Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

PRIMAL-DUAL GENETIC ALGORITHMS FOR ROYAL ROAD FUNCTIONS

Shengxiang Yang [*]

[*] *Department of Mathematics and Computer Science*
University of Leicester
University Road, Leicester, LE1 7RH, UK
s.yang@mcs.le.ac.uk *http://www.mcs.le.ac.uk/~syang*

Abstract: Based on Holland's simple genetic algorithm (SGA) there have been many variations developed. Inspired by the phenomenon of diploid genotype and dominance mechanisms broadly existing in nature, we have proposed a primal-dual genetic algorithm (PDGA), see (Yang 2002). Our preliminary experiments based on the Royal Road functions have shown that PDGA outperforms SGA for different performance measures. In this paper we present some further experiment results, especially on the dynamic performance of PDGA over SGA, and give out our explanations and analyses about why PDGA outperforms SGA based on these results. Through the primal-dual mapping between a pair of chromosomes, PDGA's performance of exploration in the search space, especially during the early generations, is improved and thus its total searching efficiency is improved. *Copyright©IFAC 2002*

Keywords: Genetic algorithm, crossover, dominant, search, parallelism, optimization

1. INTRODUCTION

Ever since Holland first proposed the genetic algorithm, usually called Holland's simple genetic algorithm (SGA) (Holland 1975), there have been many variations and extensions both in GA's macro-structure such as the hybrid GAs, master-slave GAs and parallel GAs, and in GA's microstructure, including chromosome representation schemes, replace strategies, selection methods, mating policies, crossover operators, and mutation operators, see (Goldberg 1989, Michalewicz 1996, Mitchell 1996). Most GAs studied so far are haploidy-based, i.e., they operates on a set of single-stranded chromosomes. Haploid genotype is the simplest found in nature. GA's researchers have also studied diploid genotypes (pairs of chromosomes) and dominance mechanisms for a long history (Bagley 1967, Brindle 1981, Goldberg and Smith 1987, Ng and Wong 1995).

Inspired by the phenomenon of diploid genotype and dominance mechanisms broadly existing in nature, we have proposed a primal-dual genetic algorithm (PDGA in brief), see (Yang 2002). PDGA operates on a pair of chromosomes that are primal-dual to each other in the sense of Hamming distance in genotype. We have carried out some primary experiment results comparing the performance of PDGA over SGA based on the Royal Road functions (Forrest and Mitchell 1992, Mitchell *et al.* 1992, Mitchell 1996), which are tailor-made for testing GA's performance. The experiment results have shown that PDGA outperforms SGA on the Royal Road functions for several different performance measures.

In this paper we present some further experiment results and give out our explanations and analyses about why PDGA outperforms SGA based on these results. Through the primal-dual or Hamming distance mapping between a pair of chromosomes, PDGA's performance of exploration in the search space is improved, especially during the early generations of GA's searching progress and thus its total searching efficiency is improved.

2. PRIMAL-DUAL GENETIC ALGORITHMS

2.1 *Simple Genetic Algorithms*

The simple genetic algorithm, as one kind of generation-based evolutionary algorithm, maintains a population of candidate solutions or haploid chromosomes to a given problem which are evaluated according to a problem-specific fitness function that defines the environment for the evolution. New population is created by selecting relatively fit members of the present population and recombining them through crossover and mutation operations.

As a meta-heuristic optimization algorithm, an important issue on GA is the balance between exploration (the investigation for new, useful adaptations in the search space) and exploitation (the use and propagation of these adaptations). An efficient algorithm should use both techniques. PDGA was proposed with the aim of improving GA's exploration capacity in the search space.

2.2 *Primal-Dual Genetic Algorithms*

Here we only consider binary bit string representation of genotype and define a pair of chromosomes to be *primal-dual* to each other if their Hamming distance (the number of locations at which corresponding bits differ) is the maximum, i.e., equal to their length. For example, given a chromosome $x = (x_1, x_2, ..., x_L)$ of fixed length L, its dual chromosome is defined as $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_L)$ where $\bar{x}_i = 1 - x_i$ $(i = 1..L)$. Given this definition we say that x is mapped to $\bar{x}$ by the primal-dual mapping or Hamming distance mapping, vice versa.

PDGA is quite simple relative to other genetic algorithm variants, as shown in its pseudocode in Figure 1. From Figure 1 we can see that PDGA differs from SGA only in the evaluation of chromosomes in the population and the genetic operations are all the same for PDGA and SGA.

2.3 *Haploid over Diploid*

In the general diploid form a genotype carries one pair of homologous chromosomes. Each position in the genome has two or multiple allele values. A dominance mechanism for determining which allele value for a gene will be expressed is required to adjudicate when the allele values do not agree (Goldberg 1989).

```
  Procedure PDGA:

$t := 0$;
initializePopulation($P(0)$);
for each $x$ in $P(0)$ do    {evaluate $P(0)$}
  evaluate $x$ and $\bar{x}$;
  if $f(\bar{x}) > f(x)$ then $x := \bar{x}$;    {replace}
endfor
repeat
  $P' :=$ selectForReproduction($P(t)$);
  recombine($P'$);
  mutate($P'$);
  for each $x$ in $P'$ do    {evaluate $P'$}
    evaluate $x$ and $\bar{x}$;
    if $f(\bar{x}) > f(x)$ then $x := \bar{x}$;  {replace}
  endfor
  $P(t+1) :=$ selectForSurvival($P(t), P'$);
  $t := t + 1$;
until terminated = true;  {e.g., $t > t_{max}$}
```

Fig. 1. Pseudocode for PDGA.

From the above description of PDGA, it seems that it is diploidy-based, however, PDGA is in fact a haploid-like genetic algorithm since we needn't explicitly keep track of each chromosome's dual chromosome. That is, the dual chromosome $\bar{x}$ of a primal chromosome x can be looked as the shadow of x and only shows its body through the primal-dual mapping when x is evaluated. Of course, if $\bar{x}$ proves to be better, it will embody itself and throw x into its shadow. In this sense, we can call PDGA *pseudo-diploid* or *implicitly diploid.*

PDGA differs from those diploidy and dominance based genetic algorithms (Bagley 1967, Brindle 1981, Goldberg and Smith 1987, Ng and Wong 1995) in the three aspects. First, dominance in PDGA is phenotype oriented (i.e., the dominance mechanism works on the primal-dual pair by taking whichever has higher fitness as the dominant chromosome) instead of gene oriented and thus need no special dominant scheme as in those GAs. Secondly, dominance in PDGA operates on the pair of chromosomes that are primal-dual to each other instead of randomly chosen chromosomes. Finally, in PDGA only the dominant chromosome of the primal-dual pair instead of the pair of chromosomes undergoes genetic operations.

3. SCHEMA THEOREM

The building block hypothesis and schema theorem of Holland are the theoretical foundations of GA (Holland 1975, Goldberg 1989). Holland first proposed the notation of schema to describe a subset of all binary vectors of fixed length that have similarities at certain positions. A schema is typically specified by a vector over the alphabet

{0, 1, *}, where the "*" denotes a "wildcard" matching both 0 and 1. Given a schema S, its order $o(S)$ is the number of fixed positions within S and its defining length $l(S)$ is the maximum distance between fixed positions within S. For example, given $S = 01{*}{*}1{*}$, $o(S) = 3$ and $l(S) = 4$.

The building block hypothesis states that crossover combines short, low-order, high-fit schemas into increasingly fit, higher-order schemas. The schema theorem states that short, low-order, better than average schemas (also called building blocks) receive an exponentially increasing number of trials in the subsequent generations. Given a population size of N individuals, the number of schemas processed at each generation is of the order N^3, this is called GA's *implicit parallelism.*

However, the schema theorem and building block hypothesis do not state how crossover, one major source of GA's search power, works to recombine highly fit schemas from short, low-order ones.

4. ROYAL ROAD FUNCTIONS

To investigate schema processing and recombination in detail, Mitchell, Forrest and Holland designed a class of fitness landscapes, called Royal Road functions (Figure 2 and Figure 3), to capture the essence of building blocks (Mitchell *et al.* 1992, Forrest and Mitchell 1993, Mitchell 1996).

```
S1   = 11111111********************************************************
S2   = ********11111111************************************************
S3   = ****************11111111****************************************
S4   = ************************11111111********************************
S5   = ********************************11111111************************
S6   = ****************************************11111111****************
S7   = ************************************************11111111********
S8   = ********************************************************11111111
Sopt = 1111111111111111111111111111111111111111111111111111111111111111
```

Fig. 2. Royal Road Function R_1.

```
S1   = 11111111********************************************************
S2   = ********11111111************************************************
S3   = ****************11111111****************************************
S4   = ************************11111111********************************
S5   = ********************************11111111************************
S6   = ****************************************11111111****************
S7   = ************************************************11111111********
S8   = ********************************************************11111111
S9   = 1111111111111111************************************************
S10  = ****************1111111111111111********************************
S11  = ********************************1111111111111111****************
S12  = ************************************************1111111111111111
S13  = 11111111111111111111111111111111********************************
S14  = ********************************11111111111111111111111111111111
Sopt = 1111111111111111111111111111111111111111111111111111111111111111
```

Fig. 3. Royal Road Function R_2.

Royal Road functions R_1 and R_2 contain tailor-made building blocks and thus are good test problems to investigate GA's performance with respect to schema processing and recombination. They are defined using a list of schemas. Each schema s_i is given a coefficient c_i which is equal to its order (i.e., $c_i = o(s_i)$). For R_1, $c_i = 8$ for all s_i ($i = 1..8$) while for R_2, $c_i = 8$ for s_i ($i = 1..8$), $c_i = 8$ for s_i ($i = 9..12$), and $c_i = 8$ for s_i ($i = 13, 14$). The fitness of a bit string x for both $R_1(x)$ and $R_2(x)$ is computed by summing the coefficients c_i corresponding to each of the given schema s_i of which x is an instance. That is, $R_1(x)$ and $R_2(x)$ are defined as follows:

$$R_1(x) = \sum_{i=1}^{i=8} c_i \delta_i(x) \text{ and } R_2(x) = \sum_{i=1}^{i=14} c_i \delta_i(x)$$

where $\delta_i(x) = \{1, \text{if } x \in s_i; 0, \text{otherwise}\}$. If x is an instance of exactly two of the order-8 schema of R_1, $R_1(x) = 16$, e.g., $R_1(111111110..011111111) = 16$. Similarly, the optimal solutions for R_1 and R_2 are given as follows: $R_1(s_{opt}) = R_1(111..1) = 64$ and $R_2(s_{opt}) = R_2(111..1) = 192$.

For R_1 and R_2, with PDGA we can evaluate a primal-dual chromosome pair at the same time with only a few extra instructions than when we evaluate one single chromosome with SGA. For this reason we count the evaluations of a primal-dual pair as one evaluation in our experiments.

5. COMPUTER EXPERIMENT STUDY

5.1 *Experiment Design and Performance Measures*

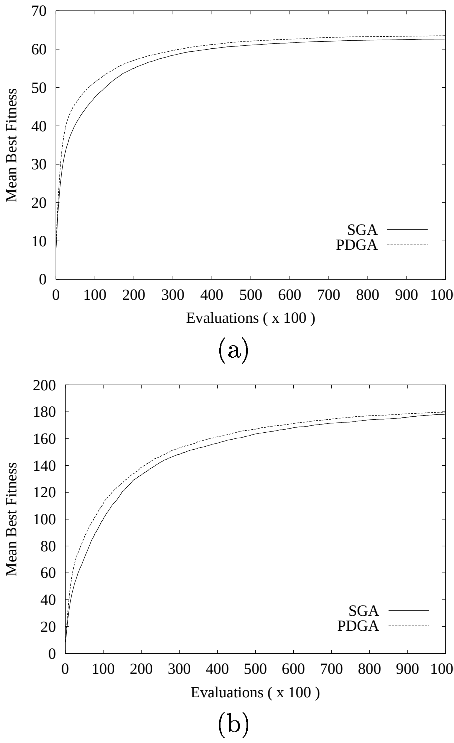
In our primary experiments in (Yang 2002), we have shown that PDGA outperforms SGA on the following performance measures: function evaluations to optimum, percentage of achieving optimum over a number of runs against generations, percentage of optimal members in the population for a run against generations.

According to our primary experiments, for our following experiments, we fix the parameters with population size $N = 128$, crossover probability $P_c = 0.7$, and mutation probability $P_m = 0.01$. We also use sigma truncation scaling and elitist schemes, one-point crossover and bit mutation for both PDGA and SGA. In this paper, we'll test the dynamic performance of PDGA and SGA with respect to the efficiency of function evaluations and the processing of schema recombination.

5.2 *Experiments on Efficiency of Evaluation*

To compare the dynamic performance of efficiency of function evaluation, we carried out 1000 runs of PDGA over SGA on R_1 and R_2 respectively with the same 1000 random seeds for each combination of algorithm and function. We record the best fitness found every 100 function evaluations for each run. Here only those chromosomes that were changed by crossover and mutation operations were evaluated and counted. The statistic results over 1000 runs are shown in Figure 4. From Figure

4 we can see that PDGA overruns SGA quite well, especially during the early generations.



(a)

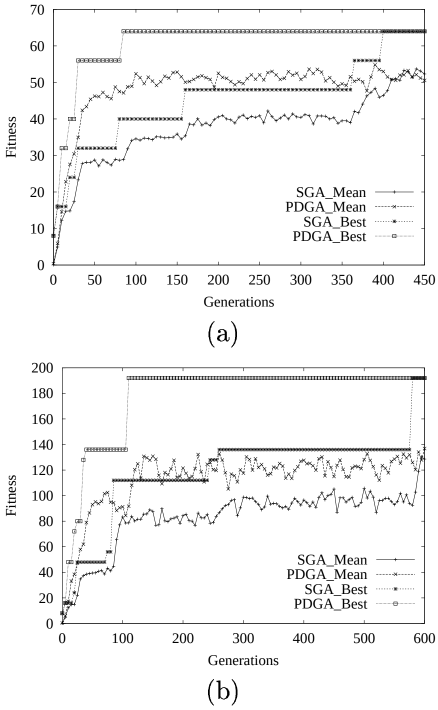
(b)

Fig. 4. Comparisons of PDGA vs. SGA with respect to mean best fitness over 1000 runs against evaluations on (a) R_1 and (b) R_2.

5.3 *Experiments on Schema Recombination*

To further compare the dynamic searching process of PDGA and SGA, we give out the result of a typical run of PDGA and SGA on R_1 and R_2 in Figure 5 with respect to mean fitness and best fitness achieved against generations. The data are plotted every 5 generations. On R_1, PDGA achieves optimum at generation 85 with 9252 evaluations while SGA at generation 400 with 43264 evaluations. On R_2, PDGA achieves optimum at generation 108 with 11784 evaluations while SGA at generation 578 with 62360 evaluations. From Figure 5 we can see that PDGA overruns SGA quite well especially during the early generations.

To study the dynamic process of schema recombination, we also recorded the percentage of individuals in the population that are instances of each schema every 5 generations. Figure 6 shows the corresponding schema recombination process of the same typical run on R_1 as in Figure 5(a). The result on R_2 is similar but not shown here. From Figure 6 we can see that PDGA obtained schemas much faster than SGA, especially on schema 4 and schema 6. SGA suffered much more heavily than PDGA by the phenomenon called “hitchhiking” (Schaffer *et al.* 1990, Mitchell 1996). For example, with SGA schema s_6 appeared at about generation 70 and 350 but suppressed by s_5 and s_7. Hitchhiking results from undesirable schemas being coupled or hitchhiked along with desirable schemas during recombination, thereby producing above average individuals and later getting sampled at a higher rate during selection. Hitchhiking seriously limits GA’s implicit parallelism and causes premature convergence.



(a)

(b)

Fig. 5. Mean and best fitness against generations for a typical run of PDGA vs. SGA on (a) R_1 and (b) R_2.

6. DISCUSSIONS AND ANALYSES

From above experiments we can see that PDGA outperforms SGA, especially during the early stage of GA’s searching process. In this section, we give out our explanations and analyses to this result. We have said that PDGA is proposed with the aim of improving GA’s exploration capacity in the search space through the primal-dual mapping. Here the mapping function has the key role in PDGA’s performance. We illustrates the effects of the primal-dual mapping in Figure 7, where the attribute(s) axis represents the combination of different attributes (e.g., schemas for Royal Road functions) of a chromosome.

Figure 7(a) illustrates an original population during the early generation of GA’s searching progress, where there are many individuals with low fitness. With SGA the genetic operations perform directly on the population without the primal-dual mapping, as shown in Figure 7(b); while with PDGA before genetic operations the primal-dual mapping maps those chromosomes

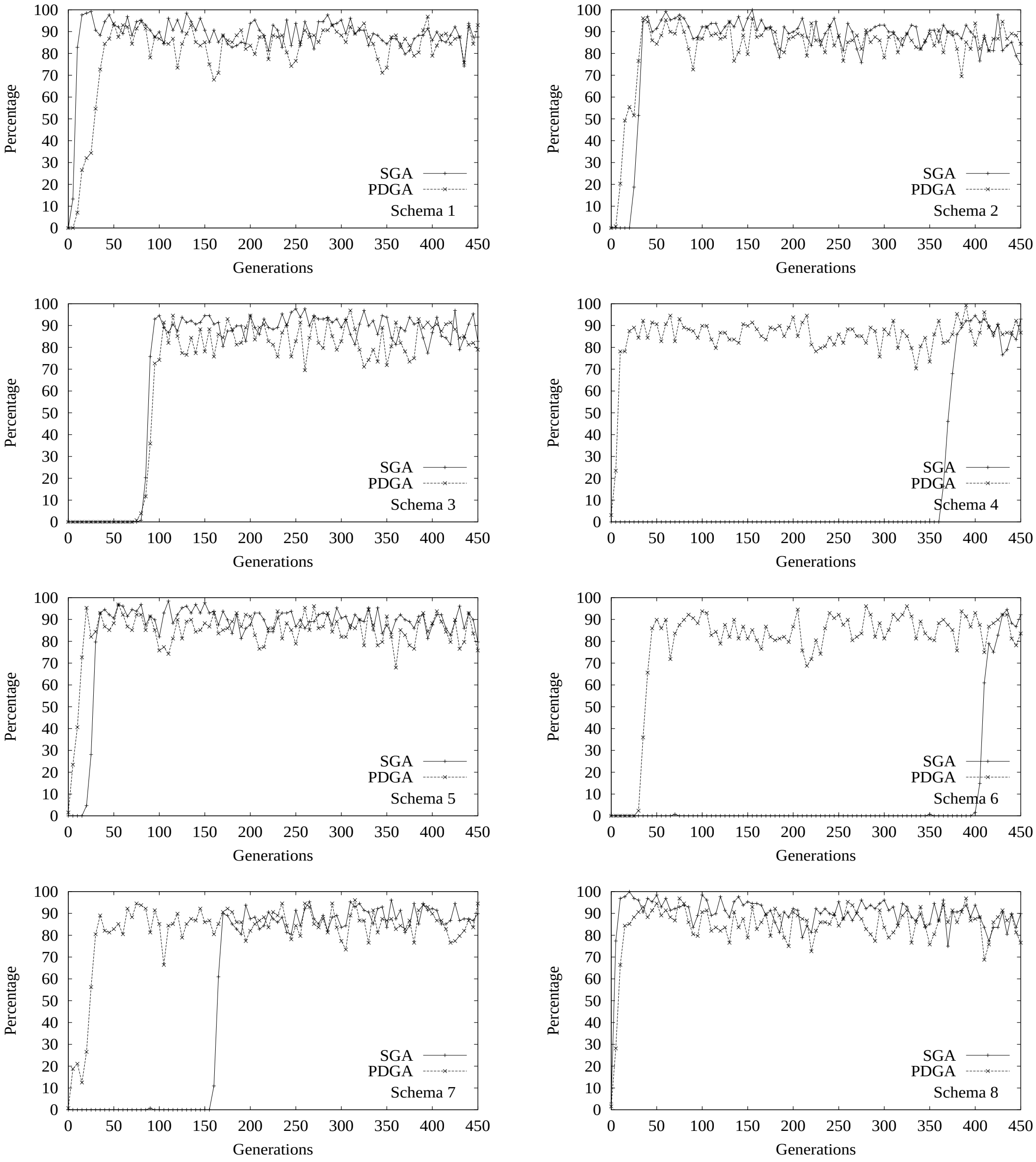


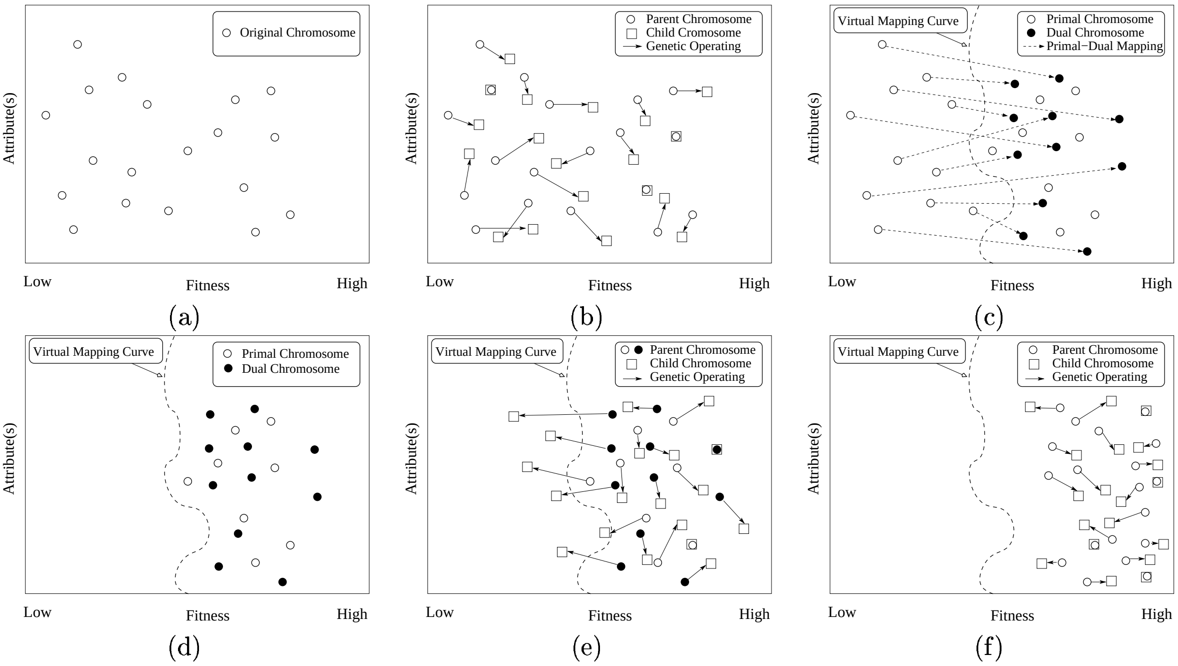
Fig. 6. Percentage of individuals in the population that are instances of the given schema against generations of a typical run of PDGA vs. SGA on R_1.

with fitness left to the virtual mapping curve, i.e., low fitness, into their dual ones that have high fitnesses and are right to the virtual mapping curve (see Figure 7(c)) and the genetic operations now perform on the mapped better population (see Figure 7(d)), as shown in Figure 7(e). From Figure 7 (a) to Figure 7(e) we can see that the primal-dual mapping greatly improves GA's performance during the early generations. However, after certain generations the primal-dual mapping has little effect because the mean fitness of the population has become quite high and new chromosomes created by the genetic operations seldom fall into the left side of the virtual mapping curve, as shown in Figure 7(f).

The primal-dual mapping also helps moderating the hitchhiking phenomenon for it improves the diversity of the population during the early stage of GA's searching progress.

7. CONCLUSIONS

Inspired by the phenomenon of diploid genotype and dominance mechanisms broadly existing in nature, in (Yang 2002) we have proposed a primal-dual genetic algorithm and have carried out some primary experiments showing that PDGA outperforms SGA on the Royal Road functions for different performance measures. In this paper we present some further experiment results comparing the dynamic performance of PDGA over SGA, and give out our explanations and analyses about why PDGA outperforms SGA.



(a) (b) (c)

(d) (e) (f)

Fig. 7. Illustration of the effects of the primal-dual mapping. (a) original population; (b) genetic operating without primal-dual mapping; (c) for population in (a), mapping those chromosomes with fitness left to the virtual mapping curve into their dual ones; (d) after the mapping; (e) genetic operating on the mapped population; (f) after certain generations the primal-dual mapping loses its effect.

Through the primal-dual mapping, PDGA's performance of exploration in the search space is improved, especially during the early generations of GA's searching progress and thus its total searching efficiency is improved. This mapping function works well during the early generations by shortening genetic operations performed on low fitness chromosomes and thus speed up GA's convergence. However, whence the mean fitness of the population becomes quite high, it loses its effect. For the future research on PDGA, we believe that dynamic primal-dual mapping function instead of the static Hamming mapping that can adapt itself with GA's searching progress will further improve PDGA's performance.

8. REFERENCES

Bagley, J. D. (1967). *The Behaviour of Adaptive Systems Witch Employ Genetic and Correlation Algorithms.* Doctoral Dissertation, University of Michigan.

Brindle, A. (1981). *Genetic Algorithms for Function Optimization.* Doctoral Dissertation, University of Alberta. Edmonton, Canada.

Forrest, S. and M. Mitchell (1992). Relative building-block fitness and the building-block hypothesis. In: *Foundations of Genetic Algorithms 2.* pp. 109–126. Morgan Kaufmann.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley. Reading, MA.

Goldberg, D. E. and R. E. Smith (1987). Nonstationary function optimization using genetic algorithms with dominance and diploidy. In: *Proc. of the 2nd Int. Conf. on Genetic Algorithms.* pp. 59–68.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press. Ann Arbor.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs.* 3rd ed.. Springer-Verlag. New York.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* MIT Press. Cambridge, MA.

Mitchell, M., S. Forrest and J. H. Holland (1992). The royal road for genetic algorithms: fitness landscapes and GA performance. In: *Proc. of the 1st European Conf. on Artificial Life.* pp. 245–254. MIT Press. Cambridge, MA.

Ng, K. P. and K. C. Wong (1995). A new diploid scheme and dominance change mechanism for non-stationary function optimization. In: *Proc. of the 6th Int. Conf. on Genetic Algorithms.* pp. 159–166. Morgan Kaufmann.

Schaffer, D. J., L. J. Eshelman and D. Offnut (1990). Spurious correlations and premature convergence in genetic algorithms. In: *Foundations of Genetic Algorithms 1* (G. Rawlins, Ed.). pp. 102–112. Morgan Kaufmann.

Yang, S. (2002). Genetic algorithms based on primal-dual chromosomes for royal road functions. In: *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation* (A. Grmela and N. E. Mastorakis, Eds.). pp. 174–179. WSEAS Press.