System Level Optimisation of a 5.8 GHz ETC Receiver using Memetic Algorithm and PSO Method

Lai Jiang, Yan Yin, Hang Yu, Yan Li, and Zhen Ji

***Abstract*—Radio Frequency (RF) front-end system level optimization is a multi-dimensional optimization problem and is usually an arduous and experience-based work because all system parameters trade with each other. Since memetic algorithm (MA) and particle swarm optimization (PSO) present efficient ways for multi-dimensional optimization problems, the two methods were used to optimize the key system parameters of a 5.8 GHz RF front-end. The transfer function for the output signal-to-noise ratio (SNR_{out}) was firstly derived based on the gain, noise figure and inter-modulation product of each sub-block component, and it was used as the required fitness function for the optimization process. Both of the MA and the PSO methods provide a fast convergence, and the design parameters were obtained for the optimum SNR_{out} values in both cases. The effectiveness of the two methods was compared. In order to verify the optimized parameters, ADS simulations was used and the final results show that the proposed methods work efficiently.**

***Index Terms* — RF front-end, fitness function, Memetic Algorithm, Particle Swarm Optimization, ADS simulation**

I. Introduction

Traffic congestion has become a prominent problem in many major cities in China. Thanks to the rapid development of wireless communication technologies, it is possible to employ radio frequency identification (RFID) technology to construct urban intelligent traffic control system, which aims at improving the traffic efficiency. Many new techniques, such as the floating car data transfer (also known as floating cellular data), inductive loop detection and video vehicle detection, have been developed for dynamic traffic measurements [1-3]. Among those, electronic tolling collection (ETC) is known as an efficient method for congestion alleviation. In an ETC system, the vehicle is allowed to be charged without slowing down, and therefore the traffic speed can be largely improved. The Chinese standard of Dedicated Short Range Communications (DSRC) set the ETC operating frequency to be 5.8 - 5.9 GHz, and a Radio Frequency (RF) receiver (Fig. 1) that complies with DSRC standard [4] is the key component for such a system.

The pivotal part of the ETC receiver is the RF front-end. The architecture of the ETC receiver is given in Fig. 1. In a receiver, the RF front-end is the first block after antenna and it processes the incoming signal directly at high frequency. The performance of the RF front-end is crucial to the entire ETC system, since its gain, noise figure and inter-modulation product all contribute to the resulting signal-to-noise ratio (SNR) at the receiver baseband. The design of ETC RF front-end is not a trivial work as it is a multi-dimensional optimization problem and all parameters trade with each other. No rigorous procedure available, former techniques often set them according to the designer's experience and great design effort is generally required.

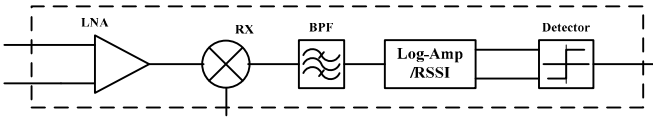

Fig.1 Architecture of 5.8 GHz RF receiver.

Evolutionary algorithms have become widely used for optimizing multi-dimensional problems. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are the modern stochastic extensive search strategies that used most conventionally [5], and both of them can provide a high quality solution within tractable time even on complex circumstances through simulating a natural evolution process [6].

An issue of the GA method is that it is usually not well suited for fine-tuning structures which are close to optimal solution. Hybridizing GA with local search (also known as Memetic operators or 'Meme'), which is commonly referred as Memetic algorithm (MA) [7, 8], can be used in such situations. The MA process, which is a population-based approach, is capable of fine tuning and improving the convergence of the GA. The efficacy of the MA method has already been shown on a broad species of genuine world problems [9].

The PSO is also a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling [10]. Although the PSO shares many similarities with the MA method, there are still differences between the two methods. Unlike the MA, the PSO has no evolution operators such as crossover and mutation. Moreover, there are few parameters to adjust in the PSO [11], which makes it more attractive. As the PSO works well in a wide variety of applications with slight variations, it is a fast and cheap way compared with other methods.

As described in Fig. 1, a RF receiver generally consists of a low noise amplifier (LNA), a mixer, a band pass filter (BPF) and a demodulation block (including RSSI and the detector). In this paper, the MA and PSO methods are used to help search the systematic level parameters of an ETC

Manuscript received July 31, 2011; revised August 15, 2011. This work is partially supported by the projects 60901016 and 61001185 supported by NSFC, the project 10151806001000016 supported by Guangdong R/D Fund, and the SRF for ROCS, SEM.

L. Jiang, Y. Y and H. Yu in are with college of Computer Science and Software Engineering, Shenzhen University and Shenzhen City Key Laboratory of Embedded System Design, Shenzhen, Guangdong, China, P.R.

Y. Li and Z. Ji are with college of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China, P.R.

front-end aiming at maximizing the output signal-to-noise ratio (SNR_{out}). Since the SNR_{out} is primarily determined by the receiver fronted consisting of the LNA and the mixer, a fitness function including its key parameters, such as the gain, the noise figure and the IIP3 of each circuit component, is derived. Then, the MA method and the PSO method were used to automatically evolve parameters in order to get the best SNR_{out}. To validate this process, the ETC receiver front-end with the optimized parameters is also modeled in ADS, and the efficacy of the algorithm method in receiver front-end optimization is proved by comparing the simulation results.

This paper is divided into five sections. In Section II, the key parameters of the receiver front-end are used to derive the overall SNR_{out}, the resulting equation is also used as the fitness functions required for the algorithm method. The receiver frontend optimization process using the algorithm method and the difference between the two methods are described in detail in section III. The validity of the optimization process was proved by ADS simulation and discussion is given in section IV. Finally, concluding remarks are given in section V.

II. Fitness Function Derivation

A. RF front-end SNR_{out} fitness function

Primarily because of its robustness, simplicity, and the corresponding low power consumption, direct conversion architectures are generally chosen in ETC receiver designs. The systematic architecture of a direct-conversion receiver front-end is shown in Fig. 2. In this study, the signal amplitude at the input of each circuit component is represented as A_{s1} and A_{s2}, and the noise amplitude as A_{n1} and A_{n2}, respectively.

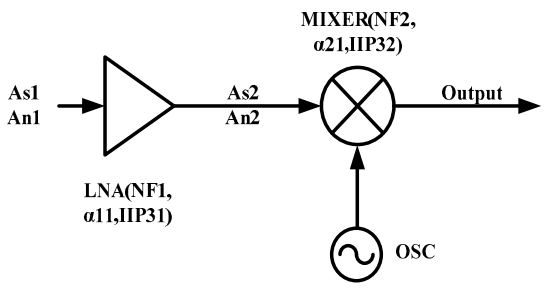

Fig.2 Direct conversion RF front-end in the ETC system.

The SNR_{out}, which is used to represent the communication quality, is first deduced based on key parameters of the LNA and the Mixer. Totally six variables, including the small signal gain *Gs*, noise figure *NF* and input third order intercept point *IIP3* of both the LNA and the mixer, are used for this derivation (index 1, 2 represent the LNA and the Mixer, respectively). The signal and noise power at the output can be calculated separately if the accurate power gain and the noise figure of each component were known.

The *NF* value of a system is described as the ratio between the output noise power and the power generated by the system according to the input noise, and can be written as:

$$NF = N_{out.total} / (N_{in} \times G_n) \tag{1}$$

where $N_{out.total}$ is the output noise power, N_{in} is the input noise power, and G_n is the noise power gain of the system.

It can be shown that the SNR_{out} can be related to the NF as

$$\begin{aligned} NF &= \frac{N_{out.total}}{(N_{in} \times G_n)} = \frac{N_{out.total}}{(N_{in} \times G_n)} \times \frac{S_{in}}{S_{in}} \\ &= \frac{S_{in} / N_{in}}{G_s \times S_{in} / N_{out.total}} \times \frac{G_s}{G_n} = \frac{SNR_{in}}{SNR_{out}} \times \frac{G_s}{G_n} \end{aligned} \tag{2}$$

, where S_{in} is the input signal power, SNR_{in} and SNR_{out} are the signal to noise ratio at the input and the output, G_s and G_n represent the gain of the signal and the noise, respectively.

To derive SNR_{out}, (2) can be apply to each circuit component. As the LNA is the first stage of the receiver, its NF-to-SNR expression is firstly calculated (The output SNR of the LNA is noted as SNR_{out1}). Because the limited circuit linearity, the noise gain is generally different from the signal gain. Therefore, SNR_{out1} can be written as

$$SNR_{out1} = \frac{S_{in}}{N_{in}} \times \frac{G_{n1}}{G_{s1}} \times \frac{1}{NF_1} = SNR_{in} \times \frac{G_{n1}}{G_{s1}} \times \frac{1}{NF_1} \tag{3}$$

where S_{in} is the input signal power and the N_{in} is the input noise power.

When the input signal amplitude A_{s1} and the noise amplitude A_{n1} are given, the noise and signal amplitude at the component output can be obtained by using the more-generalized circuit gain including the 2nd order effect, as in (3) and (4).

$$\sqrt{G_{s1}} = \alpha_{11} + \frac{9}{4}\alpha_{13} \times (A_{s1})^2 \tag{4}$$

$$\sqrt{G_{n1}} = \alpha_{11} + \frac{9}{4}\alpha_{13} \times (A_{n1})^2 \tag{5}$$

where α_{11} is the small signal gain of LNA, α_{13} is the nonlinear system coefficients.

Coefficient α_{13} can be calculated from the specified $IIP3_1$ and α_{11}, as

$$\alpha_{13} = -\frac{1}{3} \times \frac{\alpha_{11}}{(IIP_{31})^2} \tag{6}$$

where α_{21} is the gain of mixer.

SNR_{out1} can then be obtained as in (7).

$$SNR_{out1} = \frac{S_{in}}{N_{in}} \times \frac{G_{s1}}{G_{n1}} \times \frac{1}{NF_1} = SNR_{in} \times \frac{G_{s1}}{G_{n1}} \times \frac{1}{NF_1} \tag{7}$$

Similarly, the noise and signal gain (G_{s2} and G_{n2}) can be obtained using the nonlinear coefficients of the Mixer:

$$\sqrt{G_{s2}} = \alpha_{21} + \frac{9}{4}\alpha_{23} \times (A_{s2})^2 \tag{8}$$

$$\sqrt{G_{n2}} = \alpha_{21} + \frac{9}{4}\alpha_{23} \times (A_{n2})^2 \tag{9}$$

The input signal-to-noise ratio of the Mixer is SNR_{out1}. Following the same procedure, the output SNR of the mixer, which is the targeted SNR_{out} can be written as (10).

$$SNR_{out} = SNR_{out2} = SNR_{out1} \times \frac{G_{s2}}{G_{n2}} \times \frac{1}{NF_2} \tag{10}$$

Plugging (4) to (9) into (2), a closed formed expression relating SNR_{out} can be derived

$$SNR_{out} = SNR_{in} \times \frac{1}{NF_1} \times \frac{1}{NF_2} \times \left(\frac{1 - \frac{3}{4} \times \frac{1}{IIP_{31}^2} \times A_{s1}^2}{1 - \frac{3}{4} \times \frac{1}{IIP_{31}^2} \times A_{n1}^2} \right) \times \left(\frac{1 - \frac{3}{4} \times \frac{1}{IIP_{32}^2} \times A_{s2}^2}{1 - \frac{3}{4} \times \frac{1}{IIP_{32}^2} \times A_{n2}^2} \right) \tag{11}$$

In (11), signal amplitude at the Mixer input A_{s2} can be uniquely determined by the signal amplitude at the LNA input A_{s1} and the LNA gain G_{s1}.

With specified SNR_{in}, SNR_{out} can be determined by total six independent variables. In order to optimize the receiver front-end for the best SNR_{out}, MA method can be used due to its efficacy shown in multi-dimensional problem optimization, and (11) can be used as the required fitness function.

III. Implementation of the MA and the PSO Methods

A. Memetic algorithm implementation

TABLE 1. Initial Parameters for the MA

NF_1	NF_2	G_{s1}
0~15 dB	5~20 dB	0~15 dB
G_{s2}	IIP_{31}	IIP_{32}
-15~10 dB	-10~3 dBm	-10~10 dBm

The MA method is used in this work to search for the optimum SNR_{out} values that achieve the best system performance. In order to have a large population that guarantees the occurrence of good genes, the individual number of the population is set to 10000. The max number of the generation is set to 90000, which ensures that the results are close enough to the real max value. Besides, the mutation rate and the crossover rate are originally set to 0.01 and 0.8. The input SNR value is set to -7.18 dB and the initial individuals of the 6 parameters were randomly generated according to their ranges (as given in Table I). The fitness value of every individual, which undergoes selection in the presence of variation-inducing operators such as mutation and recombination (crossover), is then computed.

The MA's flow chart is shown in Fig. 3. After initialization, the first generation is generated randomly, we get every individual's locally optimum value through local search which is Davies, Swann, and Campey with Gram-Schmidt orthogonalization (DSCG) in the spirit of Lamarckian learning [12, 13]. The local search's program flow chart is shown in Fig.4. Once the local search program is called, the independent variables' neighborhood (p) is initialized. Through traversing the variables' neighborhood, the local fitness value f(xb) could be found.

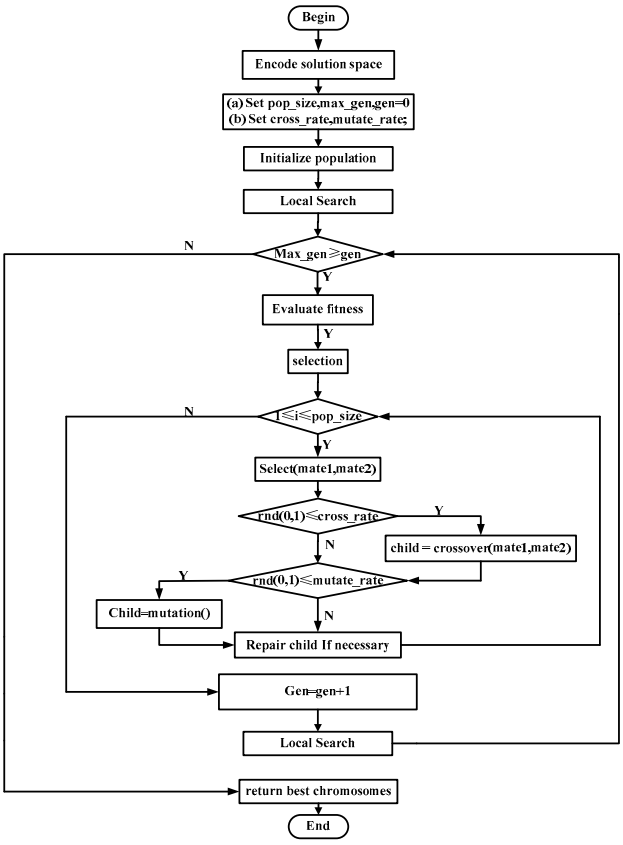

Fig.3 The program flow chart of the MA method.

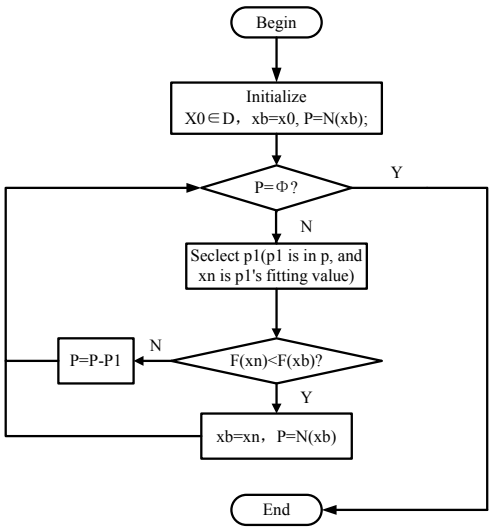

Fig.4 Local Search's flow chart.

The selected individuals from the initial search are then crossed according to the given probability (Cross Rate, *CR*) to generate offspring. Mutations are then generated according to the given mutation rate (*MR*). Both cross-generated offspring and mutations are used to construct the grand generation. In every generation, a local search is performed, and the fitness values of the new generation are evaluated. The evolution continues until the generation of MA turns into Max_gen, and the fitness value reaches its maximum. In the latest generation, the best individual is considered as the one that maximizes the fitness value.

In the study, three groups of different *CR* and *MR* values are used in order to search for the best solution, and the resulting SNR_{out} values are given in Table 2.

TABLE 2. THE PARAMETER VALUES AND THE *SNRout* OBTAINED IN MA

	MR	***CR***	SNR_{out}
1	0.01	0.8	0.0070109
2	0.08	0.8	0.0070109
3	0.01	0.5	0.0070109

The convergence curves for the three studied cases are given in Fig. 5. As shown in the figure, all three groups converge to the same maximum value after ~ 60000 generations. High *MR* value can help generations jump out the locally optimals more quickly, and therefore curve 2 shows a faster converging rate compared with curve 3. With a high *CR* value, curve 3 achieves the fastest convergence among the three studied cases. The optimal solutions found by the MA were given in Table 3.

TABLE 3. THE PARAMETER VALUES AND THE *SNRout* OBTAINED IN MA

NF_1	NF_2	G_{s1}
0.19 dB	15 dB	14.9 dB
G_{s2}	IIP_{31}	IIP_{32}
8 dB	-10 dBm	-10 dBm

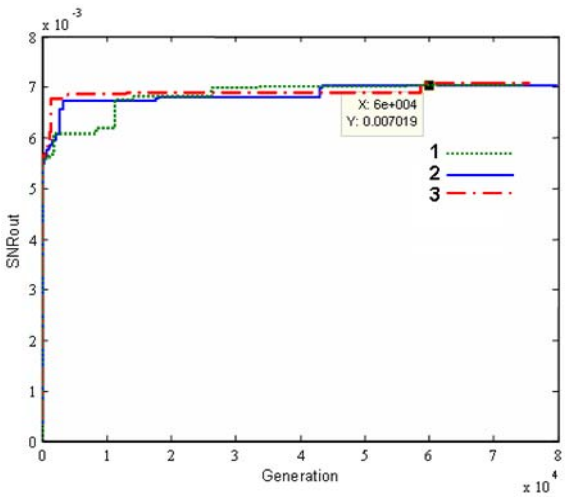

Fig.5 The MA convergence curves, the curves 1, 2 and 3 correspond to the different *MR* and *CR* combinations of Table 2.

The convergence curves for the three studied cases are given in Fig. 5. As shown in the figure, all three groups converge to the same maximum value after ~ 60000 generations. High *MR* value can help generations jump out the locally optimals more quickly, and therefore curve 2 shows a faster converging rate compared with curve 3. With a high *CR* value, curve 3 achieves the fastest convergence among the three studied cases. The optimal solutions found by the *MA* were given in Table 3.

B. PSO Implementation

The PSO method is the second way used in this work to search for the optimum SNR_{out} values that achieve the best system performance.

The system is initialized with a population of random solutions, which is called particles, and searches for optima by updating generations. The initial population x is set to have 5 particles, and, the number of generation is set to 10000, which is large enough for the PSO method since it gets convergent at about the 5th generation. Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. The initial problem space of PSO method is also depicted in Table I. (The fitness value is also stored.) This value is called pbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest.

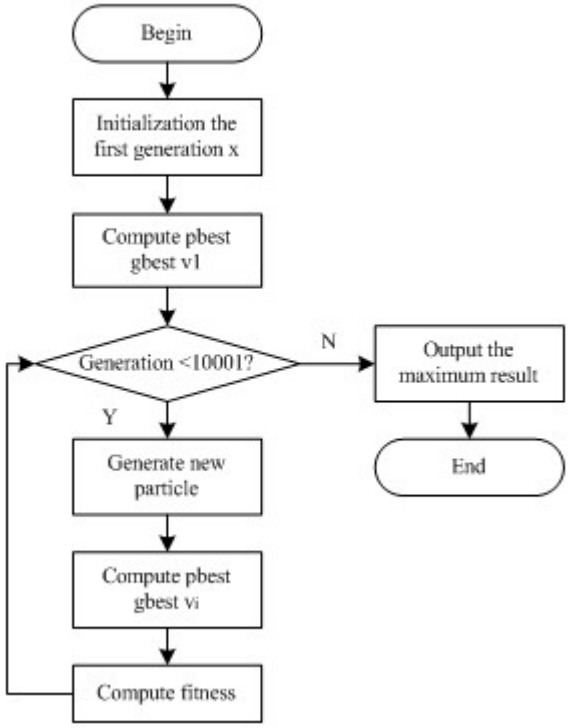

Fig.6 Local Search's flow chart.

The most important step of the PSO process is to generate new particles, which consists of changing the velocity of (accelerating) each particle and updating the position and velocity of each particle. The velocity update formula and the position update formula are given by (12) and (13), respectively.

$$v(j+1) = w(j+1) * rand(1,1) + w(j+1) * velocity \quad (12)$$

$$x(j+1) = x(j) + v(j+1) \quad (13)$$

In (12), *w(j)* is a weight used to accelerate the velocity which is in the correct direction to improve the value of pbest. The *velocity* is a variable that follows the rules indicated in (14). By this means, the particle is insured to traverse towards the correct direction.

$$velocity = v \quad \text{when } fitting(j+1) > fitting(j) \quad (14)$$
$$= 0 \quad \text{else}$$

In the studies, three groups of different "weight" values were used in order to search for the best solution, and the resulting SNR_{out} values are given in Table 4. Fig.7 gives the corresponding convergence curves. The process was

convergent after 2 generations for all of the three weight values and the corresponding SNR_{out} value was about 0.007.

TABLE 4. THE PARAMETER VALUES AND THE *SNRout* OBTAINED IN PSO

	1	2	3
Weight	1.5	2	3
$\boldsymbol{SNR_{out}}$	0.00709223	0.00709223	0.00709223

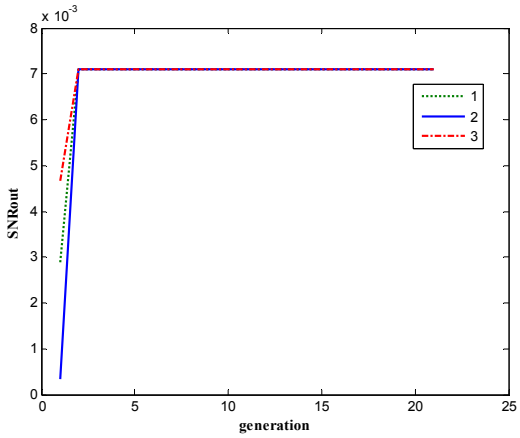

Fig.7 The convergence curve of PSO. The curves 1, 2 and 3 correspond to the different weight values of Table 4.

C. *Comparision of the MA and the PSO method*

The results obtained by the MA and PSO methods are summarized in Table 5. In order to evaluate the CPU time needed for the two optimization methods, the convergence times were obtained by the time computing function in Matlab. The values are also given in Table 5.

TABLE.5 RESULTS OBTAINED BY THE MA AND PSO METHODS

Parameters	Ranges	Best solution of MA	Best solution of PSO
NF_1	1.12~31.6	1.1189056	1.12
α_{11}	1~5.62	5.6198879	5.62
$IIP3_2$	0.1~1	0.0999788	0.1
SNR_{out}		0.0070108	0.00709223
Convergence time		22.625 s	7.52 s

For both methods, the same input SNR value (-7.18 dB) was used as the input signal. The initial individuals of the 3 parameters (NF_1, α_{11} *and* $IIP3_2$) were randomly generated according to their ranges (as specified in Table 5). In the MA method, the fitness values of every individual were computed in the presence of variation-inducing operators such as mutation and recombination (crossover). In the PSO method, their values were obtained through the process that pbest gradually grows to gbest. The SNR_{out} values obtained by the two methods are very close, and the corresponding parameters are almost the same. Since both of the two methods converge on the same value, it could be concluded that the right SNRout and parameters were obtained. However, the efficacies of the two methods differ a lot. According to the convergence time given in Table 4, the PSO converges much faster. While it took about 22 seconds for the MA to get convergence, only about 1 second was needed for the PSO evolution process. The main reason is that a much less generation number is sufficient for the PSO method, and the PSO swarm is smaller than the MA population.

IV. VALIDATION

In order to validate the efficacy of the optimization method, a RF frontend with algorithm-optimzed cirucit paramters is simulated using the Advanced Design System (ADS). The simulation testbench is given in Fig. 8.

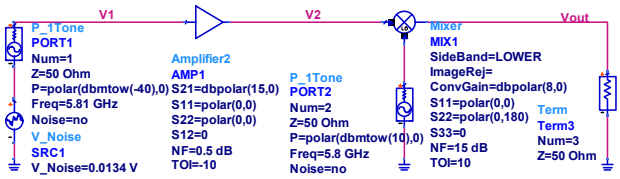

Fig.8 The ADS testbench.

To show the circuit parameters from the algorithm optimization can indeed result to the best output signal-to-noise ratio, we simulated the SNR_{out} with various G_{s1} values, while keeping the other parameters unchanged. The simulation results are shown in Fig.9. Clearly, the SNR_{out} rises monotonously with the small signal gain of the LNA, and it can be proved that the value of G_{s1} found by the algorithm method is the optimum.

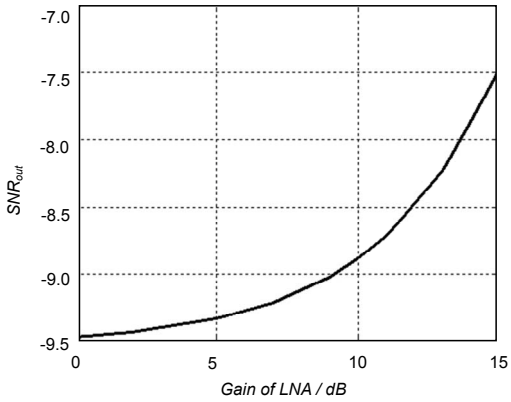

Fig. 9 The SNR_{out} vs. the G_{s1}.

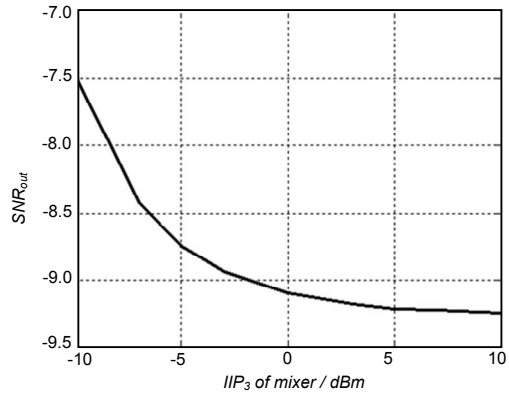

Fig. 10 The SNR_{out} vs. the $IIP3_2$.

For a second study, the inter-modulation product of the mixer $IIP3_2$ is varied, while the other parameters are kept unchanged. Fig. 10 plots the simulation results. It can be seen that the SNR_{out} decreased in a mononous maner with the $IIP3_2$, and the best performance can be achieved when $IIP3_2$ is set to -10 dBm, which is in compliance with the algorithm-optimized result.

V. CONCLUSION

The MA and the PSO were used for systematic level optimization of the RF front-end of a 5.8 GHz ETC receiver. The output SNR of a RF front-end is derived from 3 key circuit parameters, and the resulting mathematic expression is used as the fitness function required for the algorithm methods. The PSO has a higher efficacy than the MA in solving this optimization problem. To show the efficacy of the two methods, the receiver front-end with the algorithm-optimized parameters was simulated in ADS, and the simulation results validate the proposed approach. This method could be extended and used for complete receiver optimization in future studies.

REFERENCES

[1] T. Monahan, "'War Rooms' of the Street: Surveillance Practices in Transportation Control Centers," The Communication Review 10, 2007, pp. 367–389.

[2] A. Gühnemann, R. P. Schäfer, K. U. Thiessenhusen and P. Wagner, "New Approaches to Traffic Monitoring and Management by Floating Car Data," The 10th World Conference on Transport Research, 2004.

[3] P. J. Tarnoff, D. M. Bullock, E. Stanley et al., "Continuing Evolution of Travel Time Data Information Collection and Processing," Transportation Research Board Annual Meeting, 2009.

[4] GB/T 20851-2007, "Electronic toll collection-Dedicated short range communication Interface with Roadside Unit and Lane Controller", 2007.

[5] J. H. Holland, Adaptation in natural articial systems. 2nd edition, MIT Press, 1992.

[6] M. H. Lim, Y. Yu and S. Omatu, Ecient Genetic Algorithms using Simple Genes Exchange Local Search Policy for the Quadratic Assignment Problem, Computa-tional Optimization and Applications, 15(3)249-268, March 2000.

[7] Y. S. Ong and A. J. Keane, Meta-Lamarckian in Memetic Algorithm, IEEE Trans.Evolutionary Computation, 8(2)99-110, 2004.

[8] Y. S. Ong, M. H. Lim, N. Zhu and K. W. Wong, Classification of Adaptive Memetic Algorithms: A Comparative Study, IEEE Transactions On Systems, Man and Cy- bernetics - Part B, 36(1)141-52, 2006.

[9] Z. X. Zhu, Y. S. Ong, M. Dash, Markov Blanket-Embedded Genetic Algorithm for Gene Selection. Pattern Recognition, 2007.

[10] James Kennedy , Russell Eberhart, "Particle Swarm Optimization", in proc. 1995 Neural Networks Conf., 1995, PP.1942 – 1948 vol. 4.

[11] Xinxin Hu, Lijin Wang, YiwenZhong, "An Improved particle Swarm Optimization Algorithm for Site Index Curve Model", in proc. Granular Computing, 2008. Grc 2008. IEEE International Conf., 2008, pp. 486 – 490.

[12] Q. C. Nguyen, Y. S. Ong, and H. M. Lim, "A probabilistic memetic framework," IEEE Trans. Evol. Comput., vol. 13, no. 3, pp. 604–623, 2009.

[13] H. P. Schwefel, "Evolution and Optimum Seeking", John Wiley&Sons, 1995.