Vol. 41 No. 10 IPSJ Journal Oct. 2000

Regular Paper

Empirical Model with Cooperative-Competitive Genetic Operators to Improve GAs: Performance Investigation with 0/1 Multiple Knapsack Problems

Hernán E. Aguirre,† Kiyoshi Tanaka† and Tatsuo Sugimura†

This work explores an empirical model that puts genetic operators in a cooperative-competitive stand with each other. Two parallel operators produce offspring and fulfill specific roles: *Self-Reproduction with Mutation* (SRM) as a permanent source of diversity to induce the appearance of beneficial mutations and *Crossover and Mutation* (CM) to propagate them in the population. An extinctive selection mechanism subjects CM's and SRM's offspring to compete for survival and to guarantee the preservation of beneficial mutations for the next generation. SRM is implemented with an adaptive mutation schedule, which acts depending on SRM's contribution to the actual population, and varies mutation rates from high to small values to keep control of the exploration-exploitation balance. Two adaptive mutation schemes are investigated for SRM, adaptive dynamic segment (ADS) and adaptive dynamic probability (ADP). Mutation in CM is applied with a constant small probability. Thus, the expected cooperation between CM and SRM emerges resulting in higher search velocity and higher search reliability. The proposed model is investigated with the 0/1 multiple knapsack NP-hard combinatorial optimization problem where it outperforms a canonical genetic algorithm as well as other enhanced GAs.

1. Introduction

Holland[1)] defines crossover as the main genetic operator. Its role is to construct high order building blocks (hyperplanes) from low order ones. Mutation, on the other hand, is considered a "background" operator. Its primary role is to replace allele values lost from the population assuring that crossover has a full range of alleles so that the adaptive plan is not trapped on local optima. In canonical genetic algorithms these operators are applied one after the other; crossover yielding recombination of alleles via exchange of segments between pairs of chromosomes and then mutation inverting bits with a low probability rate p_m per bit.

Recent works, however, have shown that mutation can play a more significant role in genetic algorithms. Hinterding, et al.[2)] investigate the role of mutation as an independent reproduction operator (a new chromosome is produced either by crossover or mutation but not both). They conduct various experiments to determine the mixture of operators' rates that produces the best results within a number of evaluations. Best results were often obtained with quite low rates of crossover and hence high rates of mutation in the functions where their algorithm was applied. Also, applying mutation after crossover, similar to canonical genetic algorithms, but varying mutation rates using either time dependent functions or self-adaptive schemes for mutation rate control have proven to be effective approaches to improve the performance of genetic algorithms[3)~5)].

Moreover, theoretical as well as empirical studies of recombination and mutation in EAs in general[6)] and crossover and mutation in GAs in particular[7),8)] have given more insights to better characterize the roles of these operators. Spears[7)], theoretically, and Wu, et al.[8)], empirically, demonstrate that there are some important characteristics of each operator that are not captured by the other. Some common conclusions derived from these works are: (i) crossover is more powerful than mutation to propagate as well as to recombine building blocks (construction or exploitation), (ii) mutation introduces diversity in the population and is more powerful than crossover in terms of destroying building blocks (disruption or exploration), and (iii) the genetic search can take advantage of crossover's recombination ability only when the population is diverse and contain appropriate building blocks. Other very important points mentioned in Ref. 7) are: (iv) it is possible to control the amount of explo-

---

† Faculty of Engineering, Shinshu University

ration that mutation performs by adjusting the mutation rate, and (v) crossover can simultaneously achieve higher levels of construction and survival than any particular amount of mutation; for mutation, on the other hand, it seems difficult to simultaneously achieve high levels of construction and survival.

The balance between crossover, mutation, and selection has a long history and is a very important issue in genetic algorithms. The way this balance is sought will always rely on the understanding of how the genetic operators work. Although the studies to which we refer above were mainly aimed to clarify and show the relevance of the individual contributions of genetic operators, they also suggest forms in which appropriate balances could be better achieved leading to improve GAs.

From this point of view, this work explores an empirical model of GA that pursues better balances over the course of a run 9),10). The model uses two parallel operators to create offspring and assign them specific roles: *Self-Reproduction with Mutation* (SRM) as a permanent source of diversity to favor the appearance of beneficial mutations, and *Crossover and Mutation* (CM) to promote the increase of beneficial mutations in the population. An extinctive selection mechanism keeps the best individuals among them to guarantee the preservation of beneficial mutations for the next generation. The parallel formulation of genetic operators tied to extinctive selection creates a cooperative-competitive environment for CM and SRM. SRM is provided with an adaptive mutation schedule that varies mutation rates from high to small values depending on SRM's own contribution to the population (mutants survival ratio) keeping control of the exploration-exploitation balance and maintaining competition between the two operators' offspring during the entire course of evolution. The proposed method is applied to a NP-hard combinatorial optimization problem, the 0/1 multiple knapsack problem 11), where it outperforms not only a canonical genetic algorithm but also other enhanced genetic algorithms 5),12).

2. Proposed Model

2.1 Beneficial Mutations

Crucial factors that facilitate the existence of beneficial mutations in the population that will contribute to the progress of evolution are (i) rate at which favorable mutations occur, which is directly related to mutation rate and population size, (ii) propagation of beneficial mutations, sustained by crossover and (iii) how favorable mutations are, related to the intensity of selection (addressed in Section 2.3). It should be considered that these factors are not independent but strongly related to each other.

An augment in the population size will provide with more individuals to mutate increasing the possibility for favorable mutations to appear. However, smaller populations are desirable for many practical situations. In such cases, an increase in the mutation rate seems to be the only form to induce a higher rate for favorable mutations.

Once favorable mutations appear, its propagation to the rest of the population has to be considered in order to increase the number of beneficial mutations of each individual. To illustrate this point, as an example, let us suppose that two beneficial mutations, m' and m'', arise in the population. There are two ways in which the same organism might come to have both of them. (i) One of the mutations, m'', may originate in an individual that already has the other mutation, m'. (ii) Crossing over two organisms, one who has m' but not m'' and one who has m'' but not m'. In the first case, relaying only in mutation, beneficial mutations must increase one after the other, or serially. However, when crossover is used, they can increase at the same time in parallel. The crossover potential for parallel combination of beneficial mutations can help to increase their number in the population improving the fitness of the individuals.

This suggests that crossover combined with higher mutation rates may give improved performance in genetic algorithms.

2.2 Parallel Genetic Operators

One way in which we could try to combine crossover with higher mutation rates is to simply apply one operator after the other, as in a canonical genetic algorithm. Let us discuss the possible effects of crossover followed by mutation on favorable recombinations and beneficial mutations. First, we should consider that the crossover's benefits could be diminished according to the probability of mutation. If mutation probabilities are high, then although crossover alone could be doing a good job it is likely that some of the just created favorable recombinations may be lost because of the high disruption introduced by mutation. We could think of this case as a mutation interference with crossover's job. Another possibility is that mutation could be doing a good job but crossover may produce poor performing individuals (for instance, the recombination of individuals located on different peaks in multimodal landscapes can produce individuals located in the valleys between peaks where the fitness is low [6]). Thus the survivability of beneficial mutations could also be affected by ineffective crossing over operations. We can think of this case as a crossover interference with mutation's job. Other possibilities are when the effects of crossover and mutation aggregate either in a favorable or unfavorable manner or are neutral and don't interfere with each other.

Another way to combine crossover with higher mutation rates is to apply the operators in parallel. The parallel formulation of genetic operators can avoid those cases in which the operators interfere with each other implicitly increasing the levels of cooperation between operators to introduce and propagate beneficial mutations. It also sets the stage for competition between operators' offspring.

In order to take advantage of recombination and favor beneficial mutations simultaneously, we explore an empirical model of GA that applies two genetic operators in parallel to produce offspring, each one fulfilling a specific role. (i) *Self-Reproduction with Mutation*. Its role is to introduce diversity by means of mutation inducing the appearance of beneficial mutations. (ii) *Crossover and Mutation*. Its main role is to recombine information via exchange of segments between chromosomes promoting the increase of beneficial mutations in the population. We call these operators SRM and CM briefly from now on.

We expect SRM to be the major source of diversity in our model and its benefit depends upon the favorable mutations that it is able to create during the entire period of evolution. The search process goes through various stages and mutation should be sensible and supportive to all of them. At the beginning, emphasis on exploring the search space is required and mutation should seed its progeny in behalf of a global search. However, as the process goes on exploitation of profitable regions, without losing global search capabilities, are also needed. The exploration-exploitation balance over the course of a run is a crucial factor in the search process and it is incorporated in our model by providing SRM with a varying mutation scheme.

Since the introduction of diversity is expected from SRM, what we expect from CM is to work as a good replication-recombination operator. Thus, the mutation probability $p_m^{(CM)}$ related to CM, a replication error, is kept at a low rate. Note that serial small mutation after crossover in CM does not contradict the concept of our model (crossover parallel to high mutation) because the additional disruption caused by mutation gives little affection to the replication-recombination task of CM. In this way, the CM's strengths to propagate and recombine building blocks are not diminished by the high levels of disruption and exploration that SRM introduces during the various stages of the search. The main reason for applying small mutation after crossover is to keep the appropriate balance between CM and SRM. (This is verified with Fig. 6 and Fig. 7 in Section 4.4)

2.3 Extinctive Selection

As mentioned above, the parallel formulation of genetic operators tries to avoid interferences between operators. However, it does not prevent SRM from creating deleterious mutations or CM from producing ineffective crossing over operations. To cope with these cases we also incorporate in the model the concept of extinctive selection that has been widely used in Evolutionary Strategies [13]. Thus through extinctive selection the offspring created by CM and SRM coexist and compete for survival. **Figure 1** illustrates the expected effect of extinctive selec-

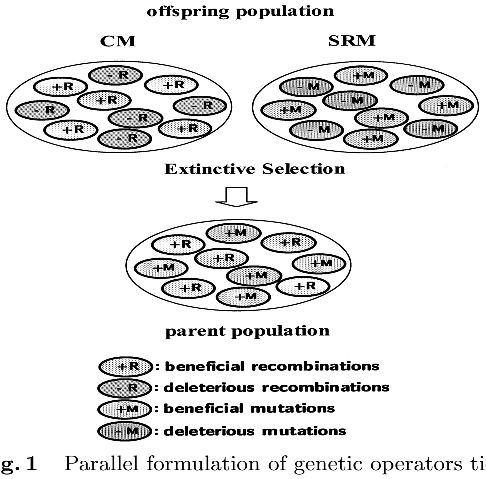


Fig. 1 Parallel formulation of genetic operators tied to extinctive selection.

tion: eliminate the poor performing individuals created by both CM and SRM.

Note that in our model the positive effect of crossover and mutation is never lost due to the interaction between CM's and SRM's offspring. That is, (i) we allow the best offspring of the two parallel operators to coexist in the same parent population through extinctive selection, and (ii) selection can choose individuals for reproduction based only on their fitness without any consideration about which operator created the individuals in the previous generation.

The parallel formulation of genetic operators tied to extinctive selection creates a cooperative-competitive environment for the offspring created by CM and SRM and gives the chances of interaction between them. Information about the relative success of the mutants is feedback to adapt mutation rates in SRM.

3. Improved Genetic Algorithm

3.1 Algorithm

The algorithm of the improved GA based on the proposed model is presented below and its block diagram is also shown in **Fig. 2**.

```
begin
 $t := 0$
 initialize ($P(0)$)
 evaluate ($P(0)$)
 while (not termination condition)
 begin
   $P'(t)$ = crossover and mutation ($P(t)$)
   $P''(t)$ = self-reproduction with mutation
     ($P(t)$)
 evaluate ($P'(t) \cup P''(t)$)
   $P(t+1) = (\mu, \lambda)$ proportional selection
     ($P'(t) \cup P''(t)$)
   $t := t + 1$
 end
end
```

CM is applied to individuals selected from the parent population $P(t)$ analogous to canonical GA as explained by Holland[1] and Goldberg[14], therefore the following sections describe only SRM and the selection mechanism.

3.2 SRM (Self-Reproduction with Mutation)

In order to produce offspring with SRM, individuals are selected from the parent population $P(t)$, an exact copy is created and then mutation is applied with a mutation probability $p_m^{(SRM)}$. SRM is provided with an adaptive mutation schedule that acts depending on SRM's own contribution to the population, keeping control of the exploration-exploitation balance, and maintaining competition between the two operators' offspring during the entire course of evolution. Two mutation schemes are investigated for SRM: (i) adaptive dynamic-segment (ADS), and (ii) adaptive dynamic-probability (ADP).

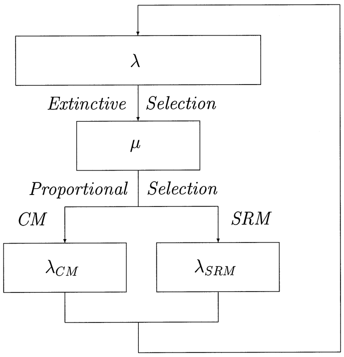


Fig. 2 Block diagram of proposed algorithm.

3.2.1 ADS (Adaptive Dynamic-Segment)

ADS directs mutation only to a segment of the chromosome using constant mutation probabilities per bit

$$p_m^{(SRM)} = \begin{cases} \alpha \, (\text{if the bit is in the segment}) \\ 0 \, (\text{otherwise}) \end{cases}$$

while the mutation segment size ℓ is dynamically adjusted every time a normalized mutants survival ratio, γ, falls under a threshold, τ. The normalized mutant survival ratio is specified by

$$\gamma = \frac{\mu_{SRM}}{\lambda_{SRM}} \cdot \frac{\lambda}{\mu} \tag{1}$$

where μ_{SRM} is the number of individuals created by SRM present in the parent population $P(t)$ after selection, λ_{SRM} is the offspring number created by SRM, λ is the total offspring number ($\lambda_{CM} + \lambda_{SRM}$), and μ is the number of individuals in $P(t)$. The segment reduction is summarized below:

```
$\ell = n \ (t = 0)$
if ($\gamma < \tau$) and ($\ell > 1/\alpha$)
  $\ell = \ell/2$
```

where the segment size ℓ varies from n (bit string length) to $1/\alpha$, $[n, 1/\alpha]$ following a step decreasing approach as shown in **Fig. 3**.

The segment initial position, for each chromosome, is chosen at random, $s_i = N[0, n)$, and its final position is calculated by

$$s_f = (s_i + \ell) \bmod n. \tag{2}$$

With this scheme, the average number of flipped bits goes down from $n\alpha$ to 1, $[n\alpha, 1]$.

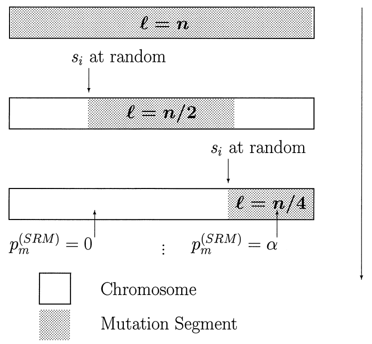


Fig. 3 ADS (adaptive dynamic-segment) mutation.

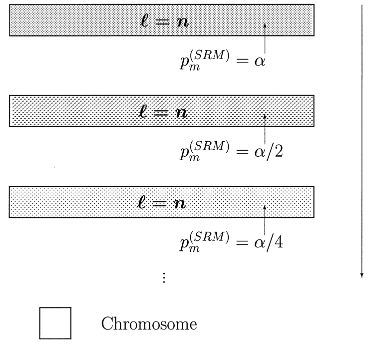


Fig. 4 ADP (adaptive dynamic-probability) mutation.

3.2.2 ADP (Adaptive Dynamic-Probability)

With ADP, every bit in the chromosome is always subject to mutation with probability $p_m^{(SRM)}$ varying each time γ falls under τ:

$p_m^{(SRM)} = \alpha \ (t = 0)$
if $(\gamma < \tau)$ **and** $(p_m^{(SRM)} > 1/n)$
  $p_m^{(SRM)} = p_m^{(SRM)}/2$

In other words, the segment size is kept constant, $\ell = n$, but $p_m^{(SRM)}$ follows a step decreasing approach from α to $1/n$ per bit, $p_m^{(SRM)} = [\alpha, 1/n]$ as shown in **Fig. 4**.

Both schemes, ADS and ADP, impose an adaptive mutation rate control with the same expected average number of flipped bits; the difference lies whether mutation is applied locally inside the segment (ADS) or globally at a chromosome level (ADP). Also, in the case of ADS two different probabilities per bit are used meanwhile in ADP a uniform mutation probability per bit is used.

3.3 (μ, λ) Proportional Selection

Among the various extinctive selection mechanism available in the EA literature[13] we chose (μ, λ) Proportional Selection[13],[15] to implement the required extinctive selection mechanism. We are interested in comparing the proposed algorithm with a canonical genetic algorithm and using this selection mechanism we can switch from the former to the latter by simply setting $\mu = \lambda$. Selection probabilities for this kind of selection are computed by

$$P_s(\boldsymbol{x}_i^{(t)}) = \begin{cases} \dfrac{f(\boldsymbol{x}_i^{(t)})}{\displaystyle\sum_{j=1}^{\mu} f(\boldsymbol{x}_j^{(t)})} & (1 \le i \le \mu) \\ 0 & (\mu < i \le \lambda) \end{cases} \qquad (3)$$

where $\boldsymbol{x}_i^{(t)}$ is an individual at generation t which has the i-th highest fitness value $f(\boldsymbol{x}_i^{(t)})$, μ is the number of parents and λ is the number of offspring. This kind of selection has been characterized as dynamic, extinctive, pure selection[13],[15].

4. Experimental Results and Discussion

4.1 The 0/1 Multiple Knapsack Problem

The 0/1 multiple knapsack problem consists of m knapsacks of capacities $c_1, c_2, \ldots, c_m$ and n objects. Each object has a profit p_i $(1 \le i \le n)$, weights w_{ij} $(1 \le j \le m)$, and it is either placed in all m knapsacks or in none at all. The 0/1 multiple knapsack problem can be formulated to maximize the function

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} p_i x_i \qquad (4)$$

subject to

$$\sum_{i=1}^{n} w_{ij} x_i \le c_j \qquad (j = 1, \ldots, m) \qquad (5)$$

where $x_i \in \{0, 1\}$ $(i = 1, \ldots, n)$ are elements of a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, which is the combination of objects we are interested in finding. A solution vector $\boldsymbol{x}$ should guarantee that no knapsack is overfilled and the best solution should yield the maximum profit. An $\boldsymbol{x}$ that overfills at least one of the knapsacks is considered as an infeasible solution.

4.2 Experimental Setup

We study our model and test its effectiveness using various 0/1 multiple knapsack problems[11] which from previous efforts[5],[12] seem to be fairly difficult for GAs to find global optimum solutions.

Experiments are conducted using the following algorithms: (i) a cGA (CM and propor-

Table 1 Genetic algorithms parameters.

Parameter	cGA	GA (μ,λ)	GA-SRM (μ,λ)
Representation	Binary	Binary	Binary
Selection	Proport.	(μ,λ) Proport.	(μ,λ) Proport.
Scaling	Linear	Linear	Linear
Mating	$(\boldsymbol{x}_i,\boldsymbol{x}_j),\ i\neq j$	$(\boldsymbol{x}_i,\boldsymbol{x}_j),\ i\neq j$	$(\boldsymbol{x}_i,\boldsymbol{x}_j),\ i\neq j$
Crossover	one point	one point	one point
p_c	0.6	0.6	1.0
$p_m^{(CM)}$	$1/n$	$1/n$	$1/n$
$p_m^{(SRM)}$	-	-	$\begin{cases}\alpha=0.5,\ \ell=[n,1/\alpha]\ (ADS)\\ \alpha=[0.5,1/n],\ \ell=n\ (ADP)\end{cases}$
$\mu:\lambda$	-	$1:2$	$1:2$
$\lambda_{CM}:\lambda_{SRM}$	-	-	$1:1$

tional selection), (ii) a GA (μ,λ) Proportional Selection (CM and extinctive proportional selection), and (iii) the proposed GA-SRM (μ,λ) Proportional Selection (CM, SRM and extinctive proportional selection). Unless stated otherwise, the genetic algorithms we use here are set with the parameters specified in **Table 1**.

Every experiment consisted of 100 independent runs. Each run was set with different seeds for the random initial population and ended after T evaluations were performed (the number of generations for each experiment is calculated as T/λ).

The objective function we use in our study introduces the same penalty term used in Ref. 12) to deal with infeasible solutions (no repair strategy is used). Thus the fitness function is specified by

$$f(\boldsymbol{x}) = h(g(\boldsymbol{x}) - s\cdot\max\{p_i\}) \qquad (6)$$

where $h(\cdot)$ denotes linear scaling 14) and s $(0\leq s\leq m)$ is the number of overfilled knapsacks.

4.3 Operators' Balance

First, we investigate the importance of SRM and the operators' balance for offspring creation. Three general cases are considered. (i) The parent population $P(t)$ has the same number of individuals as CM's and SRM's offspring populations, $\lambda_{SRM}=\mu=\lambda_{CM}$. (ii) $P(t)$ is smaller than CM's but bigger than SRM's population, $\lambda_{SRM}<\mu<\lambda_{CM}$. (iii) $P(t)$ is bigger than CM's but smaller than SRM's population, $\lambda_{SRM}>\mu>\lambda_{CM}$.

In the case of equal size populations, both operators could allocate all its offspring to $P(t)$. Therefore, there is competition between the two operators' offspring for every spot in $P(t)$. The normalized mutant survival ratio γ, specified by Eq. (1), reflects the number of mutants *winners* that survive after competing with CM's offspring. Also, in this case the number of mutants that survive selection equals the number of CM's offspring being eliminated.

However, if one of the offspring populations is smaller than $P(t)$, then it could at most cover a fraction of the parent population. Hence competition for survival between operators is not for the μ spots but rather for μ^c specified by the size of the smaller population since the best $\mu-\mu^c$ of the exceeding population need not to compete in order to survive. For example, if the bigger population corresponds to SRM, μ_{SRM} in Eq. (1) includes not only the mutants *winners* but also those that survive without competition. Also, the number of mutants that survive selection does not equal to the number of CM's offspring being eliminated.

To reflect the competition between operators when different offspring population sizes are used the mutants survival ratio of Eq. (1) is extended to

$$\gamma = \frac{\mu^w_{SRM}}{\lambda^c_{SRM}}\cdot\frac{\lambda^c}{\mu^c} \qquad (7)$$

where μ^w_{SRM} is the number of individuals created by SRM that compete and survive selection (mutants *winners*), λ^c_{SRM} is the offspring number created by SRM that undergoes competition, λ^c is the total offspring number that compete for survival $(\lambda^c_{CM}+\lambda^c_{SRM})$, and μ^c is the number of spots that SRM's and CM's offspring compete for in the parent population $P(t)$. Note that Eqs. (1) and (7) are the same if equal size populations are used (case (i)).

The balance between operators for offspring creation is studied using Eq. (7). We set $(\mu,\lambda)=(50,100)$, and conduct several experiments especially for *Weing 7*☆ in which our scheme varies from an all CM regime to a 90% SRM regime. A 100% CM regime in

---

☆ For this problem we set $T=2\times10^5$. The same number of evaluations were used in Refs. 5) and 12) using offspring populations of 100 and 50 individuals, respectively.

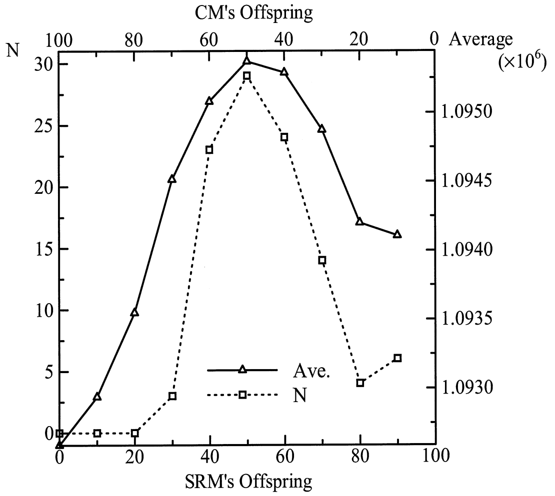


Fig. 5 Operators' balance and search ability.

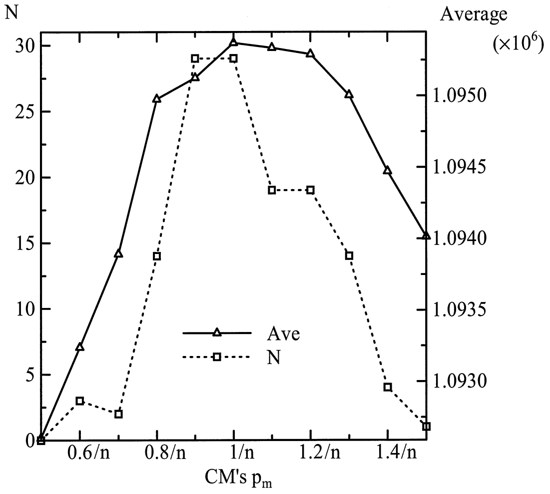


Fig. 6 CM's mutation probability and search ability.

our case represents a genetic algorithm that applies "background" mutation after crossover and uses (μ,λ) Proportional Selection, i.e., GA (50,100). Also, note that because SRM's adaptive mutation schedule is based on a mutant survival ratio, which reflects the competition among SRM's and CM's offspring, it is not possible to test the algorithm with an all SRM regime and simultaneously keep its adaptive feature.

The relationship between operators' offspring balance and search ability (best individuals' average and number of times the global optimum was found in 100 runs, *Average* and *N* respectively) is shown in **Fig. 5** when SRM is implemented with ADS. From this figure the following observations can be drawn. Ratios that favor SRM's offspring, i.e., $\lambda_{SRM} > 50\%$, produce better results than its opposites. A $\lambda_{CM} : \lambda_{SRM} = 1 : 1$ ratio is the best choice for stable and robust performance that simultaneously maximizes *N* and *Average*. In the following sections we use the best 1 : 1 operators' balance.

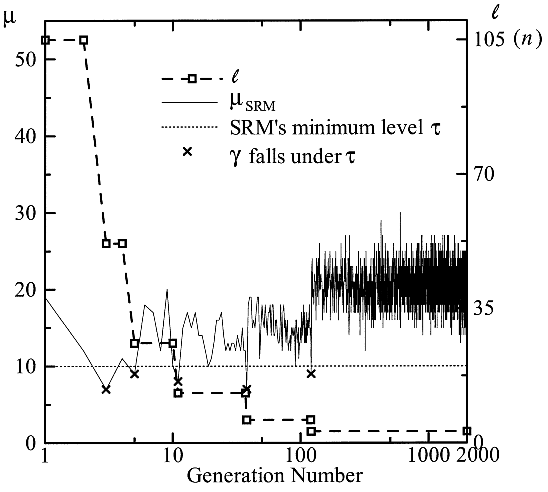
4.4 Mutation Probability in CM

Second, we fix $\lambda_{CM} : \lambda_{SRM} = 50 : 50$ and study the relevance of CM's mutation probability. The model's searching ability for $p_m^{(CM)}$ values in the range $[0.5/n, 1.5/n]$ are shown in **Fig. 6**. From this figure the following observations are relevant. A $p_m^{(CM)} = 1/n$ turns out to be the probability that gives us the highest values for *Average* and *N*, that is a coincidence with the results in Ref. 13). Values of $p_m^{(CM)} > 1/n$ are less deteriorative than values of $p_m^{(CM)} < 1/n$ are. Segment size reduction, ℓ, as well as the number of individuals produced by SRM that survive selection, μ_{SRM}, are shown for one of the runs for $p_m^{(CM)} = 1/n$ in **Fig. 7** (a). Here we can observe that SRM contributes to the survivor parent population in every generation of the search process. The key factor for SRM to be an effective operator lies in its own regulation mechanism, i.e., the mutation rate is adjusted every time the number of mutants that survive selection falls under a minimum level τ. Also, the average number of SRM's offspring that survive selection increases as the mutation segment is reduced.

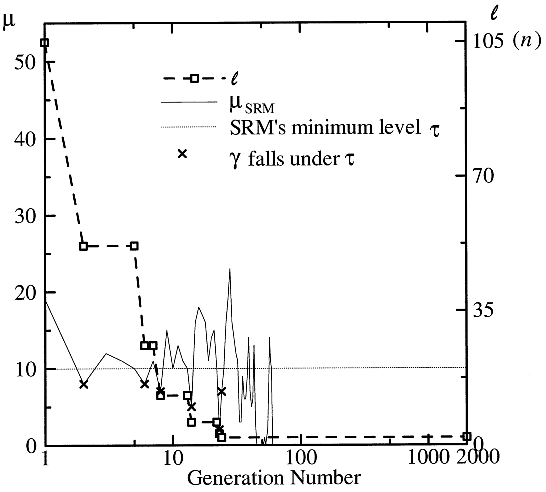
For $p_m^{(CM)} \leq 0.5/n$ we observe that SRM's offspring fitness cannot compete with CM's offspring, which is specially critical during the early stages of the search, causing a premature reduction of SRM's mutation rate, lost of diversity in the population and convergence to a local optimum. Another typical figure on SRM's contribution μ_{SRM} and segment size reduction ℓ for $p_m^{(CM)} = 0$ (without mutation after crossover) is shown in Fig. 7 (b) to compare with Fig. 7 (a). As mentioned in Section 2.2, small mutation after crossover is required in CM to achieve a robust search performance by keeping an appropriate balance between CM and SRM.

4.5 Extinctive Selection Pressure

Next, we set $\lambda_{CM} : \lambda_{SRM} = 50 : 50$, $p_m^{(CM)} = 1/n$, and vary μ to study the effect that extinctive selection has in our model. **Figure 8** shows results for $(\mu, \lambda) = (\{10, 20, \ldots, 90\}, 100)$.



(a) $p_m^{(CM)} = 1/n$.



(b) $p_m^{(CM)} = 0$.

Fig. 7 SRM's offspring number after extinctive selection.

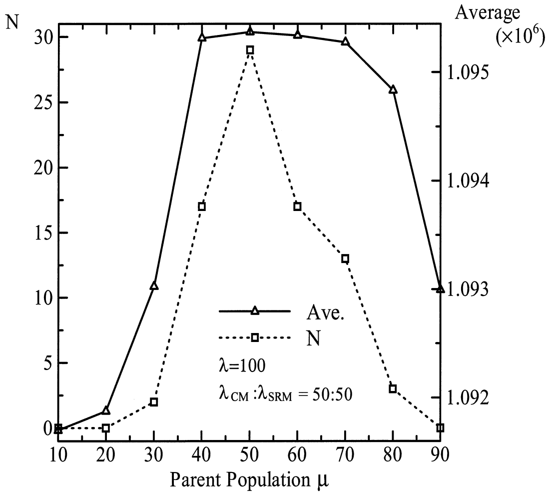


Fig. 8 Extinctive selection pressure and search ability.

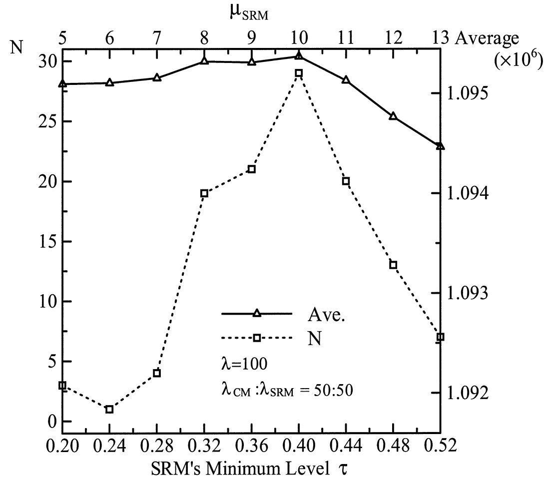


Fig. 9 SRM's minimum level and search ability.

High values of *Average* are attained for ratios of extinctive selection pressure in the range $\mu/\lambda = [40/100, 70/100]$. For $\mu < 50$ both CM and SRM produce offspring in excess of the parent population's requirement ($\lambda_{CM} > \mu$ and $\lambda_{SRM} > \mu$). In this case, there exists competition for survival even among CM's offspring, and SRM's offspring have to outperform CM's best offspring to survive. As we reduce the parent population size, competition conditions become severer.

On the other hand, for $\mu > 50$ neither CM alone nor SRM can cover the parent population's demand ($\lambda_{CM} < \mu$ and $\lambda_{SRM} < \mu$). In this situation, even if CM totally outperforms SRM, the latter has guaranteed at least $\mu - \lambda_{CM}$ of its best progeny for the next generation. However, with this scheme we do not facilitate the removal of CM's offspring that are performing poorly. Note that for the worst CM's offspring to be eliminated it has to be worse than the best $\mu - \lambda_{SRM}$ SRM's offspring.

4.6 SRM's Minimum Level τ

Figure 9 plots *Average* and N for values of τ in the range [0.20,0.52] for *Weing 7*. SRM's offspring that survive selection increases as mutation rates are reduced. Therefore, if τ is too small the mutation rate could remain too high at the end of the search, i.e., after certain point there are no further reductions in SRM's mutation rate because the mutants survival ratio γ is always higher than τ. In our experiments, we observe that the minimum mutation rates were

$3/n$ in more than 90% of the runs for $\tau \leq 0.28$, and $1.5/n$ in 92% of the runs for $\tau = 0.32$. In both cases *Average* is high but there is a big difference in *N*.

As we increase τ, the minimun value of the mutation rate will tend to be $1/n$ and its reduction will be faster. In this example, the average time (on the hundred runs) at which $1/n$ mutation rate is reached is about $0.5T$ for $\tau = 0.48$, and $0.25T$ for $\tau = 0.52$. A proper reduction speed of SRM's mutation rates guarantees a high *Average* and a SRM's mutation rate close to $1/n$ during the final stage of the search helps to locate the global optimum.

From Fig. 9, it can be observed that there is a broad range for the threshold τ in which the *Average* is very high. Also, that there is a safety-range in which both *Average* and *N* are high. Similar behavior is observed on other problems used to test the model.

4.7 Contribution of Parallel Genetic Operators and Extinctive Selection

Based on the optimum parameters obtained in Sections 4.3–4.6, in order to isolate the contributions of parallel genetic operators and higher selection pressure induced by extinctive selection we conduct several experiments using a cGA, GA (μ, λ), and GA-SRM (μ, λ). **Figure 10** plots the average objective fitness in 100 runs of the best-so-far individual over the generations by a cGA (100), GA (50,100), and GA-SRM (50,100). From this figure we can see that the higher selection pressure of extinctive selection causes an increase on search velocity. GA (μ,λ) in this problem also exhibits higher convergence reliability than cGA without extinctive selection; however, GA (μ,λ) is still not able to find the global optimum and the *Average* is lower compared to GA-SRM (μ,λ) (see Table 5 for *Weing 7*).

The only difference between GA (μ,λ) and GA-SRM (μ,λ) is the inclusion of adaptive mutation, SRM, in the latter. Therefore any difference in performance between these algorithms can be attributed to SRM. To better observe SRM's contribution we conduct experiments in which starting with a GA (μ,λ) configuration (all CM and extinctive selection) after a predetermined number of evaluations the algorithm switches to a GA-SRM (μ,λ) configuration (CM, SRM and extinctive selection). **Figure 11** plots results by an algorithm that makes the configuration transition from GA (50,100) to GA-SRM (50,100) at $\{0.02T, 0.05T, 0.10T, 0.20T, 0.5T\}$ evaluations, respectively. As a reference it also includes the results presented in Fig. 10 by GA (μ,λ) and GA-SRM (μ,λ). From Fig. 11 we can see that as soon as SRM is included fitness starts to pick up increasing the convergence reliability of the algorithm. Also, early transitions produce higher performance. For example, final results for the algorithms that perform transitions at $0.10T$ and $0.50T$ are $(N, Ave)=\{(22, 1095242.1), (10, 1094912.8)\}$, respectively.

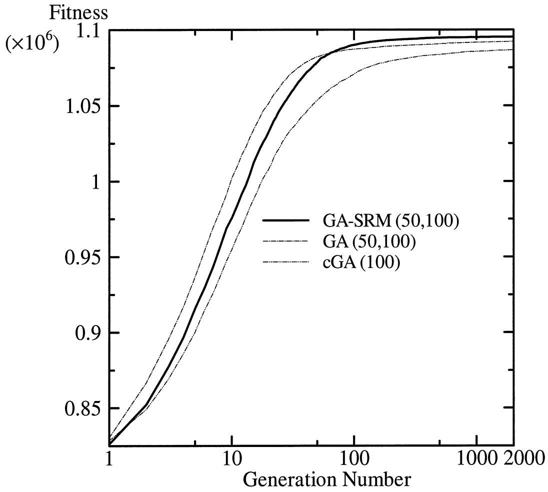


Fig. 10 Average fitness in 100 runs of the best-so-far individual.

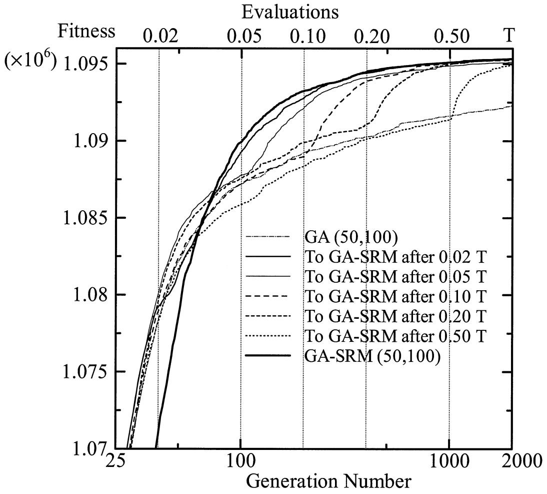


Fig. 11 Configuration transition: GA (50,100) to GA-SRM (50,100).

Summarizing Figs. 10 and 11, GA-SRM (μ,λ) gains its increase on search velocity from extinctive selection and its higher convergence reliability from the inclusion of parallel adaptive

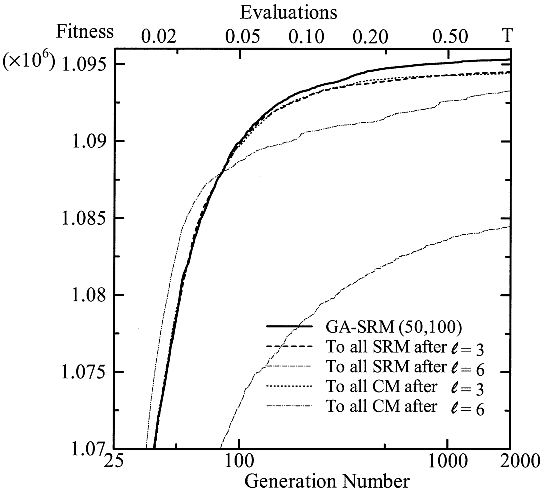


Fig. 12 Configuration transition: GA-SRM (50,100) to all SRM or all CM.

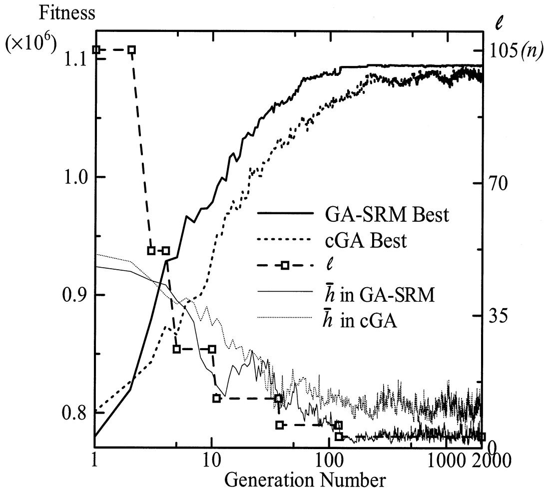


Fig. 13 Diversity and search ability.

mutation.

To further clarify the contribution of the interaction between CM and SRM during the latest stages of the search we also conduct experiments in which starting with a GA-SRM (μ,λ) configuration, after the mutation rate on SRM has reached a predetermined value, the algorithm switches either to a all CM regime with extinctive selection or to a all SRM regime with extinctive selection (in the latter case no further reductions on SRM's mutation rate are done). **Figure 12** plots results by an algorithm that makes the configuration transition from GA-SRM (50,100) when the mutation segment length in SRM has reached $\ell = \{6, 3\}$. As a reference it also includes the results presented in Fig. 10 by GA-SRM (μ,λ). From Fig. 12 we can see that neither CM nor SRM alone but the interaction of both CM and SRM leads to a higher convergence reliability.

To explain why the interaction of both CM and SRM works better than CM alone we look at diversity values and performance simultaneously. **Figure 13** presents the fitness value of the best individual in the population and the average hamming distance to the best individual $\bar{h}$ over the generations for one of the runs by cGA and GA-SRM. The SRM's mutation segment reduction ℓ is also presented for GA-SRM.

We can see that cGA ends up with values of $\bar{h}$ higher than GA-SRM after T evaluations. Also, at the end of the run the number of diverse individuals in the parent population is about 95% in cGA and 83% in GA-SRM. We let the cGA run for $4T$ and found that $\bar{h}$ and the number of diverse individuals remains at the same levels and that the quality of the solution does not increase significantly (see below at the end of Section 4.8). The higher levels of diversity observed in cGA after T evaluations and the lack of improvement in the quality of solutions after $4T$ evaluations seem rather contradictory. Our explanation for this comes from the highly multimodal nature of the landscape that the problem used in our simulations has and it is in accordance with the findings in Ref. 6). In Ref. 6) it is shown that the recombination of high fitness individuals located on multiple peaks can often produce poor performing individuals in the valley between peaks, with the side effect of increasing the values of diversity metrics such as $\bar{h}$ and the number of diverse individuals. On the other hand, the presence of multiple peaks will have less influence on the mutation of a high fitness individual on a particular peak (assuming small mutation probabilities). The values of $\bar{h}$ in Fig. 13 rather than being an indication of the cGA's ability to keep diversity show that CM alone has difficulties pulling the population to higher peaks. In the case of GA-SRM, the lost of effectiveness by CM during the final stages of the search is supplemented by the augment of SRM's contribution conform mutation rates are reduced as shown in Fig. 7 (a).

4.8 Search Ability and Evaluation Times

The global search ability under the influence of various evaluation times is observed by letting the algorithms run for $4T$ evaluations. **Ta-**

Table 2 Results for Weing 7 using GA-SRM (50,100) with ADS under various evaluation times.

Maximum	0.25 T	0.5 T	$T = 2 \times 10^5$	2 T	4 T
1095445	2	11	26	57	77
1095382	8	29	47	41	23
1095357	2	8	15	1	
1095352	1	2	2	1	
< 1095352	87	50	3		
Average	1094854.1	1095177.6	1095345.5	1095417.4	1095430.5
Stdev	602.8	545.3	337.2	41.6	27.63

ble 2 presents results by GA-SRM for some intermediate times, where *Stdev* denotes the value of standard deviation around *Average*. The previous figures and Table 2 empirically show the effect of the proposed cooperative-competitive model in terms of higher search velocity and higher search reliability (reach better solutions with small *Stdev* values). Note the *Average*, *N* and *Stdev* for $0.25T$ and $0.5T$. They also indicate that SRM is a continuous and effective source of diversity, which at the expense of time could be used to improve the search results. For example, when the algorithm was allowed to run for $2T$ evaluations for this particular problem, a remarkable improvement was achieved finding the global maximum 57% of the times with *Average* greater than the second known optimum and very small *Stdev* values. Results by cGA and GA (μ,λ) after $4T$ evaluations are (*N*, *Ave*, *Stdev*)={(0, 1087312.0, 1555.2), (0, 1093797.3, 1624.1)}, respectively.

4.9 Two-point and Uniform Crossover

We also conducted experiments using two point and uniform crossover. Results using two point crossover by cGA and GA (μ,λ) after T evaluations are (*N*, *Ave*, *Stdev*)={(0, 1086886.8, 1590.8), (0, 1092647.5, 3032.0)}, respectively. Similarly, results using uniform crossover are (*N*, *Ave*, *Stdev*)={(0, 1090392.1, 1020.7), (0, 1093748.4, 1776.9)}.

These results are better than those obtained with one point crossover, yet the global optimum solution could not be found. In the case of GA-SRM (μ,λ), however, the obtained results are quite similar to one point crossover.

4.10 Impact of the Population Size

The impact of the population size in the method's robustness is verified in **Table 3** (a) and (b) by GA-SRM using ADS and ADP, respectively. All population configurations use the same T evaluations. The right number indicates the value of the global (local) optimum and the left one the number of times it was found. At the bottom of each column, *Average* and *Stdev* are also presented. The model using only 40% of the population size still produces high values for *Average* and *N*. These results are encouraging and show that another important benefit of the cooperative-competitive model could be the reduction of the population size. Results by a larger population configuration, i.e., GA-SRM (100,200), are also included. Under the same evaluation time, we could not see considerable difference in the results by GA-SRM from (100,200) to (30,60) configurations for this particular problem.

4.11 Comparison

Finally, various results are additionally presented and compared in this section. In **Table 4** column *Problem* indicates the knapsack instance name *Name*, the number of objects n (it corresponds to the chromosome bit string length), the number of knapsacks m and the known global optimum value *Max*. Column *Parameters* shows the specific values set for τ (used only in GA-SRM), CM's mutation probability $p_m^{(CM)} \approx 1/n$, and number of evaluations T. **Table 5** shows results by cGA (population of 100 individuals), GA (50,100), and GA-SRM (50,100) using either adaptive dynamic segment *SRM-ADS* or adaptive dynamic probability *SRM-ADP*.

As a reference, **Table 6** presents results for the same problems reported by Khuri, et al.[12] running a genetic algorithm for the same T evaluations with a population of 50 individuals. **Table 7** shows the latest results by Bäck, et al.[5] particularly for *Weing 7* running a genetic algorithm with constant mutation rates and other enhanced genetic algorithms that apply varying mutations for the same $T = 2 \times 10^5$ evaluations using offspring populations of 100 individuals.

From Table 5 it can be seen that the proposed method outperforms cGA and GA (μ,λ) in every knapsack test problem where simulations

Table 3 Results for Weing 7 using GA-SRM with different population sizes.

(a) ADS.

GA-SRM (100,200) $p_m^{(CM)} = 0.01$ $\tau = 0.46$		GA-SRM (50,100) $p_m^{(CM)} = 0.01$ $\tau = 0.40$		GA-SRM (30,60) $p_m^{(CM)} = 0.01$ $\tau = 0.33$		GA-SRM (20,40) $p_m^{(CM)} = 0.01$ $\tau = 0.30$	
22	1095445	26	1095445	22	1095445	15	1095445
58	1095382	47	1095382	43	1095382	34	1095382
2	1095357	15	1095357	12	1095357	13	1095357
4	1095352	2	1095352	3	1095352	5	1095352
1	1095295	3	1095266	2	1095266	1	1095295
4	1095264			5	1095264	4	1095266
						7	1095264
9	$<$ 1095264	7	$<$ 1095264	13	$<$ 1095264	21	$<$ 1095264
Ave = 1095344.1		Ave = 1095345.47		Ave = 1095350.46		Ave = 1095265.45	
$Stdev$ = 267.9		$Stdev$ = 337.17		$Stdev$ = 174.50		$Stdev$ = 498.52	

(b) ADP.

GA-SRM (100,200) $p_m^{(CM)} = 0.01$ $\tau = 0.46$		GA-SRM (50,100) $p_m^{(CM)} = 0.01$ $\tau = 0.40$		GA-SRM (30,60) $p_m^{(CM)} = 0.01$ $\tau = 0.33$		GA-SRM (20,40) $p_m^{(CM)} = 0.01$ $\tau = 0.30$	
11	1095445	11	1095445	13	1095445	5	1095445
21	1095382	25	1095382	24	1095382	29	1095382
11	1095357	8	1095357	14	1095357	4	1095357
1	1095352	2	1095352	3	1095266	3	1095295
3	1095295	2	1095295	3	1095264	2	1095266
15	1095264	4	1095266			13	1095264
		6	1095264				
38	$<$ 1095264	42	$<$ 1095264	43	$<$ 1095264	44	$<$ 1095264
Ave = 1095050.63		Ave = 1094908.34		Ave = 1094877.47		Ave = 1094823.49	
$Stdev$ = 752.3		$Stdev$ = 1106.3		$Stdev$ = 986.56		$Stdev$ = 1032.36	

Table 4 Knapsack test problems.

Problem				*Parameters*		
Name	n	m	*Max*	τ	$p_m^{(CM)}$	T
Petersen 3	15	10	4015	0.48	0.067	5×10^3
Petersen 4	20	10	6120	0.52	0.050	10^4
Petersen 5	28	10	12400	0.48	0.036	5×10^4
Petersen 6	39	5	10618	0.48	0.030	10^5
Petersen 7	50	5	16537	0.48	0.020	10^5
Sento 1	60	30	7772	0.52	0.017	10^5
Sento 2	60	30	8722	0.52	0.017	10^5
Weing 7	105	2	1095445	0.40	0.01	2×10^5

were conducted. Also, although direct comparisons are not possible between GA-SRM (μ,λ) and the algorithms used in Refs. 5) and 12), looking at Tables 5–7 we can see that the proposed algorithm gives better results. It should be specially noticed the results obtained for *Weing 7* where the proposed GA-SRM found the global optimum 26% of the runs whereas genetic algorithms with constant mutation rate could not find the global optimum and the algorithm that uses a time dependent hyperbolic deterministic schedule for mutation rate control with a (15,100) selection mechanism found it only 3% of the runs 5),12).

In combinatorial problems it is important to find better solutions. Based on the observations of the final feasible solutions reached by the algorithms used in our simulations, we can see that for *Weing 7* in the ranges [1095000,1095445], [1094000,1095445], and [1093000,1095445], there are at least 30, 134, and 189 peaks of different heights, respectively. This data might help to visualize the quality of solutions reached by the algorithms.

We should mention that ADS and ADP exhibit similar behavior. Although we obtained better results with ADS for most of the knapsack test problems used here, at this time we cannot conclude whether ADS is superior to ADP. Also, the difference in performance between ADS and ADP might be relevant to the kind of epistasis 16) the test problem has. For

Table 5 Results for various knapsack test problems.

	cGA (100)			GA (50,100)			GA-SRM (50,100)					
							SRM-ADS			SRM-ADP		
Name	*N*	*Average*	*Stdev*	*N*	*Average*	*Stdev*	*N*	*Average*	*Stdev*	*N*	*Average*	*Stdev*
Petersen 3	48	4007.0	12.0	85	4013.4	3.8	100	4015.0	0.0	97	4014.7	1.2
Petersen 4	6	6031.1	50.8	35	6099.7	59.6	42	6112.5	8.4	54	6113.5	8.1
Petersen 5	2	12278.0	55.7	50	12375.1	67.5	94	12398.9	5.5	98	12399.8	1.2
Petersen 6	-	10454.5	36.5	4	10524.7	67.4	16	10588.2	37.6	16	10587.3	24.5
Petersen 7	-	16300.9	53.7	-	16367.8	93.6	23	16485.2	53.1	21	16474.2	50.0
Sento 1	-	7505.1	50.5	14	7712.7	57.8	85	7770.3	5.4	67	7765.1	9.0
Sento 2	-	8506.3	33.9	-	8682.1	31.7	55	8718.5	5.3	50	8717.7	6.6
Weing 7	-	1085421.8	1881.2	-	1092615.0	2843.4	26	1095345.5	337.17	11	1094908.3	1106.3

Table 6 Results by Khuri, et al.[12].

Name	*N*	*Average*
Petersen 3	83	4012.7
Petersen 4	33	6102.3
Petersen 5	33	12374.7
Petersen 6	4	10536.9
Petersen 7	1	16378.0
Sento 1	5	7626
Sento 2	2	8685
Weing 7	-	1093897

Table 7 Results for Weing 7 using other mutation schedules[5].

Mutation Schedule	*Selection*	*N*	*Average*
Constant mutation rate $p_m = 1/n$	(15,100) selection	-	1091268
	proportional selection	-	1093924
Self-adaptive	(15,100) selection	-	1092743
	proportional selection	-	1094311
Time-dependent hyperbolic deterministic	(15,100) selection	3	1094711
	proportional selection	-	1094479

the test problems used here we have no knowledge how high epistasis the test problems have for contiguous bits. More investigation should be done to clarify this point in the future.

5. Conclusions

This paper has explored an empirical model to improve GAs in which the genetic operators have been put in a cooperative-competitive stand with each other. The cooperativeness is sought through the parallel interaction of genetic operators and its integration with extinctive selection. SRM (*Self-Reproduction with Mutation*) acts as a permanent source of diversity to induce the appearance of beneficial mutations and CM (*Crossover and Mutation*) propagate them in the population. The extinctive selection mechanism subjects SRM's and CM's offspring to compete for survival and to guarantee the preservation of beneficial mutations for the next generation. SRM is provided with an adaptive mutation schedule that keeps control of the exploration-exploitation balance.

We found that the sexual operator CM performs better than the asexual operator SRM during the initial stages of the search. On the other hand, SMR's contribution significantly increases as the search progresses, mutation rates are reduced, and the population approaches the global optimum. Also, in spite of CM's initial effectiveness, configurations favoring SRM (mutation in general) result into better performance than configurations favoring CM. However, if the operators are properly balanced and SRM is implemented to be competitive to CM, the cooperation expected from them emerges producing a higher search velocity and higher search reliability. That was attained by introducing mutation in CM with an appropriate probability $p_m^{(CM)} = 1/n$, a $1:1$ offspring balance for CM and SRM, and a $(\mu, \lambda) = (50, 100)$ extinctive pressure in our simulations. Consequently, for the 0/1 multiple knapsack NP-hard combinatorial optimization problem the improved GA using SRM outperforms a canonical genetic algorithm as well as other enhanced GAs.

With regards to the applicability of this model to other problems we should say that so far we have observed similar behavior for image halftoning[17] problem and flowshop scheduling problem[18]. Results for these problems will be reported elsewhere.

As future works, we plan to analyze more deeply the model by using test problem generators, observe the conditions that favor the emergence of cooperative behavior between genetic operators in different classes of problems, and include higher levels of adaptation. Also, as it was mentioned before, another important issue that needs to be addressed is the effect of epistasis on the cooperative model and particularly on ADS and ADP.

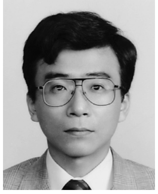
References

1) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press (1975).
2) Hinterding, R., Gielewski, H. and Peachey, T.C.: The nature of mutation in genetic algorithms, *Proc. 6th Int'l Conf. on Genetic Algorithms*, pp.65–72, Morgan Kaufmann (1995).
3) Fogarty, T.C.: Varying the Probability of Mutation in the Genetic Algorithm, *Proc. 3rd Int'l Conf. on Genetic Algorithms*, pp.104–109, Morgan Kaufmann (1989).
4) Bäck, T.: Optimal Mutation Rates in Genetic Search, *Proc. 5th Int'l Conf. on Genetic Algorithms*, pp.2–8, Morgan Kaufmann (1993).
5) Bäck, T. and Schutz, M.: Intelligent Mutation Rate Control in Canonical Genetic Algorithms, *Lecture Notes on Artificial Intelligence*, Vol.1079, pp.158–167, Springer-Verlag (1996).
6) Spears, W.: The Role of Mutation and Recombination in Evolutionary Algorithms, PhD Thesis, George Mason University (1998).
7) Spears, W.: Crossover or mutation?, *Proc. Foundations of Genetic Algorithms Workshop 2*, pp.221–237 (1993).
8) Wu, A., Lindsay, R. and Riolo, R.: Empirical Observations on the Roles of Crossover and Mutation, *Proc. 7th Int'l Conf. on Genetic Algorithms*, pp.362–369, Morgan Kaufmann (1997).
9) Aguirre, H., Tanaka, K. and Sugimura, T.: Cooperative Crossover and Mutation Operators in Genetic Algorithms, *Proc. Genetic and Evolutionary Computation Conference*, p.772, Morgan Kaufmann (1999).
10) Aguirre, H., Tanaka, K. and Sugimura, T.: Cooperative Model for Genetic Operators to Improve GAs, *Proc. IEEE Int'l Conf. on Information, Intelligence, and Systems*, pp.98–106 (1999).
11) Beasley, J.E.: Obtaining Test Problems via Internet, *Journal of Global Optimization*, Vol.8, pp.429–433 (1996).
12) Khuri, S., Bäck, T. and Heitkotter, J.: The Zero/One Multiple Knapsack Problem and Genetic Algorithms, *Proc. 1994 ACM Symp. on Applied Computing*, pp.188–193, ACM Press (1994).
13) Bäck, T.: *Evolutionary Algorithms in Theory and Practice*, Oxford Univ. Press (1996).
14) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading (1989).
15) Bäck, T. and Hoffmeister, F.: Extended Selection Mechanism in Genetic Algorithms, *Proc. 4th Int'l Conf. on Genetic Algorithms*, pp.92–99, Morgan Kaufmann (1991).
16) Altenberg, L.: Fitness Landscapes: NK Landscapes, *Handbook of Evolutionary Computation*, pp.B2.7:5–10, IOP and Oxford University Press (1997).
17) Kobayashi, N. and Saito, H.: Halftone Algorithm Using Genetic Algorithm (in Japanese), *IEICE Trans.*, Vol.J78-D-II, No.10, pp.1450–1459 (1995).
18) Yamada, T. and Reeves, C.R.: Landscape Analysis of Flowshop Scheduling Problem and Genetic Local Search (in Japanese), *Trans. IPS Japan*, Vol.39, No.7, pp.2112–2123 (1998).

(Received October 12, 1999)
(Accepted September 7, 2000)

Hernán Aguirre received his Engineer degree in computer systems from Escuela Politécnica Nacional, Ecuador, in 1992. Since 1997 he has been a Monbu-sho (Japanese Ministry of Education) research scholarship recipient. In 2000, he received the M.S. degree from Shinshu University, Japan, where he is currently a doctoral candidate. His research interests include evolutionary computation, neural networks, artificial intelligence, and their applications. He is a member of IEEE, IEICE, JSAI and ISGEC.

Kiyoshi Tanaka received B.S. and M.S. degrees from National Defense Academy, Yokosuka, Japan in 1984 and 1989, respectively. In 1992, he recieved Dr.Eng. degree from Keio University, Tokyo, Japan. In 1995, he joined the Department of Electrical and Electronic Engineering, Faculty of Engineering, Shinshu University, Nagano, Japan, where he is currently an associate professor. His research interests include image processing, information security, chaos & fractals, evolutionary computation, and their applications. He is a member of IEEE, IEICE, SPIE, EURASIP and SITA.



Tatsuo Sugimura received B.S., M.S. and Dr.Eng. degree from Osaka University, Osaka, Japan in 1976, 1979 and 1982, respectively. In 1982, he joined Matsushita Electric Industrial Co., Ltd., Japan. In 1985, he joined Fukuoka Institute of Technology, Fukuoka, Japan as an associate professor. In 1991, he moved to the Department of Electrical and Electronic Engineering, Faculty of Engineering, Shinshu University, Nagano, Japan, where he is currently an associate professor. His research interests include information theory, error correction codes, finite field, and thier applications. He is a member of IEEE, IEICE, IPSJ and SITA.