International Journal of Multimedia and Ubiquitous Engineering
Vol.12, No.2 (2017), pp.161-170
http://dx.doi.org/10.14257/ijmue.2017.12.2.12

Mobile Multimedia Applications Optimization at Dynamic Data Process Level

Xiaosheng Wang[1], Qian Yu[1], Yanhui Guo[1] and Chunguang Zhou[2]

[1]Information & Technology School, Shandong Women's University, Jinan, China
[2]Computer Science & Technology School, Jilin University, Changchun, China
[1]xswang@sdwu.edu.cn, [2]cgzhou@jlu.edu.cn

Abstract

The mobile intelligent multimedia devices such as smart phones and tablet computers rely on dynamic data type to store and process information, as a result, the memory size demands and energy consumption of the system are increased. In order to better solve this problem in mobile device, the paper presents a novel optimization approach based on a hybrid evolutionary algorithm combined Differential Evolution algorithm with NSGA-II to implement the optimization at data structures process level for mobile multimedia applications according to three-objective optimization models minimizing memory accesses, memory footprint and energy consumption. The hybrid algorithm maintains advantages of the original differential evolution algorithm, only in the choice operating uses fast non-dominance and crowding distance sorting strategies of NSGAII to enhance the performance of multi-objective optimization. Case analysis results shows that the hybrid algorithm has better effect.

***Keywords**: Mobile device, Multimedia Applications, Dynamic data Process Level, Multi-objective Optimization, Hybrid Differential Evolutionary Algorithm*

1. Introduction

Recently, mobile devices such as smart phones and tablet computers, have become indispensable part of our daily life. However, the battery drain problem always bothers us [1]. In addition, most modern mobile devices run multimedia applications, which rely mainly on dynamic data structures to store and process data. Therefore, the key to reducing the power of mobile devices is to optimize memory accesses, memory footprint *etc.* at data structures level for multimedia application of mobile devices.

Up to know, some methods which have been performed are basically based on the power/energy consumption optimization analysis and evaluation for mobile applications [1-2], they have not considered multi-objective optimization at dynamic data structures level. Since dynamic data structures optimization is a multi-objective NP-hard problem, thus multi-objective evolutionary algorithm must be used in applications optimization at data structures. The work has been performed at dynamic data process level for multimedia applications. Jiang *et al.* [3] proposed a method of low power consumption data structure design for embedded applications, but it employed handwork optimization to applications without evolution algorithm. Risco-Martin *et al.* [4, 10] used multi-Objective evolutionary algorithms such as genetic algorithm (GA) to optimizing data structures for embedded applications, but this can generate issues such as the premature converge of the population to a local optimum or general loss of diversity.

In recent years, very many kind of different multi-objective evolution algorithm have been presented. Representative evolutionary algorithms are MOGA, SPEA, PAES, NSGA-II and multi-objective Differential Evolutionary (DE) algorithm and so on. These multi-objective evolutionary algorithm can optimize simultaneously synergistic multiple objectives, and can obtain a set of parallel Pareto optimal solution, but they all have some

ISSN: 1975-0080 IJMUE

problems: MOGA and DE have higher exploring efficiency, but their distribution of Pareto optimal solution is not ideal. SPEA and PAES have a good performance in terms of uniformity of solutions, but they have still space for improvement and high computational complexity. NSGA-II although has the ability to approach optimal Pareto front and ensures that the Pareto optimal solutions obtained have a good spread, but NSGA-II sometimes exhibits weak global optimization capabilities (such as high-dimensional problems encountered), thus it requires a combination of other algorithm to be improved. Differential Evolutionary algorithm (DE) is a random search algorithm with direct parallelism, due to unique memory capability it can dynamically track the current search conditions to adjust its search strategy, it has succeed in many applications. Recently, many scholars studied improvements of DE algorithm to satisfy different requirements [5-6]. Qian *et al.* [7] proposed adaptive DE algorithm to solve the multi-objective optimization problem, Cai *et al.* [8] proposed CDE applying K clustering method to solve the optimization problem of uncertainty, Zhang *et al.* [9] proposed an elite cloud variation based on external archive and adjust mutation operator F and crossover rate R according to successful parameter values.

The paper designs a hybrid Differential Evolutionary algorithm based on DE and cycle crowed distance of improving NSGA-II (CC-DE). The CC-DE overcomes the shortcoming of purely random selection method exists, and retains the more optimal solutions to the next generation to improve optimization efficiency and to ensure diversity solution. Compared with the NSGA-II, the individual breeding method of the CC-DE can ensure the objective vector to search the optimal solutions more effectively in the solutions space. In CC-DE algorithm, mutation and crossover operations in evolutionary operation using DE algorithm, which adaptation mutation operator will change with the evolution generation in order to find the Pareto solution, and the selection operation will use improved cycle sorting and crowding mechanism of NSGA-II to cut shortly the double population with parent and offspring and the relatively best individual is selected into the next generation so that the convergence of the algorithm speed and makes the solution converges to the true Pareto front and uniformly extend along the frontier. We use the CC-DE algorithm to search the optimal dynamic data structures to achieve memory optimization based on three objectives (memory access, memory usage and energy consumption) for mobile multimedia applications. Case study results show the effectiveness of the algorithm.

2. Multi-Objective Optimization Model for Mobile Multimedia Applications

This paper considers a two-Layer memory architecture, which consists of a shared main memory and a L1 data cache. Our objective is to obtain a set of data structure with minimizing memory accesses, memory footprint and energy consumption for each variable in the mobile application, the result (a solution) is represented by a set of pairs (c, s) $\{c_i \in C, s_j \in S\}$, where, C is containers, S is a set of data structures proposed by Baloukas *et al.* [10], *i.e.* S={AR, AR_P, DLL, DLL_O, SLL, SLL_O, DLL_AR, SLL_AR}, where, AR is array, AR_P is array of pointers, DLL is double-link list, DLL_O is double-link list with roving pointer, SLL is single link list, SLL_O is single link list with roving pointer, DLL_AR is double link list of arrays, and SLL_AR is single link list of arrays.

Some parameter is defined as follows: N_e is number of elements saved in the container in the worst case, N_{ae} is average of the number of elements saved, T_e is size of the elements with bytes, T_p is size of the pointers with bytes, N_{rd} and N_{wt} are number of read accesses and write accesses respectively, N_{mis} is Cache misses, E_{mis} is the energy consumption per access to main memory, E_{rw} is the energy consumption per access to the Cache, E_{st} is the static energy by the main memory, N_{rw} is number of reads or writes to the L1 data Cache, and k is scaling factor that depends on the selected type of data structure.

These parameters are obtained by profiling the the application of mobile device. Multi-objective optimization model of mobile multimedia applications at data structure level is following equations:

$$\text{Minimize } MA(c,s)=(N_e\times(N_{rd}+N_{wt})+N_{ae})\times k \tag{1}$$

$$\text{Minimize } MU(\mathrm{c,s})=S_{av}=(T_p+N_e\times(T_p+T_e))\times k \tag{2}$$

$$\text{Minimize } E(\mathrm{c,s})=N_{mis}\times E_{mis}+N_{rw}\times E_{rw}+S_{av}\times E_{st} \tag{3}$$

where, formula(1) represents minimizing memory accesses, (2) represents minimizing memory footprints and (3) represents minimizing the energy.

3. The Optimization Process at Data Structure Level

The Figure 1 describes overall process to perform optimization at data structures level for mobile multimedia applications.

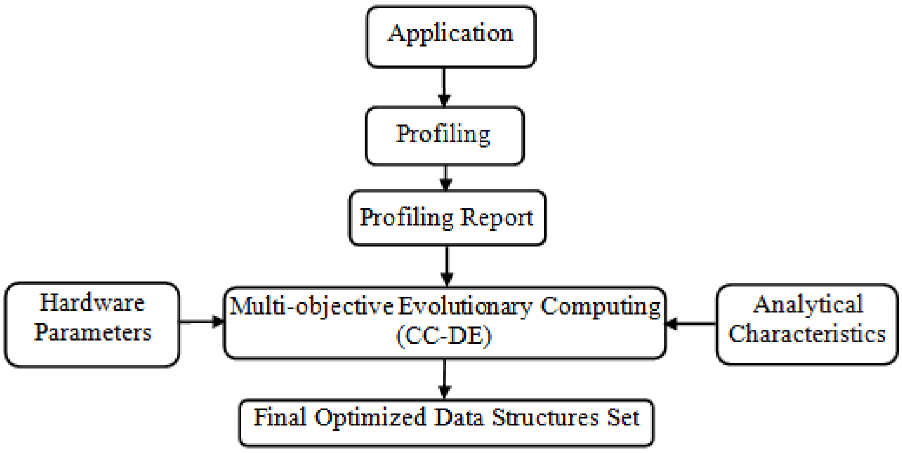

Application
Profiling
Profiling Report
Hardware Parameters
Multi-objective Evolutionary Computing (CC-DE)
Analytical Characteristics
Final Optimized Data Structures Set

Figure 1. Framework of Optimization Process at Data Structures Level for Mobile Applications

Profiling: Through compiling or executing we can profile the application of mobile device to obtain the actual execution information of the application and need create a profiling report of the application where the following information is recorded: number and location of the accesses of an element, addition of an element, removing of an element, the clearing of the container, iterator operations, constructor, destructor, copy constructor, swap operation and so on.

Multi-objective Evolutionary Computing: All the following parameter can be extracted from the profiling report as input parameter of the proposed memory optimization model: the number of candidate variables, number of elements stored in the data structure, average of the number of elements stored, size of the elements in bytes, size of the pointers in bytes, number of real access, number of write accesses and cache misses. The multi-objective evolutionary algorithm CC-DE runs to estimate the quality of the solution (c_i, s_j) using proposed optimization model *i.e.* formula (1) to (3) in the dynamic data structure exploration so as to achieve purpose of minimizing memory access, memory footprint and energy consumption.

Finally, a set of optimized data structure instances {$c_i \in C$, $s_j \in S$ } is obtained, the data structure variable of the optimized application should be replaced by the set of instance computed to implement the optim1zation for memory application in mobile device.

4. The Hybrid Differential Evolution Algorithm

4.1. Differential Evolution Algorithm Description

1) Initialization. In DE algorithm, the initialization of population is operated by the following expression (4):

$$x_{ij,G=0} = rand\ [0,1](upper(x_i)\text{-}lower(x_i))+lower(x_i)$$
$$i=1,2,\cdots,N,\ j=1,2,\cdots,D \tag{4}$$

where, $upper(x_i)$ and $lower(x_i)$ represents the boundary of variable, G is evolutionary generation, N is population size, and D is the dimensions of the individual.

2) Mutation. For each objective individual $x_{i,G}$, variation vector is generated by the formula (5) :

$$v_{i,G+1} = x_{r_1,G} + F*(x_{r_2,G} - x_{r_3,G}) \tag{5}$$

where, $F\in[0,2]$ is the scaling factor. randomly selects r_1, r_2, $r_3\in(1,\ldots,N)$ and $i\neq r_1\neq r_2\neq r_3$.

3) Crossing. The objective vector $x_{i,G}$ and mutation vector v_i are crossed to obtain test vector $u_{i,G+1}$.

if (rand(j)) $\leq CR$ or $j=rand(i)$): $U_{ij,G+1} = v_{ij,G+1}$

$$\text{otherwise: } U_{ij,G+1} = x_{ij,G} \tag{6}$$

where, $rand(j)\in[0,1]$, CR is cross probability constant, $rand(i)\in[1,2,\cdots,D]$.

4)Selection. The offspring is selected by comparing fitness value of the test vector and objective vector, for example of optimization with minimizing, the equation for the selection operation is:

if ($f(u_{i,G+1} < f(x_{i,G})$): $x_{i,G+1} = u_{i,G+1}$

$$\text{otherwise: } x_{i,G+1} = x_{i,G} \tag{7}$$

4.2. Hybrid Evolutionary Algorithm Based on DE and Improved NSGA-II

The standard DE can't be directly applied to the multi-objective optimization problem since its select operation is simple on single-objective optimization. Therefore, the paper designs a hybrid intelligent algorithm based on DE and cycle crowding distance of improved NSGA-II (CC-DE), which overcomes the shortcomings of DE algorithms. In the algorithm, the mutation and crossover operation use DE to change adaptation mutation operator with the evolution generation to find the Pareto solution for mutation operation, but the selection operation will cut short operation for the two-populations with the parent and offspring based on fast non-optimal sorting and crowding mechanism and the relatively best individual is selected into the next generation so as to be fasted convergence speed and makes the solution converges to the true Pareto front and uniformly extended along the frontier. The cycle crowding distance sorting computation with the pseudo-code as follows:

```
cycle_crowddistance_sort ($R$, $t$, $k$)
{
//non-dominated solutions set $R$ includes $t$ solutions, according to crowding distance screening and retained $k$ solutions.
    denoted as $P_{i,distance}$=0, $i$=1,2,$\cdots$,$N$;
    while ($t\neq k$)
   {  //initialize each individual distance:
       for ( $i$=1;$i\leq t$)  $R[i]$.distance=0;
       for (each sub-target $m$) {
```

```
    R=sort(R, m);
    R[0].distance=∞; R[t].distance=∞;
    for (i=2;i≤ t-1)
        $R[i].distance=R[i].distance+(R[i+1].m-R[i-1].m)^2$;
    }
    // finds out the solution with the minimum crowed distance:
    i=min_distance(R);
    R[i]=Φ; //deletes the solution with the minimum crowed distance
    t=t-1;
  } //end while
} //end function
```

4.3. Performance Test and Results Analysis

In order to test the performance of CC-DE algorithm based on DE and NSGA-II, we select multi-objective test functions ZDT1~ZDT3 [11-12] to explore optimization, which are widely used in multi-objective optimization, where, Pareto frontier of ZDT1 is convex function, ZDT2 is non-convex function, ZDT3 is non-continuous function.

(1) Performance metrics

In order to compare performance of the CC-DE and other optimization algorithm, we need to evaluate the obtained set of non-dominated solutions in following two performance metrics which are used widely.

- Coverage metric

$$r=\sum_{i=1}^{n} d_i / n \tag{8}$$

where, n is the number of non-dominated vectors obtained by algorithm, d_i is Euclidean distance between the non-dominated frontier *P'* obtained and real Pareto front *F'* in the most recent members.

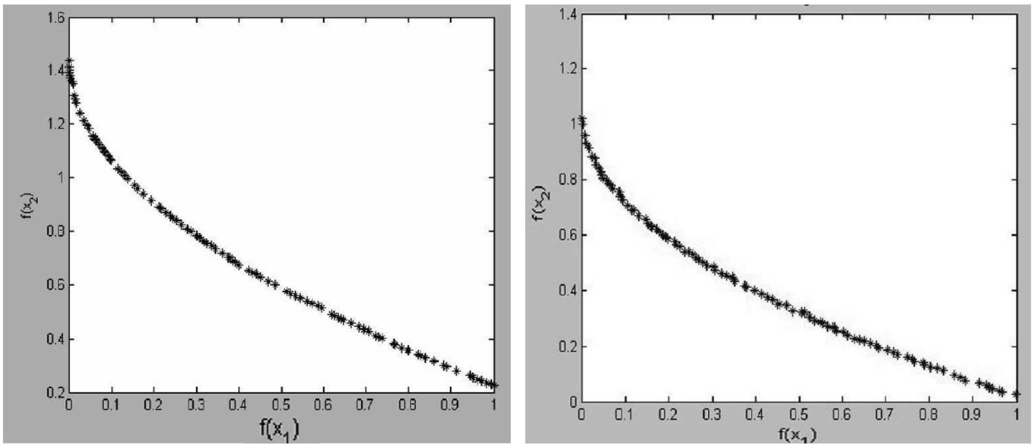
- Diversity metric

$$\delta=\frac{d_f+d_i+\sum_{i=1}^{n}(d_i-d_a)}{d_f+d_i+(n-1)d_a} \tag{9}$$

where, d_i is same as formula (8) in coverage metric, d_a is the average of all d_i, d_f and d_1 are the distance between extreme solutions from real Pareto front *F'* and boundary solutions obtained from Pareto front *P'*.

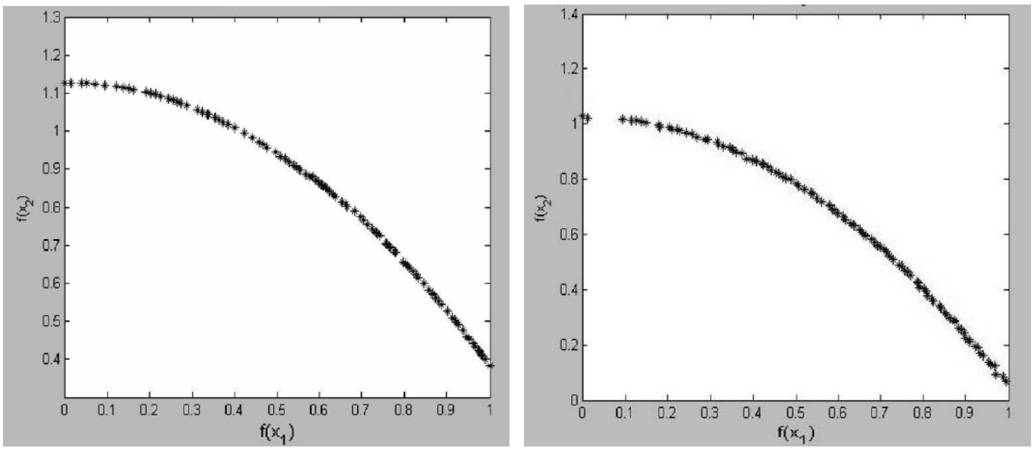
(2) The test results and comparative analysis

NSGA-II is multi-objectives optimization elitist algorithm [13], and it has the good serviceability to the complex non-linear problem, and it is often used as the comparison object with other evolutionary algorithms. Thus, we will compare the CC-DE with NSGA-II in performance of test problem ZDT1-ZDT3. Due to limited space, we only give comparison results of performance optimization for test function ZDT1 and ZDT2. In the test, in order to ensure algorithms comparative consistency for each test function, the parameters are as follows: the population size is set to 100, iterative times is 200, mutation operator *F* is 0.4 and crossover *CR* is 0.2 in CC-DE algorithm. The comparison results show in Figures 2 and 3.



(a) (b)

Figure 2. Test Function ZDT1 Comparative Performance Optimization Results. (a) NSGA-II, (b) CC-DE



(a) (b)

Figure 3. Test Function ZDT2 Comparative Performance Optimization Results. (a) NSGA-II, (b) CC-DE

The comparison results show that in the case of real coding and the same population size and maximum evolution generation, the results of CC-DE algorithm is better than NSGA-II, and the quality and quantity of the optimal solutions are more than the NSGA-II. It indicates that the hybrid algorithm is more effective to improve the shortcomings of the single algorithm, and the performance of the CC-DE algorithm is improved.

Simultaneously, we compare CC-DE with NSGA-II in convergence metric r and diversity metric δ for the three test problem, the results are shown in Table 1 and Table 2, where these values are average obtained by running 30 times. It shows that the CC-DE is better than NSGA-II in convergence and diversity of all the test problems. In the three test functions, the CC-DE achieves the two goals of multi-objective optimization *i.e.* converging to the real Pareto Frontier and distributing uniformly along the Pareto frontier. It indicates that the Hybrid intelligent algorithm that the CC-DE combined with NSGA-II can solve effectively the multi-objective optimization problem and has a very broad application space in this regard.

Table 1. Comparison of Convergence *R*

algorithm	ZDT1	ZDT2	ZDT3
NSGA-II	0.03348±0.00475	0.07239±0.03169	0.11450±0.00494
CC-DE	0.02041±0.00001	0.03751±0.00030	0.00112±0.00010

Table 2. Comparison of Diversity Δ

algorithm	ZDT1	ZDT2	ZDT3
NSGA-II	0.39031±0.00187	0.43077±0.00472	0.73854±0.01971
CC-DE	0.20375±0.00018	0.20261±0.00310	0.40211±0.00490

4.4. Multi-Objective Optimization Model Computing Process Based on CC-DE Algorithm

Multi-objective (*c*, *s*) optimization model computing method based on the CC-DE algorithm for mobile multimedia applications at data structures level is as follows:

Step 0: Initialization. Sets *F*, *CR etc*., optimizing variables (data structures) uses real value encoding, generates initial population.

Step 1: Calculates three objective value respectively, *i.e*. memory access *MA*(*c*, *s*), memory footprint *MU*(*c*,*s*) and energy consumption *E*(*c*,*s*).

Step 2: Executes the function cycle_crowddistance_sort( ) to calculate the Pareto Front and crowding distance for each individual, and sorts with the front.

Step 3: Differential evolution. While the algorithm end criteria is not satisfied, does binomial crossover and mutation for the populations, and obtains a mixed population with offspring and parent ,the population size becomes 2*N*.

Step 4: NSGA-II operation. According non-super-sorting of NSGA-II and improved cycle crowding distance mechanism, truncates the population to produce the next generation of size *N*.

Step 5: While the given maximum number of iteration is reached, the algorithm is ended and gives the result. Otherwise repeat Step 1-Step 4.

5. Case Test and Analysis

We have carried on our methodology to explore the optimized configuration of data structures variables for a 3D Game Simulation. The Pareto front with the main combinations of data structure implementations for the Game Simulation is obtained by using our CC-DE algorithm, simultaneously compared with the results obtained by NSGA-II in Figure 4. The figure shows that the presented approach using CC-DE can provide a range of possible data structure implementation solutions to the designer according to the weight defined for each optimization measure (*i.e*. memory accesses, memory footprint and energy consumption in our case).

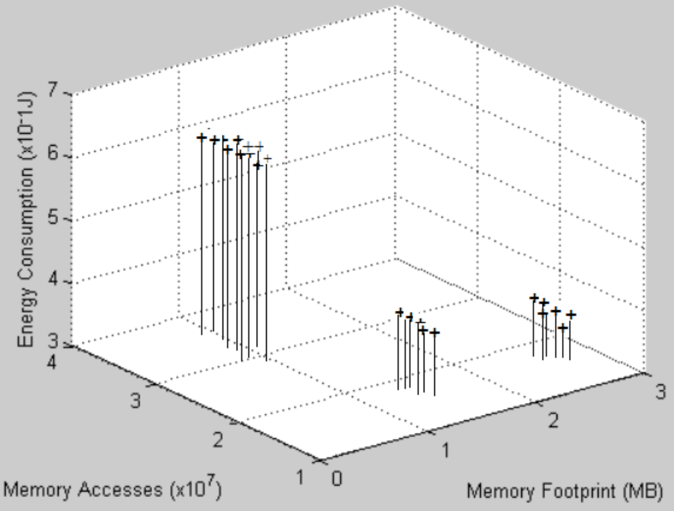


Figure 4. 3D Pareto Curve Obtained Using the Proposed CC-DE Approach

Figure 5 is presented to illustrate the optimization process. In this case test, the set of dynamic data structures was successively completed using AR, AR_P, DLL, DLL_O, SLL,SLL_O, DLL_AR and SLL_AR in S. The three objectives have been normalized to the AR and described with logarithmic scale. It shows that the level of optimization and final gains are achieved after applying CC-DE algorithm according to the proposed optimization framework in Figure1.

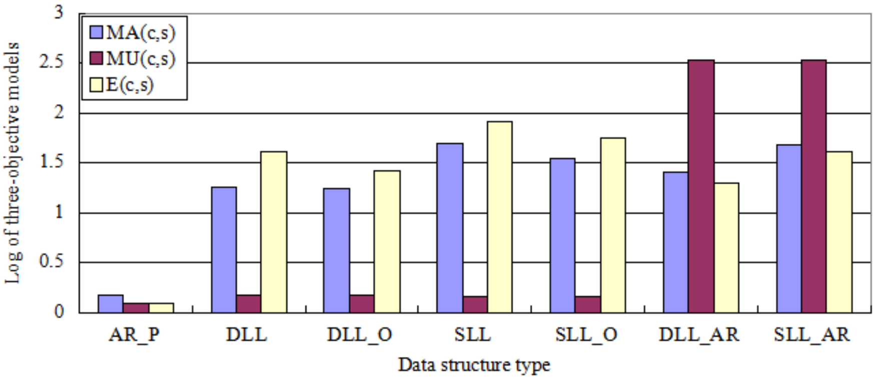


Figure 5. The Results of Three-Objective Optimization Model Computation from Various Sets of Solution

6. Conclusion

This paper analyzes the problem of solving multi-objective (c,s) optimization model with memory accesses, memory footprint and energy minimization criterion for mobile multimedia applications, and designs a hybrid DE algorithm called CC-DE which is combined advantages of DE algorithm with the cycle crowed distance of improving NSGA-II to solve the multi-objective optimization problem of the mobile multimedia applications. This approach simplifies the exploration effort of multi-layered data structure implementations in an automation way.

Experimental results shows that the presented approach using CC-DE algorithm has better performance of convergence and the diversity of solutions, and we achieved level of optimization and final gains by performing our method for an actual mobile multimedia applications, and can provide a range of possible data structure implementation solutions to the designer according to the weight defined for each optimization measure *i.e.* memory accesses, memory footprint and energy consumption in our case.

Acknowledgements

This research is supported by Natural Science Foundation of Shandong Province (ZR2013FL025) and Symbolic Computation and Knowledge Engineering of Jilin University Key Laboratory of Ministry of Education (93K-17-2010-K05).

References

[1] L. Youhuizi, C. Hui and S. Weisong, "Power behavior analysis of mobile applications using Bugu", Sustainable Computing: Informatics and Systems, vol. 4, no. 3, **(2014)**.

[2] S. Hao, D. Li, W. G. J. Halfond and R. Govindan, "Estimating mobile application energy consumption using program analysis", 35th International Conference on Software Engineering (ICSE2013), San Francisco, CA, USA, **(2013)**.

[3] L. Jiang, C. Peng, H. Cai, Y. Chen and X. Chen, "Low power consumption data structure design for embedded applications in real-time systems", IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, **(2012)**.

[4] J. L. R. Martin, D. Atienza, R. Gonzalo and J. I. Hidalgo, "Optimization of dynamic memory managers for embedded systems using grammatical evolution", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'09), Montreal Quebec Canada, **(2009)**.

[5] S. Das and A. Abraham, "Differential evolution using a neighborhood-based mutation operator", IEEE Transactions on Evolutionary Computation, vol. 3, no. 3, **(2009)**.

[6] X. S. Wang and G.C. Xu, "Robot Path Planning Based on Chaos Concise Differential Evolution and RFNN Control", The Open Automation and Control Systems Journal, vol. 6, no. 1, **(2014)**.

[7] W. Y. Qian and A. J. Li, "Adaptive differential evolution algorithm for multi-objective optimization problems", Applied Mathematics & Computation, vol. 201, **(2008)**, pp. 1-2.

[8] Z. H. Cai, W. Y. Gong, C. X. Ling and H. Zhang, "A clustering based differential evolution for global optimization", Applied Soft Computing, vol. 11, no. 1, **(2011)**.

[9] J. Q. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution algorithm with optional external archive", IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, **(2009)**.

[10] C. Baloukas, J. L. R. Martin, D. Atienza, C. Poucet, L. Papadopoulos, S. Mamagkakis, D. Soudris, H. J. Ignacio, F. Catthoor and J. Lanchares, "Optimization methodology of dynamic data structures based on genetic algorithms for multimedia embedded systems", Journal of Systems and Software, vol. 82, no. 4, **(2009)**.

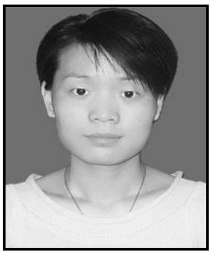
[11] K. Deb, "Multi-objective genetic optimization: problem difficulties and construction of test problems", Evolutionary Computation, vol. 7, no. 3, **(1999)**.

[12] E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach", IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, **(1999)**.

[13] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II", IEEE Transactions Evolutionary Computation, vol. 6, no. 2, **(2002)**.

Authors

Xiaosheng Wang, He received his Diploma and M.Sc. Degree in Computer Science and Technology from Jilin University, China, in 1988 and 2003, respectively. He is currently a professor, engaged in teaching and research work in Computer Engineering and Applications in Information Technology School of Shandong Women's University. His research interests include intelligent computing, embedded systems performance and power optimization.



Qian Yu, She received her Diploma and M.Sc. Degree in Computer Science and Technology from Shandong University, China, in 1998 and 2005, respectively. She is currently engaged in teaching and research work in Information Technology School of Shandong Women's University. Her research interests include data structures and algorithms, and machine learning.

Yanhui Guo, He received my Diploma and M.Sc. Degree in Computer Science and Technology from Shanxi Normal University, China, in 2006 and 2009, respectively. He is studying for PhD in Computer Science and engaged in teaching and research work in Information Technology School of Shandong Women's University. His research interests include intelligent algorithms and middleware technology.

Chunguang Zhou, He is a professor at the College of Computer Science and Technology, Jilin University. His research interest covers computational intelligence, pattern recognition and image processing.