Indian Journal of Science and Technology, *Vol 8(12), DOI: 10.17485/ijst/2015/v8i12/58562, June 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

OVSF Dynamic Code Allocation Quality based Glowworm Swarm Optimization Approach in WCDM

P. Kavipriya*

Electronics and Communication Engineering, Sathyabama University, Chennai - 600 119, India;
kaveepriya@gmail.com

Abstract

OVSF codes are used as channelization codes to support applications with different bandwidth requirements for the downlink of 3G mobile communication systems. The code assignment and reassignment problem focuses on the subject of minimizing the number of code relocations. OVSF (Orthogonal Variable Spreading Factor) code assignment method considering traffic characteristics in the WCDMA systems. The previously proposed Dynamic Code Assignment (DCA) scheme allows code reassignments to improve code utilization, but induces some service delay time to on-going calls, unlikely assumptions, including fixed service data rates and code-limited system capacities. Thus, OVSF code tree has insufficient number of available codes. **Method/Statistical Analysis:** In order to solve these problems and efficiently utilized OVSF codes, in this paper presents novel Glow warm Swarm Intelligence (GSO) based approach for active OVSF code assignment in WCDMA based networks. GSO algorithm is completely dependent on the behavioral pattern of glowworm (fireflies). The optimization algorithm requires an understanding of this behavioral pattern. The intensity of the emission of Lucifer in can be changed by glow worms which glow at different intensities. In order to improve the ability of the GSO it employs various phases until code allocation is completed. **Results:** By permitting only codes in the data branch to be reassigned the quality of service guarantee for real time traffic is possible. The simulation results shows that proposed GSO schemes not only guarantees the QoS but also gives high code utilization which is comparable with other existing techniques. **Conclusion/ Application:** Performances of these methods are evaluated in terms of blocking probability, spectral efficiency, delay and throughput. In addition to this different GSO operator are tested under varying traffic loads to increase the overall system performance.

Keywords: ABC, Dynamic Code Assignment, GSO, MAGA Performance Parameters, PSO, Resource Allocation, WCDMA

1. Introduction

In 2nd generation CDMA system such as IS-95, an Orthogonal Constant Spreading Factor (OCSF) code is assigned to each user for supporting voice or data rate services. In this system, to provide higher data rate services, multiple OCSF codes are assigned to a call. However, in the Wideband CDMA (W-CDMA) system which is proposed by 3rd Generation Partnership Project (3GPP), it is probably to provide multi-rate services for a user by assigning only one Orthogonal Variable Spreading Factor (OVSF) code which can be generated by an OVSF code tree[1,2]. In 3GPP specifications[3], orthogonal codes (known as channelization codes) are used to preserve the orthogonality between users' physical channels. In total, 256 pieces of channelization codes are available and the sprea Figure 5. Find Cut Dimension – Pseudo Code. ding factor indicates how many bits of those codes are used in the connection. WCDAMA uses OVSF codes as channelization codes for data spreading in both reverse link and forward link. So, hardware complexity

---

*Author for correspondence

of a mobile station is reduced in comparison with the case that the system employs OCSF codes because a mobile station requires only one transceiver per mobile station. Nevertheless, the system using OVSF codes may have lower spectral efficiency according to the adopted code assignment and reassignment scheme than that using OCSF codes. It is because a user requesting kR bps may not be served although the system has enough free capacity ($\geq \mathrm{kR}$) if there is not an available code of rate kR , where is the lowest (leaf) code rate and is (is an integer between 0 and $\log_2 \mathrm{SF}_{\max}$ in case that $\mathrm{SF}_{\max}$ is the maximum spreading factor). To overcome this problem, an efficient code assignment scheme is required. They are basically Walsh codes of different lengths that are able to preserve orthogonality between channels even when they are operating at different data rates. Much work in the literature[4–6] has intensively investigated code placement and replacement schemes with the objective of producing lower code blocking probability and less reassignment cost in the OVSF code tree.

Existing code assignment algorithms toward this direction[7,8] can be classified into two categories: single-code assignment and multi-code assignment. Single-code assignment assumes that each UE has only one Rake combiner so the system can only allocate one code to it. For multi-code assignment, a UE has more than one Rake combiner so its request can be honored by allocating multiple codes. Due to the dynamic nature of code usage, code assignment algorithm may suffer from code fragmentation problem, which refers to the circumstance that the system has enough capacity but cannot grant a request simply because the capacity does not belong to a single code. Code fragmentation may occur to both single-code and multi-code assignments, but it is believed that the problem is less serious in the multi-code case since data rates can be aggregated there. Code fragments can be compacted by using a code replacement algorithm that tries to exchange some allocated codes with unallocated codes of the equal spreading factor in order to obtain codes of lower spreading factors (i.e., higher data rates). Solving code placement problem involves two steps. First, determine the code rate to be allocated. Second, locate a code in the code tree that corresponds to the code rate. If UE is equipped with only one Rake combiner, the first step is straightforward: find a code rate $2^i R$ that minimizes rate consumption. As to locate a code of rate $2^i R$ in the code tree, there have been several approaches proposed.

Moreover, frequent code reassignments induce signaling cost which increases service delay time additionally in the radio access network. So, frequent code reassignments influence Quality of Service (QoS) of the on-going calls. In this paper, propose a novel OVSF dynamic code allocation quality based glowworm swarm optimization codes allocation schema and solves code blocking problem. This work focuses on providing significant results for the OVSF code assignment using a heuristic algorithm. This study presents an efficient glowworm swarm optimization algorithm for the purpose of OVSF code assignment, which could adjust the parameters adaptively based on the value of individual objective function and additionally considers the quality based requirement. GSO is applied to optimize the OVSF code tree for dynamic reassignment of the codes in a manner just described above. Premature convergence may occur when the GSO has been optimizing the OVSF code tree for a given traffic density. Specifically, if the GSO works properly, the entire populations will convergence to a single chromosome representing optimal OVSF code tree for current traffic density.

2. Basic Background Knowledge

Park et al.[9] investigates computationally efficient suboptimal Dynamic Code Assignment (DCA) schemes with Call Admission Control (CAC) for orthogonal variable spreading factor code-division multiple-access systems. The author examines two different approaches. The first approach reduces the complexity of the DCA scheme by partitioning the total resource into several mutually exclusive subsets and assigns each subset of resource to a group of users in proportion to the corresponding traffic load.

Balyan & Saini[10] propose an OVSF code assignment scheme which reduces the complexity involved as compared to crowded first assignment scheme. This is because the proposed scheme needs to check the immediate parent of the vacant code for code assignment. The crowded first scheme on the other hand checks all the ancestors of the vacant codes up to the root of the code tree. The code blocking performance can be inferior to the CFA design but the number of code searches is reduced considerably.

Tsai & Lin[11] investigate this issue from a different perspective, by applying variable service data rates and an interference-limited system capacity. They propose a three OVSF code assignment and reassignment strategies for this scenario, including sparse-first, sparse-first/right-most, and modified sparse-Hadamard Consequently, OVSF code tree has inadequate number of available codes.

Hadamard-Walsh matrix is utilized to construct a successive orthogonal variable codes which can be denoted by the tree as shown in Figure 1. A K layer code tree (Yang and Yum, 2001) is shown in the Figure 2. The OVSF code tree is called as a binary tree with K layer, where the each and every node represents a channelization code (k, m), where $k = 0,1,\dots,K,\quad m = 1,\dots.,2^K$. The lowest layer is the leaf layer and the highest layer is called as root layer. The data rate that a code can be support is called its capacity. Consider the capacity of the leaf codes (in layer K) be denoted as R. Then the capacities of the codes in the respective layer $(k-1),(k-2),\dots.,1,0$ are $2R,4R,\dots..,2^{K-1}R,2^K R$ respectively, as shown in Figure 1.

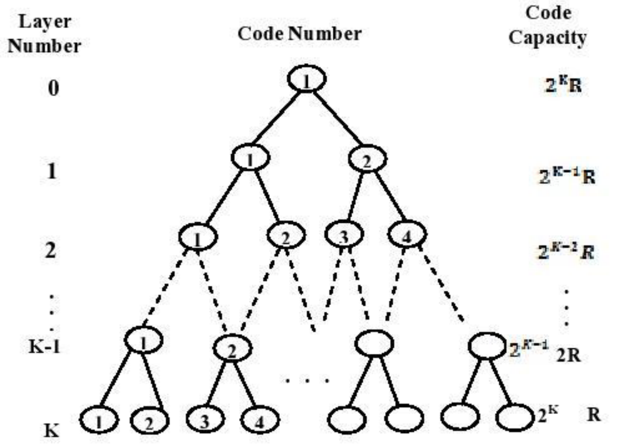

Figure 1. OVSF code tree.

Layer k has 2^k codes and they are consecutively labeled from left to right path, starting from one. The m^{th} code in layer k is denoted to code (k,m). In each layer the total capacity of all the codes is $2^k R$, it is irrelevant of the layer number. Also define the maximum spreading factor $N_{max} = 2^k$ as the total number of codes in layer K.

The tree has a k^{th}-layered structure. The root of the tree is the topmost layer (0^{th} layer) with the least Spreading Factor (SF) while the lowest layer (k^{th} layer) with highest SF is considered as leaf of the tree. The number of codes in the leaf which is equivalent to its SF denotes the capacity of the whole tree. Each code in the tree can be denoted as $C_{(SF,N)}\,1 \le n \le SF$ where SF is the spreading factor of the code and n indicates the position of the code in the layer. The spreading factor is the number of chips utilized to denote each data bit. The codes in the same layer are mutually orthogonal and in the same way, the codes in different layer are also orthogonal, if mother child relationship is not observed. The maximum capacity of the system is expressed as $capacity = 2^K R$ where K denotes the highest layer of the tree and R represents the fundamental data rate.

3. Code Blocking Scenario

Code blocking[16] is the major limitation of OVSF-CDMA system. In Figure 2, code tree with four layers is taken into consideration. The maximum capacity of the code tree is 8R. In the code tree, two codes with SF4 (for data rate 2R) and 8 (for data rate R) are occupied. Hence, the capacity used for the OVSF code is 3R. The remaining capacity of the code tree is 8R-3R=5R. If a new call with data rate 4R arrives, code from the third layer is needed. The code tree is not capable to offer code for the new call, as both the codes equivalent to 4R capacity is blocked. Thus, this is a scenario in which a new call cannot be supported even if the system has adequate capacity to deal with. This scenario called code blocking has to be avoided through efficient and optimized assignment and reassignment schemes (Davinder S Saini et al., 2008).

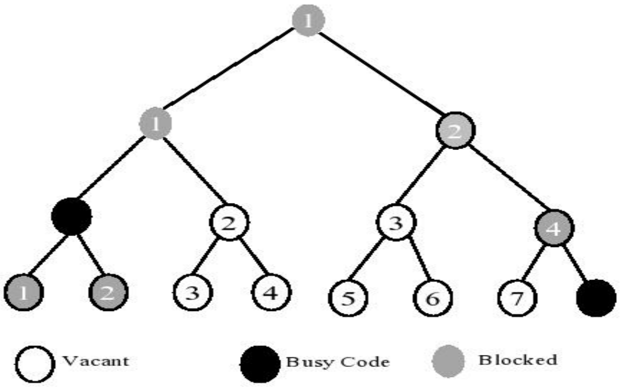

Figure 2. Code Blocking Scenario.

In order to find a suitable OVSF code to the call requested user due to the code blocking problem, reallocation of assigned OVSF codes in the code tree is help to find this possible code. The following section 4 discuss about the reallocation process starting from GSO, and then it is evaluated to existing methods to make fit the structure of OVSF code assignment strategy. Existing works concentrate on the limitation of code capacity and assumed that the well thought-out CDMA system is preferably a code-limited system. Though, in actual radio environments, CDMA systems are usually not code-limited.

4. Proposed Dynamic OVSF Code Allocation with Quality based GSO Algorithm

In this paper presents a Glowworm Swarm Optimization technique is presented in this approach to have better code blocking probability. The flowchart of the proposed approach is given in Figure 4. If a call cannot be allotted a code due to unavailability of the code with the requested rate then GSO block is executed. In the GSO block, reassignment process of OVSF codes is carried out. A sample OVSF code tree is shown in Figure 2. The OVSF code tree which is input to the GSO block is called as initial glowworm (G_{ini}) and this glowworm is denoted with the index number which belongs to active users in the given code tree (G_{ini} = [6 9 14 16 21]). Initial population glowworm with various code tree index numbers other than the number of data bit rates of each glowworm is the same as initial glowworm. The n glowworm consists of existing coded information of OVSF tree is attained by means of permutation and gives an optimized result for a problem.

The OVSF code tree which is input to the GSO block is called as initial glowworm (G_{ini}) and this chromosome is denoted with the index number which belongs to active users in the given code tree (G_{ini} = [6 9 14 16 21]).

Each glowworm encodes the object function value $J(x_i(t))$ at its current location $x_i(t)$ into a Luciferin value and transmits the equivalent data within its neighborhood. The set of neighbors $N_i(t)$ of glowworm comprises of those glowworms that have a fairly better Luciferin value and that are positioned within a dynamic decision domain, and updating through Equation (1) at each iteration.

Local-decision range update:

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\right\}\right\}; \tag{1}$$

And $r_d^i\ (t+1)$ denotes the glowworm i's local-decision range at the $t+1$ iteration, r_s denotes the sensor range, n_t represents the neighborhood threshold, the parameter β affects the rate of change of the nerghbourhood range. The number of glow in local-decision range:

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\right\}\right\}; \tag{2}$$

And $x_j(t)$ denotes the glowworm $i's$ position at the t iteration, $l_j(t)$ represents the glowworm $i's$ Luciferin at the t iteration and the set of neighbors of glowworm i comprises of those glowworms that have a relatively higher Luciferin value and that are positioned within a dynamic decision domain whose range r_d^i is bounded above by a circular sensor range r_s $(0 < r_d^i < i)$. Each glowworm i chooses a neighbour j with a probability $p_{ij}(t)$ and moves toward it. These movements that depend only on local data facilitate the glowworms to separate into disjoint subgroups and demonstrate a simultaneous behavior toward and eventually co-locate at the multiple optima of the given objective function probability distribution used to select a neighbor:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k\in N_i(t)} l_k(t) - l_i(t)}; \tag{3}$$

Movement update

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\left\|x_j(t) - x_i(t)\right\|}\right); \tag{4}$$

Luciferin-update:

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(x_i(t)); \tag{5}$$

And $l_i(t)$ depends Luciferin value of glowworm $\rho \in (0,1)$ at the iteration, results in the reflection of the cumulative goodness of the path followed by the glowworms in their current Luciferin values, the parameter γ only scales the function fitness values, $J(x_i(t))$ is the value of test function.

Each glowworm i selects a neighbor j with a probability $P_{ij}(t)$ and moves toward it. These movements that are based only on local information, enable the glowworms to partition into disjoint subgroups, exhibit a simultaneous taxis-behaviour toward and eventually co-locate at the multiple optima of the given objective function.

If an OVSF code tree denoted by best glowworm, can allocate the requested data bit rate to appropriate user, then optimization criterion is confirmed and requested data bit rate is allocated to desired user. If not, other glowworms in the population are checked. The stopping criterion for this process is either run-on until to assign the requested data bit rate to a user or until the end of predetermined loop counter.

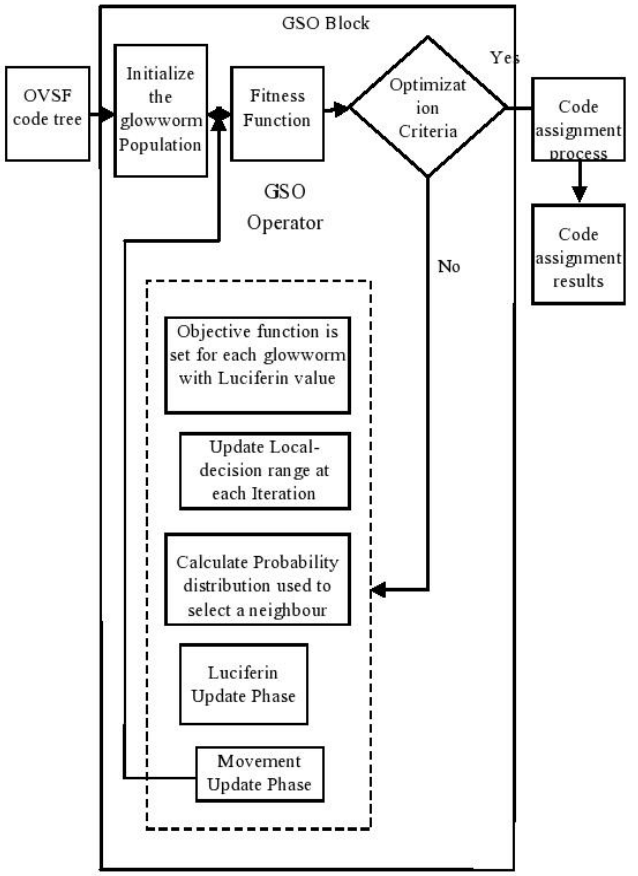

Figure 3. Flowchart of OVSF code assignment system using GSO block.

4.1 Algorithm for GSO based OVSF Code Assignment Approach

Step 1: Introduction of a call is done when there is a signal from the call processor to the resource manager for allocation of assets to a traffic channel.

Step 2: The requested data bitrate is checked in OVSF code tree to assign channels to user or not.

Step 3: If the requested user data bitrate capacity is presented then assign data bitrate to user in the OVSF code, or else call GSO block.

Step 4: In the GSO block, the reassignment process of OVSF codes is performed for location of an appropriate code for the call request.

Step 5: The initial glowworm (G_{ini}) for the OVSF code tree which is defined as follows,

$$n = SF - \sum_{i-1}^{V} H(i) \tag{6}$$

Step 6: Each initialized glowworm calculates the fitness function f(j) into a Luciferin value and transmits the equivalent codes data within its neighborhood.

$$f(j) = \frac{1}{\sum_{i=1}^{V}(G_{ini}(j) - P(j,i)) \times H(i))} \tag{7}$$

Step 7: The set of neighbors $N_i(t)$ of glowworm of the OVSF code comprises of those glowworms that have a fairly good luciferin value and that are positioned within a dynamic decision domain, and updating through Equation (2) at each iteration.

Step 8: The number of glow in local-decision range for OVSF code tree assignment is specified in equation (3).

Step 9: Each glowworm (codes in the OVSF code tree) i chooses a neighbour j with a probability $p_{ij}(t)$ and moves toward it. These movements that depend only on local data facilitate the glowworms (codes) to separate into disjoint subgroups and demonstrate a simultaneous behavior toward and eventually co-locate at the multiple optima of the given objective function $f(x_i(t))$ with probability distribution used to select a neighbor codes in the OVSF code tree in equation (3)

Step 10: Movement update in equation (4) and Luciferin-update in equation (5)

Step 11: Increasing goodness of the path followed by the glowworms in their current Luciferin values, the parameter only scales the function fitness values, $f(x_i(t))$ is the value of test function.

Step 12: Each glowworm $P_{ij}(t)$ selects a neighbor j with a probability $P_{ij}(t)$ and moves toward it. These movements that are based only on local OVSF code tree information enable the glowworms to partition into disjoint subgroups

Step 13: Eventually, each glowworm in the population is ensured for its fitness values. If an OVSF code tree is meant by best glowworm, can assign the requested data bit rate to fitting user, then the optimization criterion is confirmed and requested data bit rate is allocated to desired user. If not, other glows in the population are checked. Bring to an end to this method is either run-on until to assign the requested data bit rate to a user or until the end of encoded loop counter.

5. Experimentation Results

The main focus of this paper is to enhance the number of free codes at OVSF code tree through reassignment of presently allotted codes. Evaluation can be done by comparing the proposed code assignment approach using Glowworm Swarm Optimization (GSO), it is compared with SGA, D&D-GA and Modified Adaptive GA, PSO, ABC by simulations.

5.1 System Performance

5.1.1 Blocking Probability

The code blocking is the case where a new call requesting a transmission rate cannot be supported because all available codes for this rate are not orthogonal to other assigned codes, even if the code tree has sufficient remaining capacity. Blocking probability is the ratio of the number of blocked calls (NB) to total number of all incoming calls (NT), given by

$$\Pr(\text{blocking}) = \frac{N_B}{N_T} \tag{8}$$

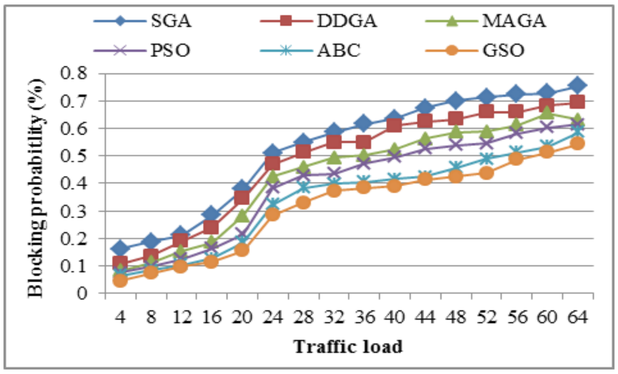

Figure 4. Blocking probability comparison.

Figure 4 shows the results of simulations for blocking probability at different traffic loads when Spreading factor is 256. It is shown that the proposed GSO performs better than other techniques like SGA, DDGA, MAGA, PSO and ABC. For instance, GSO serves more call when it is compared with MAGA and ABC algorithm when traffic load is larger than 10. At higher loads, performance improvement with the GSO becomes more significant, the results are tabulated in Table 1.

5.1.2 Spectral Efficiency

In the following, spectral efficiency is evaluated to measure the ratio of assigned data rate (R_{assigned}) over the total requested data rate ($R_{\text{requested}}$) of all incoming calls, which is given by

$$\eta(\%) = \frac{R_{\text{assigned}}}{R_{\text{requested}}} * 100 \tag{9}$$\

Code blocking probability focuses the number of users while spectral efficiency focuses this user' data bit rates.

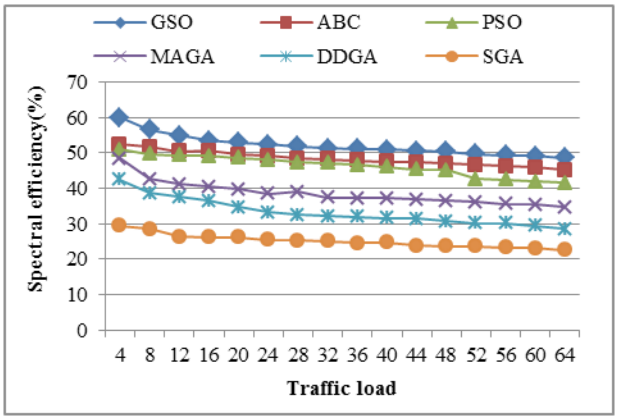

Figure 5. Spectral efficiency comparison.

Figure 5 shows the spectral efficiency of the four methods at different traffic loads. The spectral efficiency of the resource is inversely proportional to the traffic load in the system. Clearly, GSO provides the largest spectral efficiency among SGA, DDGA, MAGA, PSO and ABC at traffic loads is shown in Figure 6, the results are tabulated in Table 2.

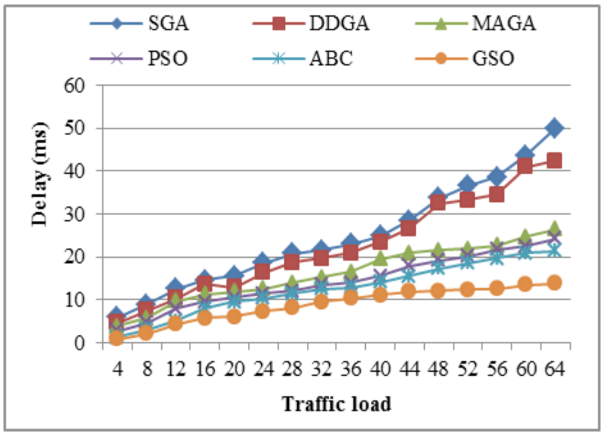

Figure 6. Delay comparison.

Figure 6 shows the delay comparison of the approaches taken for consideration. It is observed from the graph that the proposed GSO approach has lesser delay when compared among SGA, DDGA, MAGA, PSO and ABC approaches. The results delay comparison results of the each method values are tabulated in Table 3.

Table 1. Blocking probability comparison vs. methods

Traffic Load	SGA	DDGA	MAGA	PSO	ABC	GSO
4	0.162	0.109	0.082	0.075	0.065	0.0456
8	0.187	0.136	0.112	0.098	0.084	0.074
12	0.212	0.189	0.154	0.126	0.104	0.098
16	0.285	0.24	0.185	0.163	0.128	0.114
20	0.38	0.345	0.283	0.216	0.186	0.156
24	0.51	0.47	0.425	0.384	0.324	0.284
28	0.549	0.512	0.458	0.429	0.384	0.329
32	0.587	0.55	0.494	0.436	0.398	0.373
36	0.616	0.55	0.501	0.472	0.405	0.383
40	0.636	0.61	0.524	0.495	0.416	0.39
44	0.674	0.625	0.563	0.526	0.425	0.413
48	0.7	0.634	0.587	0.538	0.456	0.424
52	0.7121	0.659	0.589	0.546	0.489	0.437
56	0.724	0.66	0.612	0.582	0.512	0.487
60	0.728	0.682	0.655	0.604	0.534	0.512
64	0.754	0.694	0.632	0.613	0.586	0.543

Table 2. Spectral efficiency comparison vs. methods

Traffic Load	GSO	ABC	PSO	MAGA	DDGA	SGA
4	59.89	52.4	50.9	48.45	42.5	29.5
8	56.56	51.7	49.7	42.68	38.63	28.5
12	54.89	50.5	49.4	41.26	37.56	26.43
16	53.45	50.5	49.1	40.45	36.56	26.2
20	52.89	49.5	48.6	39.68	34.58	26.1
24	52.4	49.1	48.1	38.46	33.25	25.4
28	51.89	48.5	47.4	38.98	32.48	25.34
32	51.36	48.2	47.1	37.36	32.12	25.12
36	51.08	47.9	46.6	37.21	31.89	24.56
40	50.89	47.6	46.1	37.18	31.56	24.7
44	50.65	47.2	45.4	36.89	31.42	23.89
48	50.36	46.9	45.2	36.56	30.65	23.65
52	49.68	46.6	42.6	36.12	30.23	23.56
56	49.36	46.2	42.4	35.62	30.12	23.34
60	49.12	45.8	41.9	35.45	29.45	23.12
64	48.65	45.1	41.5	34.69	28.56	22.65

Table 3. Delay comparison vs. methods

Traffic Load	SGA	DDGA	MAGA	PSO	ABC	GSO
4	5.86	4.56	3.89	2.68	1.23	0.86
8	8.95	7.46	5.65	4.65	3.12	2.12
12	12.52	10.23	9.67	7.86	5.13	4.38
16	14.56	13.56	11.23	9.56	7.89	5.69
20	15.45	12.89	11.89	10.45	9.46	6.12
24	18.54	16.24	12.56	11.4	10.23	7.23
28	20.65	18.635	13.98	12.03	11.45	8.13
32	21.45	19.56	15.15	13.4	12.49	9.46
36	22.89	20.89	16.45	13.89	12.89	10.23
40	24.76	23.45	19.45	15.45	14.12	11.12
44	28.45	26.56	20.89	17.83	15.47	11.89
48	33.68	32.45	21.46	19.12	17.19	11.98
52	36.45	33.15	21.89	20.13	18.46	12.4
56	38.56	34.5	22.54	21.67	19.57	12.45
60	43.54	40.89	24.7	22.49	20.89	13.4
64	49.98	42.3	26.45	24.17	21.37	13.8

Table 4. Throughput comparison vs. methods

Traffic Load	SGA	DDGA	MAGA	PSO	ABC	GSO
4	0.009	0.0093	0.01	0.015	0.017	0.019
8	0.0095	0.01	0.013	0.018	0.019	0.026
12	0.012	0.0125	0.014	0.019	0.0218	0.028
16	0.0154	0.0164	0.018	0.02	0.023	0.032
20	0.0185	0.0197	0.021	0.023	0.0238	0.0356
24	0.0235	0.0242	0.025	0.027	0.0283	0.0365
28	0.0256	0.0264	0.029	0.032	0.034	0.0385
32	0.0268	0.0272	0.03	0.033	0.035	0.0394
36	0.0289	0.0293	0.032	0.034	0.0358	0.041
40	0.0312	0.0324	0.033	0.036	0.0364	0.0415
44	0.0326	0.0332	0.035	0.038	0.0368	0.0435
48	0.0365	0.0379	0.039	0.042	0.0436	0.0465
52	0.0398	0.041	0.042	0.043	0.045	0.0489
56	0.0412	0.0426	0.044	0.046	0.0475	0.0518
60	0.0423	0.0436	0.045	0.048	0.0489	0.052
64	0.0456	0.0465	0.048	0.051	0.0528	0.0556

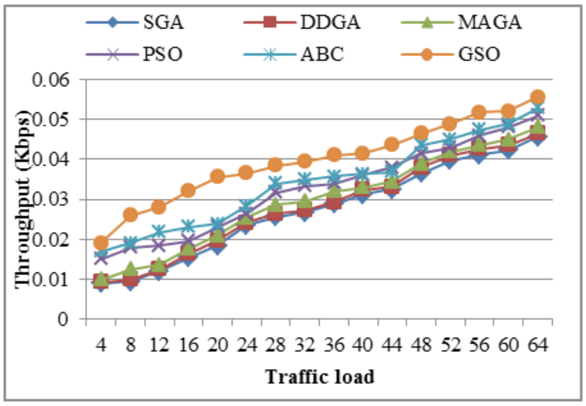

Figure 7. Throughput comparison.

Throughput of the proposed GSO approach is very higher when compared with MAGA and ABC is shown in the Figure 7. The graph shows that maximum throughput has been obtained for the proposed approach. Since the proposed work quality based data services also considered during the OVSF code tree generation, the results of the throughput comparison vs. methods are tabulated in Table 4.

6. Conclusion and Future Work

Orthogonal Variable Spreading Factor (OVSF) code assignment is a fundamental problem in Wideband Code-Division Multiple-Access (W-CDMA) systems, which plays an important role in third generation mobile communications. In the OVSF problem, codes must be assigned to incoming call requests with different data rate requirements, in such a way that they are mutually orthogonal with respect to an OVSF code tree. In order to solve these problems for WCDMA systems in order to reduce the call blocking and increase the spectral efficiency in the system, this work presents a novel dynamic GSO code assignment schema to adapt to changing traffic conditions. An OVSF code tree is a complete binary tree in which each node represents a code associated with the combined bandwidths of its two children. Dynamic code assignment of OVSF-CDMA systems is established based on the GSO. Agents in the GSO algorithm have a finite sensor range is considered as codes in the code tree which defines a hard limit on the local-decision domain used to compute their movements. The simulation results show that GSO provides the smallest blocking probability and largest spectral efficiency in the system when compared to existing methods. It has been observed that proposed GSO is a good choose in terms of additive performance when compare to existing SGA, DDGA, MAGA, PSO, ABC schemes. The performance of the proposed GSO is also examined for different OVSF codes tree in WCDMA. The main limitation of the research work is the utilization of optimization algorithms. Each optimization algorithm has its own drawbacks and hence, it may affect the overall result. Hence, the future enhancement of this research work would be to use hybrid optimization approaches. In future work optimized assignment of code is performed based on the time slots for different order of priority within priority users, which will measure the probability density function and cumulative distributive function. The Level Crossing rate, average fade duration is calculated. Distribution function for path loss is also calculated which will stabilize our system with distribution properties.

7. References

1. 3GPP, Technical Specification TS 25.213 V4.2.0. Spreading and Modulation (FDD). Dec. 2001.
2. Ferng H, et al. An OVSF code tree partition policy for WCDMA systems based on the multi-code approach. IEEE 62nd Vehicular Technology Conference; IEEE; 2005. p. 1212–6.
3. Fantacci R, Nannicini S. Multiple access protocol for integration of variable bit rate multimedia traffic in UMTS/IMT-2000 based on wideband CDMA. IEEE Journal on Selected Areas in Communications. 2000 Aug; 18(8):1441–54.
4. Rouskas AN, Skoutas DN. OVSF codes assignment and reassignment at the forward link of WCDMA 3G systems. PIMRC'02. Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications; Portugal: 2002 Sept. p. 2404–8.
5. Tsaur LF, Lee DC. Symbol rate adaptation and blind rate detection using FOSSIL (Forest for OVSF-Sequence-Set-Inducing Lineages). Proceedings of IEEE International Conference on Communications. 2001. 6:1754–9.
6. Tseng YC, Chao CM. Code placement and replacement strategies for wideband CDMA OVSF code tree management. IEEE Transactions on Mobile Computing. 2002 Oct-Dec; 1(4):293–302.
7. Minn T, Siu K-Y. Dynamic assignment of orthogonal variable spreading-factor codes in WCDMA. IEEE Journal on Selected Areas in Communications. 2000 Aug; 18(8):1429–40.

8. Shueh F, Liu Z-E, Chen W-S. A fair, efficient, and exchangeable channelization code assignment scheme for IMT-2000. Proceedings of 2000 IEEE International Conference on Personal Wireless Communications; 2000. p. 429–33.
9. Park J-S, Huang L, Kuo C-CJ. Computationally efficient dynamic code assignment schemes with call admission control (DCA-CAC) for OVSF-CDMA systems. IEEE Transactions on Vehicular Technology. 2008; 57(1):286–96.
10. Balyan V, Saini DS. Call elapsed time and reduction in code blocking for WCDMA networks. SoftCOM 2009, IEEE 17th International Conference on Software, Telecommunication sand Computer Networks; 2009.
11. Tsai Y-R, Lin L-C. Quality-based OVSF code assignment and reassignment strategies for WCDMA systems. IEEE Transactions on Vehicular Technology. 2009; 58(2):1027–31.
12. Yeh C-C, Wang P-L. Dynamic Bandwidth Allocation and Scheduling for OVSF-WCDMA Systems. 2010 International Conference on Communications and Mobile Computing (CMC); IEEE; 2010.
13. Balyan V, Saini DS. Integrating new calls and performance improvement in OVSF based CDMA Networks. International Journal of Computers & Communication. 2011; 5(2):35–42.
14. Saini DS, Sharma N. Performance improvement in OVSF based CDMA networks using flexible assignment of data calls. International Journal of Computer Networks. 2011.
15. Chao C-M, Tseng Y-C, Wang L-C. Reducing internal and external fragmentations of OVSF codes in WCDMA systems with multiple codes. IEEE Transactions on Wireless Communications. 2005; 4(4):1516–26.
16. Gong D, Li Y. Dynamic system analysis and generalized optimal code assignment of OVSF-CDMA systems. IEEE International Conference on Communications, ICC'08; IEEE; 2008.