

An Analysis of a Reordering Operator with Tournament Selection on a GA-Hard Problem

Ying-ping Chen¹ and David E. Goldberg²

¹ Department of Computer Science and Department of General Engineering
University of Illinois, Urbana, IL 61801, USA

ypchen@illigal.ge.uiuc.edu

² Department of General Engineering
University of Illinois, Urbana, IL 61801, USA

deg@uiuc.edu

Abstract. This paper analyzes the performance of a genetic algorithm that utilizes tournament selection, one-point crossover, and a reordering operator. A model is proposed to describe the combined effect of the reordering operator and tournament selection, and the numerical solutions are presented as well. Pairwise, s -ary, and probabilistic tournament selection are all included in the proposed model. It is also demonstrated that the upper bound of the probability to apply the reordering operator, previously derived with proportionate selection, does not affect the performance. Therefore, tournament selection is a necessity when using a reordering operator in a genetic algorithm to handle the conditions studied in the present work.

1 Introduction

In order to ensure a genetic algorithm (GA) works well, the building blocks represented in the chromosome of the underlying problem have to be tightly linked. Otherwise, studies [1,2] have shown that a GA may fail to solve problems without such prior knowledge. Because it is difficult to guarantee that the chosen chromosome representation can provide tightly linked building blocks for processing, linkage learning operators should be adopted to overcome the difficulty, which is called the *coding trap* [3]. Currently, one way to conduct linkage learning is to use the (*gene number, allele*)-style coding scheme and reordering operators in a genetic algorithm. Reordering operators, including inversion [4, 5,6,7,8], order-based crossover operators [9,10,11,12,13,14,15], and so on, have already been studied for quite some time. The effectiveness of using an *idealized reordering operator* (IRO) has been demonstrated [3], but an upper bound on the probability to apply the IRO was also pointed out in the same work.

Since the introduction of the minimal deceptive problem (MDP) as a tool for genetic algorithm modeling and performance analysis [16], the MDP has been widely used and discussed. Some studies [3,17,18] tested the GA performance with their theoretical frameworks on the MDP, while others [19,20,21]

were interested in the nature and property of the MDP and tried to understand the relationship among the epistasis, deception, and difficulty for genetic algorithms. In the present work, we use the MDP with different initial conditions as our test problems in the theoretical model because of its simplicity for analysis.

Previous analysis on reordering [3] was based on a genetic algorithm including proportionate selection, one-point crossover, and idealized reordering operator. Because genetic algorithms nowadays usually do not use proportionate selection, this paper seeks the answer to whether the effectiveness of using a reordering operator with selection other than proportionate selection changes or not. In particular, we first modularize the previous model so that different selection operators can be easily plugged into the framework. Then tournament selection, including its variants, is put into the model with the idealized reordering operator on the minimal deceptive problem, and the performance of the model is displayed and analyzed.

The organization of this paper is in the following. The next section gives a brief review of the framework, which includes the test problems, our assumptions, and the previous results. Section 3 describes the modularization and extension of the theoretical model in detail and presents the numerical solutions. Finally, the conclusions and future work of this paper are presented in Sect. 4.

2 The Framework

In this section, we introduce the problem we use in this paper for research and analysis, the assumptions we make to build the theoretical model, and the previous results based on the model.

2.1 Minimal Deceptive Problem

In order to understand how a reordering operator can help a GA to solve problems, we have to use a test problem which is hard enough so that a GA cannot solve it by itself. On the other hand, the test problem should be not so complicated that we can easily have it theoretically analyzed. In this study, we employ a problem of known and controllable difficulty as our study subject.

In particular, the *minimal deceptive problem* (MDP) [16] is adopted as the test problem. The MDP is a two-bit problem and designed to mislead a GA away from the optimal solution and toward sub-optimal ones. There are two types of MDP [16] depending on whether $f_{0,1}$ is greater or less than $f_{0,0}$, where $f_{0,1}$ and $f_{0,0}$ are the fitness for point $(0, 1)$ and $(0, 0)$, respectively. Further analysis shows that the MDP Type II is more difficult than Type I because the GA cannot converge to the optimal solution if the initial population is biased toward the sub-optimal solution.

By utilizing the MDP Type II and setting the initial condition which makes a GA diverge, we conduct our analysis on the combined effect of a reordering operator and tournament selection. Figure 1 shows the MDP Type II, and in this paper, we have the following fitness values for each point:

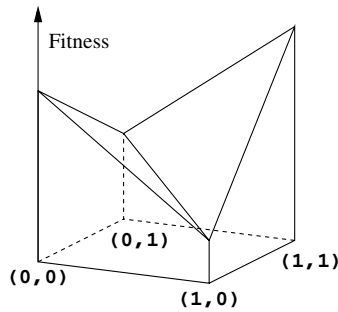


Fig. 1. The Minimal Deceptive Problem (MDP) Type II. $f_{0,0} > f_{0,1}$.

$$f_{1,1} = 1.1; \quad f_{0,0} = 1.0; \quad f_{0,1} = 0.9; \quad f_{1,0} = 0.5.$$

2.2 Assumptions

In the present paper, we study a generational genetic algorithm that combines tournament selection, one-point crossover, and a reordering operator on the MDP Type II. The following assumptions are made for simplifying the theoretical study and analysis.

First, instead of analyzing any particular reordering operator, an *idealized reordering operator* (IRO) [3] is analyzed. The IRO transfers a building block from *short* to *long* or from *long* to *short* with a *reordering probability* p_r . Here we consider the net effect produced by the IRO. The difference of a building block being short or long reflects on the effective crossover probability p_c . The longer the building block is, the more likely it will be disrupted, and vice versa.

Second, crossover events can only occur between individuals containing the building block of the identical defining length. This assumption might be untrue for actual implementations and finite populations. However, it further simplifies our analysis, makes the model more capable of displaying the transition between shorts and longs, and gives us more insights about linkage learning process.

Finally, because population portions of different schemata are considered, an infinite population is assumed implicitly as well.

2.3 Reordering and Linkage Learning

Conducting linkage learning in a GA can overcome the difficulty of the chromosome representation design when no prior knowledge about the problem structure exists. One of the straightforward methods for linkage learning is to employ the (*gene number, allele*)-style coding scheme and reordering operators. For an example of a five-bit problem, an individual 01101 might be represented as

$$((2, 1) (4, 0) (1, 0) (5, 1) (3, 1)) \quad \text{or} \quad ((5, 1) (4, 0) (3, 1) (2, 1) (1, 0)).$$

If we consider an order-two schema composed of gene 2 and gene 3, for the first case, the schema is 1***1, while it is **11* for the second case. The ordering of the (gene number, allele)'s does not affect the fitness value of the individual but affects the defining length of the schema and therefore the probability to disrupt the schema when processing. Thus, reordering operators can effectively change the linkage among genes during the evolutionary process in this manner, and it is the reason to study reordering operators as linkage learning operators in our present work.

2.4 Previous Results

A genetic algorithm with IRO on the MDP Type II was analyzed and compared to one without IRO [3]. The results showed that a GA without IRO might diverge under certain initial conditions, and IRO can help a GA to overcome such a difficulty. However, they also derived an upper bound on the probability p_r to apply the reordering operator

$$0 < p_r \leq \frac{(r-1)(1-P_f)}{r}, \quad (1)$$

where proportionate selection is used, r is the ratio of the fitness value of the optimal schema to that of the sub-optimal schema, and the converged population contains proportion of at least P_f optimal individuals.

Calculating the upper bound of p_r on the MDP Type II used in the paper is straightforward:

$$r = \frac{f_{1,1}}{f_{0,0}} = \frac{1.1}{1.0} = 1.1.$$

If at least 50% optimal solutions are desired in the converged population, the upper bound of p_r will be

$$p_r \leq \frac{(r-1)(1-P_f)}{r} = \frac{0.1}{1.1}(1-0.5) = 0.0455.$$

It was showed that if p_r is greater than the upper bound, the GA still diverges even with the help of IRO. Therefore, although IRO was demonstrated to be useful for helping a GA to overcome the *coding trap*, the upper bound of the reordering probability quite limits its applicability.

3 IRO with Tournament Selection

Now, we propose our theoretical model and analyze the combined effect of IRO and tournament selection. We start from the model developed based on using proportionate selection [3]. By separating the parts of selection and crossover and making the model modularized, we then develop the corresponding selection part of pairwise tournament selection. After adding IRO into the model, we generalize tournament selection of our model to s -ary tournament selection and probabilistic tournament selection.

3.1 Separating Selection and Crossover

Start from the model for proportionate selection [16]:

$$\begin{aligned}
 P_{0,0}^{t+1} &= P_{0,0}^t \frac{f_{0,0}}{\bar{f}} \left[1 - p_c \frac{f_{1,1}}{\bar{f}} P_{1,1}^t \right] + p_c \frac{f_{0,1}f_{1,0}}{\bar{f}^2} P_{0,1}^t P_{1,0}^t; \\
 P_{0,1}^{t+1} &= P_{0,1}^t \frac{f_{0,1}}{\bar{f}} \left[1 - p_c \frac{f_{1,0}}{\bar{f}} P_{1,0}^t \right] + p_c \frac{f_{1,1}f_{0,0}}{\bar{f}^2} P_{1,1}^t P_{0,0}^t; \\
 P_{1,0}^{t+1} &= P_{1,0}^t \frac{f_{1,0}}{\bar{f}} \left[1 - p_c \frac{f_{0,1}}{\bar{f}} P_{0,1}^t \right] + p_c \frac{f_{0,0}f_{1,1}}{\bar{f}^2} P_{0,0}^t P_{1,1}^t; \\
 P_{1,1}^{t+1} &= P_{1,1}^t \frac{f_{1,1}}{\bar{f}} \left[1 - p_c \frac{f_{0,0}}{\bar{f}} P_{0,0}^t \right] + p_c \frac{f_{1,0}f_{0,1}}{\bar{f}^2} P_{1,0}^t P_{0,1}^t,
 \end{aligned}$$

where $P_{i,j}^t$, $i, j \in \{0, 1\}$ is the portion of population of schema (i, j) at generation t , p_c is the *effective* crossover probability which combines the actual crossover probability with the disrupting probability introduced by the linkage of the schema, and \bar{f} is the average fitness value. We can separate the selection and crossover parts of the model by defining the population portion after proportionate selection as

$$Q_{i,j}^t = \frac{f_{i,j}}{\bar{f}} P_{i,j}^t \quad i, j \in \{0, 1\}.$$

By writing the model, we obtain

$$\begin{aligned}
 P_{i,j}^{t+1} &= P_{i,j}^t \frac{f_{i,j}}{\bar{f}} \left[1 - p_c \frac{f_{(1-i),(1-j)}}{\bar{f}} P_{(1-i),(1-j)}^t \right] \\
 &\quad + p_c \frac{f_{i,(1-j)}f_{(1-i),j}}{\bar{f}^2} P_{i,(1-j)}^t P_{(1-i),j}^t \\
 &= P_{i,j}^t \frac{f_{i,j}}{\bar{f}} - p_c \frac{f_{i,j}f_{(1-i),(1-j)}}{\bar{f}^2} P_{i,j}^t P_{(1-i),(1-j)}^t \\
 &\quad + p_c \frac{f_{i,(1-j)}f_{(1-i),j}}{\bar{f}^2} P_{i,(1-j)}^t P_{(1-i),j}^t \\
 &= Q_{i,j}^t - p_c Q_{i,j}^t Q_{(1-i),(1-j)}^t + p_c Q_{i,(1-j)}^t Q_{(1-i),j}^t
 \end{aligned}$$

where $i, j \in \{0, 1\}$.

Hence, the model can be described as two separate modules:

1. Proportionate selection:

$$Q_{i,j}^t = \frac{f_{i,j}}{\bar{f}} P_{i,j}^t \quad i, j \in \{0, 1\}. \tag{2}$$

2. One-point crossover:

$$P_{i,j}^{t+1} = Q_{i,j}^t - p_c Q_{i,j}^t Q_{(1-i),(1-j)}^t + p_c Q_{i,(1-j)}^t Q_{(1-i),j}^t \quad i, j \in \{0, 1\}.$$

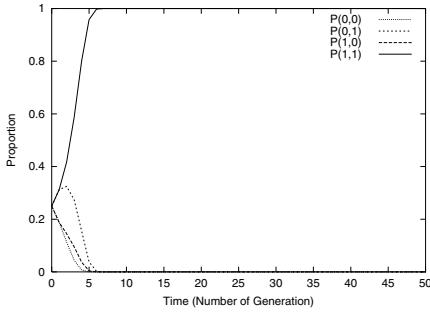


Fig. 2. Numerical solution of the MDP Type II showing convergence to the optimal solution when the initial condition is $P_{i,j}^0 = 0.25$ $i, j \in \{0, 1\}$.

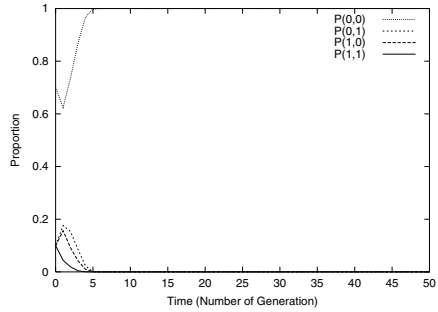


Fig. 3. Numerical solution of the MDP Type II showing divergence away from the optimal solution when the initial condition is $P_{0,0}^0 = 0.7; P_{0,1}^0 = P_{1,0}^0 = P_{1,1}^0 = 0.1$.

3.2 Pairwise Tournament Selection

After getting separate parts of the model, replacing the selection part with pairwise tournament selection is straightforward. Because the fitness values of the test function follow

$$f_{1,1} > f_{0,0} > f_{0,1} > f_{1,0},$$

we can easily write down the equations representing the portion of population after pairwise tournament selection:

$$\begin{aligned} Q_{1,1}^t &= 1 - (1 - P_{1,1}^t)^2; \\ Q_{0,0}^t &= (1 - P_{1,1}^t)^2 - (1 - (P_{1,1}^t + P_{0,0}^t))^2; \\ Q_{0,1}^t &= (1 - (P_{1,1}^t + P_{0,0}^t))^2 - (P_{1,0}^t)^2; \\ Q_{1,0}^t &= (P_{1,0}^t)^2. \end{aligned} \tag{3}$$

Substituting the proportionate selection module with the pairwise tournament selection module, we get the model combining IRO and tournament selection. Figures 2 and 3 show the numerical results of the pairwise tournament selection model for two different initial conditions. In the first initial condition, portions of all schemata are equal, i.e., $P_{i,j}^0 = 0.25$ $i, j \in \{0, 1\}$. In the second initial condition, the initial population is biased toward the sub-optimal solution that $P_{0,0}^0 = 0.7; P_{0,1}^0 = P_{1,0}^0 = P_{1,1}^0 = 0.1$.

The two initial conditions used here are identical to that used elsewhere [3] for comparison purpose. The results show that replacing proportionate selection with pairwise tournament selection alone does not make the GA capable of overcoming the difficulty. It still diverges under the second initial condition. The difference of using tournament selection is that the convergence or divergence comes much faster. Since it is well-known that the takeover time of tournament

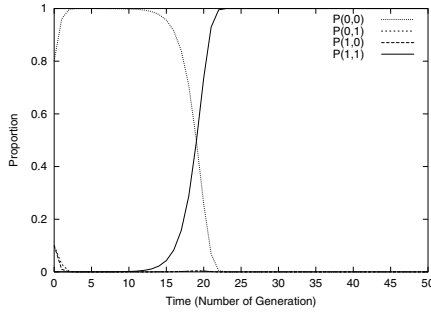


Fig. 4. Numerical solution of the MDP Type II showing convergence to the optimal solution when $p_r = 0.01$. Combined results for both short building blocks and long building blocks.

selection is much shorter than that of proportionate selection [22], the time difference is expected.

3.3 Using IRO

Apparently, replacing proportionate selection does not change the basic behavior of a GA. We now insert the *idealized reordering operator* (IRO) into our model to verify its performance. IRO is assumed to transfer a build block between its *long* version (loose linkage) and *short* version (tight linkage). For simplicity, we add another index k to the model equation terms for distinguishing short ($k = 0$) and long ($k = 1$). The difference of being long or short reflects on the effective crossover probability. If a building block is tightly linked (short), we assume that the effective crossover probability $p_{c,0} = 0$, which means the building block will not be disrupted. Otherwise, we assume $p_{c,1} = 1$, meaning the schema is very likely to be destroyed.

Because crossover events only occur between individuals of the same defining length of building blocks, we can write the crossover parts with the extra index by introducing a new intermediate portion $R_{i,j,k}^t$ as

$$R_{i,j,k}^{t+1} = Q_{i,j,k}^t - p_{c,k} Q_{i,j,k}^t Q_{(1-i),(1-j),k}^t + p_{c,k} Q_{i,(1-j),k}^t Q_{(1-i),j,k}^t \quad i, j, k \in \{0, 1\}, \tag{4}$$

where $R_{i,j,k}^t$ is the population portion of schema (i, j, k) at generation t after crossover.

After crossover, IRO is responsible for transferring a building block between its long and short version with reordering probability p_r as

$$P_{i,j,k}^{t+1} = (1 - p_r) R_{i,j,k}^t + p_r R_{i,j,(1-k)}^t \quad i, j, k \in \{0, 1\}, \tag{5}$$

where on the right hand side, the first term indicates the building blocks remaining to be the same version, and the second term specifies the building blocks transferred from the other version.

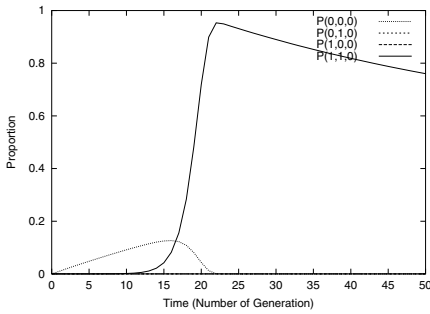


Fig. 5. Numerical solution of the MDP Type II showing convergence to the optimal solution when $p_r = 0.01$. Short building blocks.

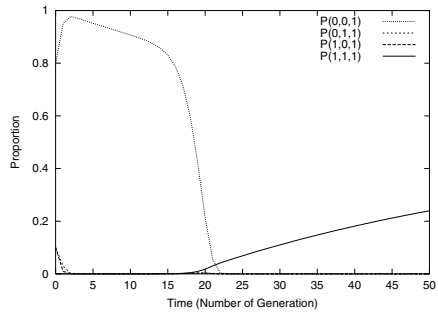


Fig. 6. Numerical solution of the MDP Type II showing convergence to the optimal solution when $p_r = 0.01$. Long building blocks.

Thus, the model with IRO consists of the following three modules:

1. Pairwise tournament selection (Equation (3));
2. One-point crossover (Equation (4));
3. Idealized reordering operator (Equation (5)).

To make the problem harder, we adopt the third initial condition that $P_{0,0}^0 = 0.8$; $P_{0,1}^0 = P_{1,0}^0 = 0.1$; $P_{1,1}^0 = 0$ [3]. This initial condition specifies that the way to have schema (1, 1) is to create it via crossover and make it stay in the population without being disrupted. We first try a low reordering probability $p_r = 0.01$ to see if the reordering operator also helps a GA to converge with tournament selection.

Figures 4, 5, and 6 show the numerical results after inserting IRO into the model. Apparently, IRO works as we expected to help the GA to converge to the optimal solutions. The process can be roughly divided into three stages. First, the short version of (1, 1) is created by the crossover. Only the short version can survive at this stage because it cannot be disrupted even both short and long versions are equally favored by the selection. Then, the optimal schema starts to takeover the population. The period of this stage is determined by the takeover time. After the optimal schema takeover the population, there is no need to maintain linkage. Therefore, the portion of long starts to grow, and the portion of short starts to decrease until reaching the balance.

Until now, there seems no fundamental difference between using proportionate selection and using tournament selection. Except for the time scale, the behavior does not seem to be different. However, if we use a higher reordering probability $p_r = 0.10$. We can get the numerical results in Figure 7.

Unexpectedly, the GA also converged to the optimal solution. Using the same reordering probability, the GA diverges instead of converges. Because the upper bound for the reordering probability was developed based on using proportionate selection, it might be different if tournament selection is used. Therefore, we

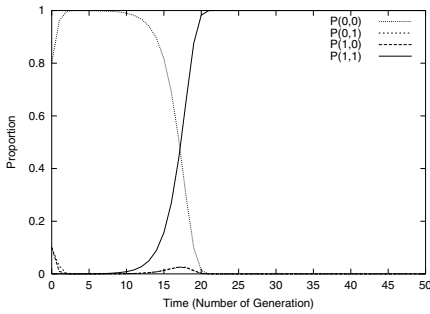


Fig. 7. Numerical solution of the MDP Type II showing convergence to the optimal solution even when $p_r = 0.10$. Combined results for both short building blocks and long building blocks.

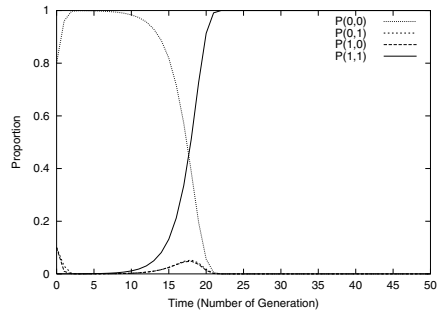


Fig. 8. Numerical solution of the MDP Type II showing convergence to the optimal solution even when $p_r = 0.25$. Combined results for both short building blocks and long building blocks.

conduct simulations with even higher reordering probabilities $p_r = 0.25, 0.75,$ and 0.99 . The results are shown in Figures 8, 9, and 10.

Surprisingly, the GA still converged to the optimal solution even with a very high reordering probability. It indicates that there might not be an upper bound for reordering probability except that $0 < p_r < 1$.

3.4 S-ary Tournament Selection

In addition to pairwise tournament selection, we also generalize the model to include the commonly used s -ary tournament selection as follows. First, we define an order function $o(\cdot)$ for each schema based on their fitness values:

$$o(0) = (-1, -1); \quad o(1) = (1, 1); \quad o(2) = (0, 0); \quad o(3) = (0, 1); \quad o(4) = (1, 0),$$

where $(-1, -1)$ is a boundary condition for convenience, and $P_{-1,-1}^t = 0 \forall t \geq 0$. Second, we define the accumulated population portion with the order given by $o(\cdot)$ as

$$A_{o(n)}^t = \sum_{m=0}^n P_{o(m)}^t \quad 0 \leq n \leq 4.$$

With the help of the ordering function and accumulated portion, we can rewrite (3) as follows:

$$Q_{o(n)}^t = \begin{cases} 0 & n = 0 \\ \left(1 - A_{o(n-1)}^t\right)^2 - \left(1 - A_{o(n)}^t\right)^2 & 0 < n \leq 4 \end{cases} \quad (6)$$

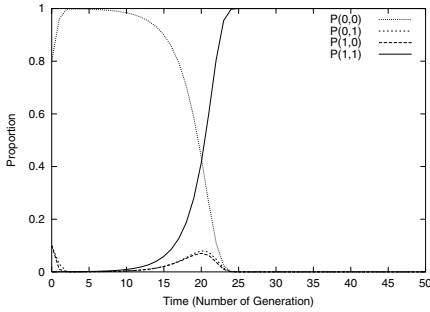


Fig. 9. Numerical solution of the MDP Type II showing convergence to the optimal solution even when $p_r = 0.75$. Combined results for both short building blocks and long building blocks.

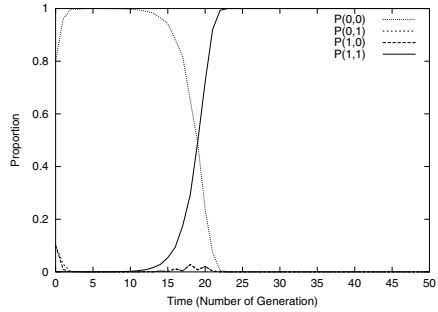


Fig. 10. Numerical solution of the MDP Type II showing convergence to the optimal solution even when $p_r = 0.99$. Combined results for both short building blocks and long building blocks.

Then we can generalize the equation to s -wise tournament selection by replacing the square with the sth power:

$$Q_{o(n)}^t = \begin{cases} 0 & n = 0 \\ \left(1 - A_{o(n-1)}^t\right)^s - \left(1 - A_{o(n)}^t\right)^s & 0 < n \leq 4 \end{cases} \quad (7)$$

3.5 Probabilistic Tournament Selection

Probabilistic tournament selection can also be modeled with our framework. Considering pairwise tournament selection, after tournament, the winner gets into the next generation with a fixed probability p , $0.5 < p \leq 1$. Pairwise tournament selection can be considered as a special case with $p = 1$.

To include probabilistic tournament selection, we start from (6). Since when some schema wins, it actually gets into the next generation with p , and it also gets selected with $1 - p$ when losing a tournament, we can modified (6) to model probabilistic tournament selection as

$$Q_{o(n)}^t = \begin{cases} 0 & n = 0 \\ p \left(\left(1 - A_{o(n-1)}^t\right)^2 - \left(1 - A_{o(n)}^t\right)^2 \right) \\ \quad + (1 - p) \left(\left(A_{o(n)}^t\right)^2 - \left(A_{o(n-1)}^t\right)^2 \right) & 0 < n \leq 4 \end{cases} \quad (8)$$

Numerical results for both s -ary tournament selection and probabilistic tournament selection are omitted because of the length limitation. Those results are also consistent with our previous observation on pairwise tournament selection. The fundamental behavior of the algorithm remains the same when different kinds of tournament selection are employed.

4 Conclusions

It has been demonstrated that an idealized reordering operator can help a genetic algorithm to overcome certain difficulty on the MDP Type II [3]. However, an upper bound of the reordering probability was also derived to explain why the reordering operator did not help the genetic algorithm when the probability was set to be a little bit high. In this paper, we extend the model to include the commonly used tournament selection. The proposed model can describe pairwise tournament selection, s -ary tournament selection, and probabilistic tournament selection. By analyzing the performance of a GA with the proposed model, we can find that there seems no upper bound on reordering probability. The genetic algorithm still converges even if the probability approaches to 1. Therefore, using the reordering operator with tournament selection can give us much better results than with proportionate selection.

Tournament selection has been widely utilized because of its excellence, including independence of fitness scaling, ease for implementation, and so on. However, based on the study, when conducting linkage learning, tournament selection becomes a necessity rather than a choice. If proportionate selection is used, the reordering probability has to be limited to ensure success. But in practice, the reordering probability might not be known or even controllable. Hence, we have to use tournament selection under this condition so that the algorithm is able to achieve its goal.

Future work along this line includes extending the model so that more operators and parameters can be described, using the model to observe and explain the linkage learning process, and deriving characteristic properties, such as tightness time, of linkage learning.

Acknowledgments. The work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

References

1. Goldberg, D.E., Deb, K., Clark, J.H.: Genetic algorithms, noise, and the sizing of populations. *Complex Systems* **6** (1992) 333–362
2. Goldberg, D.E., Deb, K., Thierens, D.: Toward a better understanding of mixing in genetic algorithms. *Journal of the Society of Instrument and Control Engineers* **32** (1993) 10–16

3. Goldberg, D.E., Bridges, C.L.: An analysis of a reordering operator on a GA-hard problem. *Biological Cybernetics* **62** (1990) 397–405
4. Bagley, J.D.: *The Behavior of Adaptive Systems Which Employ Genetic and Correlation Algorithms*. PhD thesis, University of Michigan (1967) (University Microfilms No. 68-7556).
5. Cavicchio, Jr., D.J.: *Adaptive Search Using Simulated Evolution*. Unpublished doctoral dissertation, University of Michigan, Ann Arbor, MI (1970) (University Microfilms No. 25-0199).
6. Frantz, D.R.: *Non-linearities in genetic adaptive search*. PhD thesis, University of Michigan (1972) (University Microfilms No. 73-11,116).
7. Holland, J.H.: *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI (1975)
8. Smith, S.F.: A learning system based on genetic adaptive algorithms. *Dissertation Abstracts International* **41** (1980) 4582B (University Microfilms No. 81-12638).
9. Davis, L., Smith, D.: Adaptive design for layout synthesis. Internal report, Texas Instruments, Dallas (1985)
10. Davis, L.: Applying adaptive algorithms to epistatic domains. *Proceedings of the 9th International Joint Conference on Artificial Intelligence* **1** (1985) 162–164
11. Goldberg, D.E., Lingle, Jr., R.: Alleles, loci, and the traveling salesman problem. *Proceedings of an International Conference on Conference on Genetic Algorithms* (1985) 154–159
12. Oliver, I.M., Smith, D.J., Holland, J.R.C.: A study of permutation crossover operators on the traveling salesman problem. *Proceedings of the Second International Conference on Genetic Algorithms* (1987) 224–230
13. Whitley, D., Starkweather, T., Fuquay, D.: Scheduling problems and traveling salesmen: The genetic edge recombination operator. *Proceedings of the Third International Conference on Genetic Algorithms* (1989) 133–140
14. Banzhaf, W.: The 'molecular' traveling salesman. *Biological Cybernetics* **64** (1990) 7–14
15. Starkweather, T., McDaniel, S., Mathias, K., Whitley, D., Whitley, C.: A comparison of genetic sequencing operators. *Proceedings of the Fourth International Conference on Genetic Algorithms* (1991) 69–76
16. Goldberg, D.E.: Simple genetic algorithms and the minimal, deceptive problem. In Davis, L., ed.: *Genetic algorithms and simulated annealing*. Morgan Kaufmann Publishers, Inc., Los Altos, CA (1987) 74–88
17. Yamamura, M., Satoh, H., Kobayashi, S.: An Markov analysis of generation alternation models on minimal deceptive problems. In Wang, P.P., ed.: *International Conference of Information Sciences: Fuzzy Logic, Intelligent Control & Genetic Algorithm*. Volume 1., Durham, NC, Duke University (1997) 47–50
18. Novkovic, S., Šverko, D.: The minimal deceptive problem revisited: the role of 'genetic waste'. *Computers and Operations Research* **25** (1998) 895–911
19. Grefenstette, J.J.: Deception considered harmful. *Foundations of Genetic Algorithms 2* (1993) 75–91
20. Forrest, S., Mitchell, M.: What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation. *Machine Learning* **13** (1993) 285–319
21. Naudts, B., Verschoren, A.: Epistasis and deceptivity. *Simon Stevin - Bulletin of the Belgian Mathematical Society* **6** (1999) 147–154
22. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms* **1** (1991) 69–93 (Also TCGA Report 90007).