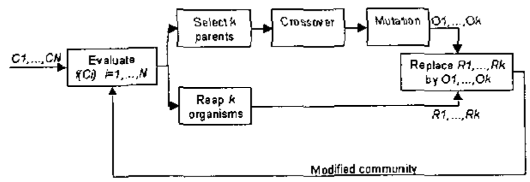

Fig. 2 A principle of genetic algorithm

In the call admission problem for cellular networks, the GA community is formulated as a collection of call admission policies, each policy being an organism. The policies are represented as bit strings in two ways:

Each policy decision is represented with a bit, 1 for accepting and 0 for denying the service. The long bit string with $OxNxS$ bits (O is the number of possible call origins (e.g *7* for hexagonal architecture), N is the number of cells in the network and S is the number of feasible states in the system) constitutes the single gene in the organism. We call this the complete representation of the policy. Complete representation is easy to implement for small systems, where the number of states is not very large. For bigger systems, the block coded policy representation is used.

The analytical complexity of a GA is primarily the complexity of the policy evaluation procedure. Blocking probabilities should be computed for the performance measure, but unlike MDP, it is possible to use approximation methods or Monte Carlo simulations instead of solving for the state equations of the system.

Another advantage of GAs is that they can be used to find solutions for any optimization problem, for which a set of candidates for the solution can be constructed and coded to binary strings. So, there is no need to confine to Markov Processes or to a linear measure of performance. The only information needed for the algorithm is the value of the performance of the candidate for the solution.

The major difference between MDP and GAs lies in the fact, that MDP always finds the optimal solution for the problem if it can handle the problem, whereas GAs usually can handle the problem, but does not necessarily find the optimum. GAs search for the optimum from a set of candidate solutions, and try to improve on the best solution found in each of iteration. The solution, after a finite number of steps, may not converge to the optimal found by MDP. These encouraging results suggest to use of GAs for more complex problems that are not solvable by MDP.

Genetic algorithms are often used together with local call admission policies [11][12][13][14].

6. Conclusions

As the mobile users' requirements grow, the requirements for computational intelligence of mobile network control mechanisms grow, too. Several commonly used mathematical approaches to CAC problem solving was mentioned in previous chapters. Their bases, advantages and disadvantages was discussed.

There are three complementary classes of computational intelligence algorithms: neural networks, fuzzy systems and genetic algorithms [15]. The relationships between these algorithms are summarized in Fig. 3. The figure illustrates that neural networks and fuzzy systems may closely interact. Neural network data processing and learning capabilities can be used to facilitate learning in fuzzy systems. Conversely, fuzzy systems may be used to embed structured knowledge within neural networks at initialization. Though genetic algorithms are somewhat separated from the other two, they can effectively perform the role of a trainer for both neural networks and fuzzy systems.

The experience have shown, that each of them can perfectly solve some particular situations, hence, under another conditions it may be very cumbersome and useless. The latest works tend to usage of various combinations of these methodologies. Usually, some level of self-learning or self-tuning is employed.

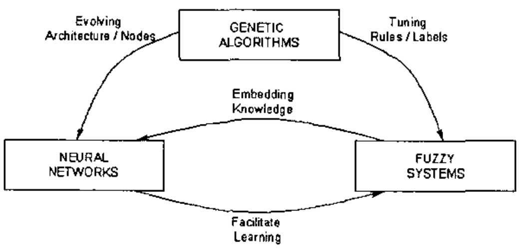

Fig. 3 Computational intelligence techniques relationship

Probably, it will take even enough exertion to find the general one, which could solve the CAC problem for different customer classes in combination with various service classes, within the full satisfaction.

References

[1] L. Dobos, J. Goril, "Call Admission Control in Mobile Wireless", *Radioengineering*, Vol.11, No. 4, pp. 17-23, December 2002

[2] S. Pavol, "CAC Algorithms for Mobile ATM Networks. Diploma Work", Technical University in Kosice, Faculty of Electrical Engineering and Informatics, Dept. of Electronics and Multimedia Telecommunications, Kosice (Slovakkia), 2001

[3] I. Bardos, "CAC Agorithms for 3-rd Generation Mobile Communication Systems. Diploma Work.", Technical University in Kosice, Faculty of Electrical Engineering and Informatics, Dept. of Electronics and Multimedia Telecommunications, Kosice (Slovakia), 2002

[4] Y. Zhang, D. Liu, "An Adaptive Algorithm for Call Admission Control in Wireless Networks", Online at http://cil.ece.uic.edu/~dliu/PS/zha-liu-glo.pdf

[5] Q. Hu, D. W. Petr, C. Braun, "Self-tuning Fuzzy Traffic Rate Control for ATM Networks", *Proceedings of the 1996 IEEE International Conference on Communications ICC'96*, pp. 424–428, Dallas (USA), 1996

[6] P. Marbach, J. N. Tsitsiklis, "A Neuro-Dynamic Programing Approach to Call Admission Control in Integrated Service Networks: The Single Link Case.", *Technical Report LIDS-P-2402*, Laboratory for Information and Decision Systems. M.I.T., November 1997

[7] P. Marbach, O. Mihatsch, J. N. Tsitsiklis, "Call Admission Control and Routing in Integrated Services Networks Using Neuro-Dynamic Programing.", *IEEE Journal on Selected Areas in Communication*, Vol. 18, No. 2, pp. 197-208, February 2000

[8] P. Tran-Gia, O. Gropp, "Structure and Performance of Neural Nets in Broadband System Admission Control", *Neural Networks in Telecommunications*, Chapter 7, pp. 127-141, Kluwer Academic Publishers, January 1994

[9] S. Chandramathi, S. Shanmugavel, "Adaptive Allocation of Resources with multiple QoS Hetrogeneous Sources in ATM Networks – A Fuzzy Approach", *Proceedings of 2002 IEEE International Conference on Communications*, New York (USA), 2002

[10] E. Altman, T. Jimenez, G. Koole, "On Optimal Call Admission Control in a Resource-Sharing System", *IEEE Transactions on Communications (2001)*, vol. 49, pp. 1659-1668, 2001

[11] A. Yener, C. Rose, "Local Call Admission Policies for Celullar networks Using Genetic Algorithms", WINLAB, Dept. Of Electrical and Computer Engineering, March 1995

[12] A. Yener, C. Rose, "Genetic Algorithms and Call Admission to Telecommunications Networks", *Computer and Operations Research* 23(5), pp. 485-499, 1996

[13] S. Piyaratna, N. Shenoy, S. Halgamuge, "A Genetic Algorithm Tuned Fuzzy Logic Based Call Admission Controller for ATM Networks", Online at: http://www.itr.unisa.edu.au/opnet/paper4.pdf

[14] A. Yener, C. Rose, "Near-Optimal Call Admission Policies for Cellular Networks Using Genetic Algorithms", *Wireless 94, The sixth International Conference on Wireless Communications*, July 1994

[15] P. Czezowski, "Computational Intelligence in ATM Call Admission Control", *IEEE 1998 Ph.D. Students Conference GRADCON'98 Proceedings*, pp. 47-50, Winnipeg (Canada), May 1998