

Comparison of Selection Methods for Evolutionary Optimization

Research Paper

Byoung-Tak ZHANG

Artificial Intelligence Lab (SCAI), Dept. of Computer Engineering, Seoul National University, Seoul 151-742, Korea

btzhang@scai.snu.ac.kr

Jung-Jib KIM

Artificial Intelligence Lab (SCAI), Dept. of Computer Engineering, Seoul National University, Seoul 151-742, Korea

jjkim@scai.snu.ac.kr

Abstract. Selection is an essential component of evolutionary algorithms, playing an important role especially in solving hard optimization problems. Most previous studies on selection have focused on more or less ideal properties based on asymptotic analysis. In this paper, we address the selection problem from a more practical point of view by considering solution quality achievable within acceptable time. The repertoire of methods we compare includes proportional selection, ranking selection, linear ranking, tournament, Genitor selection, simulated annealing, and hill-climbing. All these methods use genetic operators in one form or another to create new search points. Experiments are performed in the context of the machine layout design problem. This problem is a real industrial application having both continuous and discrete optimization characteristics. The experimental results for solving two-row machine layout problems of size ranging from 10 to 50 show strong evidence that ranking and tournament selection are, in general, more effective in both solution quality and convergence time than proportional selection and other methods. We provide a theoretical explanation of the experimental results using a predictive model of evolutionary optimization based on selection differential and response to selection.

Keywords: Evolutionary optimization, machine layout design, selection methods, selection differential, response to selection, heritability

1. Introduction

Historically, four evolutionary computation paradigms have emerged. They include evolution strategies, evolutionary programming, genetic algorithms, and genetic programming. Though during the last years these paradigms have crossbred and thus the distinctions blurred, each of them has different features.

The main difference between these evolutionary algorithms lies in their representation and genetic operators (Bäck, 1996). The operators are closely related with the underlying representation scheme of each evolutionary algorithm. Genetic algorithms typically work on fixed-size bit-strings using crossover as its main operator. Evolution strategies are based on vectors of real values for representation and use mutation as the main operator. Evolutionary programming usually manipulates graphs using mutation as the single genetic operator. Genetic programming represents individuals as trees of flexible size. Each of the evolutionary algorithms has also developed its own selection scheme. Genetic algorithms and genetic programming use proportional selection. Evolution strategies have developed

various ranking-based selection schemes. Evolutionary programming is used usually with tournament selection.

Selection plays a major role in evolutionary algorithms since it determines the direction of search whereas other genetic operators propose new search points in an undirected way. The term selective pressure is usually used to denote the strength of selection to influence the directedness of the search. Whitley, 1989, argues that population diversity and selective pressure are the only two primary factors in genetic search and that the main problem consists in the inverse relation. Hard selection emphasizes exploitation of information gained so far and leads to high convergence speed. Soft selection, on the other hand, emphasizes genotypic diversity and explorative search behavior, leading to good convergence reliability. Characterization of the distinctive features of various selection methods is a prerequisite for effective application of evolutionary algorithms and for the design of new algorithms.

Several researchers have studied selection schemes. Goldberg and Deb, 1991, provide an analysis on the convergence time and growth ratio of several selection schemes. Introducing the term takeover time, they considered the sole effect of selection, eliminating the influence of other genetic operators such as crossover and mutation. Bäck and Hoffmeister, 1991, studied the connection between takeover time and genotypic diversity. By varying selective pressure on a set of test functions, they verified that slow convergence behaviors of some selection mechanisms have to do with large genotypic diversity. Mühlenbein and Schlierkamp-Voosen, 1993, were the first to use the population genetics terms such as selection differential, response to selection, and selection intensity which are measured as the change of the average fitness of the population. Based on these concepts they presented predictive models that allow the prediction of behavior of evolutionary algorithms. Blickle and Thiele, 1995, defined selection variance as the expected variance of the fitness distribution of the population after selection and made a mathematical analysis of tournament selection to predict the fitness values. For simplicity of theoretical analysis, most of these studies have considered more or less ideal problem settings.

The aim of this paper is to shed light on the performance characteristics of various selection methods in real-life problem solving situations. We compare the performance of major selection schemes in terms of solution quality and convergence time. The repertoire of selection methods compared here includes proportional selection, ranking selection, linear ranking, tournament, Genitor selection, simulated annealing, and hill-climbing. Our problem setting is practical in several points. First, we take into account the effect of crossover, mutation, and other factors for implementing specific selection mechanisms. Second, we compare the average performance of the methods under the usual termination condition, i.e., the optimization process halts if it either reaches the maximum generation or does not improve its fitness within a prespecified number of consecutive generations. In practice, time limit does not allow the entire population to converge to a single identical solution as required by theoretical measures such as takeover time. Third, we use the machine layout problem as a testbed for this comparative study. This problem provides a useful benchmark due to its importance in engineering design, scalability of problem complexity, and its characteristic as a mixed discrete and continuous optimization task (Zhang and Mühlenbein, 1995; Zhang et al., 1997).

The paper is organized as follows. Section 2 defines the machine layout problem. Section 3 describes the evolutionary optimization approach to this problem and presents the selection methods to be compared. Experimental results are given in Section 4. Section 5 compares the empirical results with those by theoretical analysis. The evolutionary

dynamics of selection methods is analyzed by measuring the selection differential and the response to selection. Section 6 summarizes the results and their implications.

2. Machine Layout Design

Machine layout design is concerned with arrangement of physical machines in a working area. This problem can be formulated as an optimization task in which the best facility layout is sought by optimizing some measure of performance, such as material handling costs, subject to some constraints (Francis, et al. 1992). Heragu and Kusiak, 1988, have formulated a multi-row machine layout problem as a general two-dimensional continuous space allocation problem. In many practical situations, however, the machines are arranged along well-defined rows and this problem can be viewed as discrete in one dimension and continuous in another dimension (Gen and Cheng, 1997).

Let x_i and y_i be the distances from the center of machine i to vertical and horizontal reference lines, respectively (Figure 1). Let the decision variable z_{ir} be

$$z_{ir} = \begin{cases} 1, & \text{if machine } i \text{ is allocated to row } r \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The multi-row machine layout problem with unequal area can be formulated as a mixed-integer programming problem:

$$\min \sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij} f_{ij} (|x_i - x_j| + |y_i - y_j|) \quad (2)$$

$$s.t. \quad |x_i - x_j| z_{ir} z_{jr} \geq \frac{1}{2} (l_i + l_j) + d_{ij}, \quad i, j = 1, \dots, N \quad (3)$$

$$y_i = \sum_{r=1}^R l_0 (r-1) z_{ir}, \quad i = 1, \dots, N \quad (4)$$

$$\sum_{r=1}^R z_{ir} = 1, \quad i = 1, \dots, N \quad (5)$$

$$\sum_{i=1}^N z_{ir} < N, \quad r = 1, \dots, R \quad (6)$$

$$x_i, y_i \geq 0, \quad i = 1, \dots, N \quad (7)$$

$$z_{ir} = 0, 1, \quad i = 1, \dots, N, \quad r = 1, \dots, R \quad (8)$$

where the symbols are explained in Table 1.

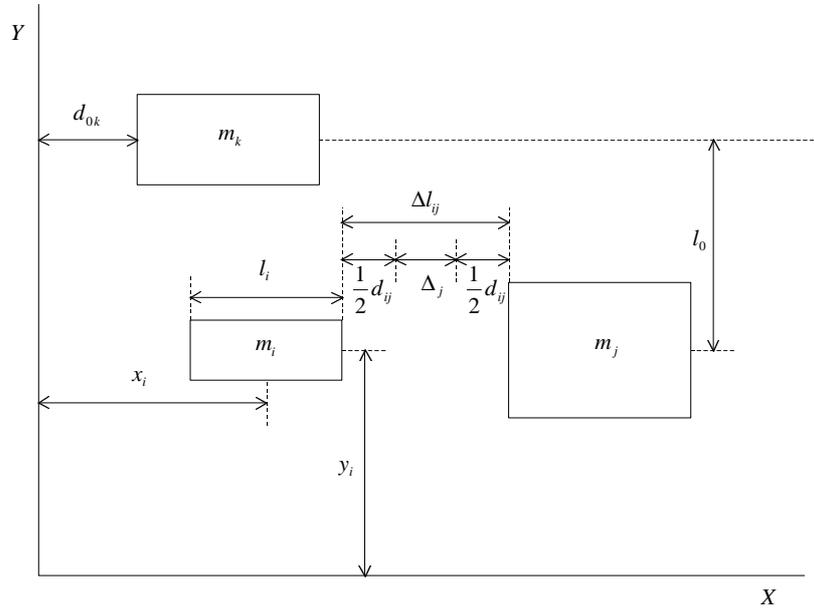


Figure 1 Machine layout problem.

Table 1 Symbols for the definition of the machine layout problem.

Symbol	Description
N	number of machines
R	number of rows
f_{ij}	frequency of trips between machines i and j
c_{ij}	handling cost per unit distance travelled between machines i and j
l_i	length of machine i
l_0	separation between two adjacent rows
d_{ij}	minimum distance (clearance) between machines i and j
x_i	distance between center of machine i and the vertical reference line l_v
y_i	distance between center of machine i and the horizontal reference line l_h

The objective is to minimize the total cost involved in making the required trips between the machines. Constraint (3) ensures that no two machines in this layout overlap. Constraints (5)-(6) ensure that one machine is assigned only to one row. Variable y_i in (4) is not necessary in this model because it can be determined by z_{ik} , but with this notation the model can be easily understood. Inequality (7) is a nonnegativity constraint.

Because of the combinatorial nature of machine layout design, heuristic techniques are the most promising approach for solving layout problems of practical size. Recently, evolutionary optimization methods have been used for machine layout design and related problems. Tate and Smith, 1995, applied genetic algorithms to shape-constrained unequal area facility layout problems. Cheng and Gen, 1996, proposed genetic algorithms for solving multi-row machine layout problems.

3. Selection Methods for Evolutionary Design

Evolutionary algorithms are population-based search methods. The search begins with randomly initialized individuals (or chromosomes) which are iteratively improved. Genetic operators are used to modify the existing individuals and better ones are selected into the next generation.

For the R -row layout problem of N machines, the chromosome a can be represented as

$$a := [(s_1, s_2, \dots, s_{R-1}), (m_{i_1}, m_{i_2}, \dots, m_{i_n}), (\Delta_{i_1}, \Delta_{i_2}, \dots, \Delta_{i_n})], \quad (9)$$

where m_{i_j} represents the machine in the j th position, and Δ_{i_j} denotes the neat clearance between machines $m_{i_{j-1}}$ and m_{i_j} . The separators s_k , $1 \leq s_k \leq N$ and $s_k < s_{k+1}$, denote the boundary position to separate the list into R parts according to the R -row requirement. A machine in the k th partition is assigned to the corresponding row.

The initial population is created by randomly generating the separator list, the machine permutation list, and the neat clearance list. Separators are simply random integers $s_k \in \{1, 2, \dots, N\}$. A new machine list is generated by a permutation of machine symbols with $m_{i_j} \in \{m_1, m_2, \dots, m_N\}$. The neat clearance list is generated within an allowable region. Each field of the layout is modified by crossover and mutation operators. The machine sequences are rearranged by crossover of parent sequences. Mutation is performed by selecting a chromosome and then picking up a random neat clearance which is perturbed.

For fitness evaluation, the x_j and y_j values of the j th machine are calculated from the coordinates of the j th machine. In case of the two-row machine layout problem, chromosomes are decoded as follows:

$$x_j = x_i + d_{ij} + \Delta_j + \frac{1}{2}(l_i + l_j) \quad (10)$$

$$y_j = \begin{cases} 0, & \text{if } m_j \text{ belongs to the first row} \\ l_0, & \text{otherwise.} \end{cases} \quad (11)$$

where x_i and Δ_j are defined as:

$$x_i = d_{0i} + \Delta_i + \frac{1}{2}l_i \quad (12)$$

$$\Delta_j = \Delta l_{ij} - d_{ij} \quad (13)$$

The fitness of an individual a_k is evaluated in terms of total cost and penalty to illegality:

$$f(a_k) = \frac{1}{F_k + \mathbf{e}_k K}, \quad k = 1, 2, \dots, I \quad (14)$$

where F_k is the total cost, k is a positive penalty value, and \mathbf{e}_k is the penalty coefficient (defined below). The total cost for the k -th chromosome is given by

$$F_k = \min \sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij} f_{ij} (|x_i^k - x_j^k| + |y_i^k - y_j^k|). \quad (15)$$

To calculate the penalty for violation of working areas, let L_k^1 and L_k^2 be the necessary working areas required by machines which are arranged in the first and second rows,

respectively, for the given chromosome a_k , and let $L_k^u = \max\{L_k^1, L_k^2\}$. Then, the penalty coefficient e_k is calculated as follows:

$$e_k = \begin{cases} 0, & \text{if } L_k^u - L \leq 0 \\ L_k^u - L, & \text{otherwise.} \end{cases} \quad (16)$$

After the fitness of chromosomes is evaluated, fitter individuals are selected into the next generation. In the following we describe the selection methods we consider in this paper.

In proportional selection, the individuals are selected according to their relative fitness values. The selection probability of i th individual at generation t is defined as

$$p_s(a_i^t) = \frac{f(a_i^t)}{\sum_{j=1}^I f(a_j^t)}. \quad (17)$$

This is a probabilistic selection method in which every individual having non-zero fitness will have a chance to be reproduced. This selection scheme is adopted by the simple genetic algorithm and believed to be the most similar mechanism that occurs in nature.

Tournament selection (Blickle and Thiele, 1995) is performed by choosing parents randomly and reproducing the best individual from this group. When the number of parents is q , this is called the q -tournament selection. This process is repeated I times to produce the next generation of individuals. The selection probabilities for q -tournament selection are given by

$$p_s(a_i^t) = \frac{1}{I^q} ((I - i + 1)^q - (I - i)^q). \quad (18)$$

Ranking selection is a selection method which assigns selection probabilities solely on the basis of the rank i of individuals, ignoring absolute fitness values. In (m, I) uniform ranking (Schwefel, 1995), the best m individuals are assigned a selection probability of $1/m$ while the rest are discarded:

$$p_s(a_i^t) = \begin{cases} \frac{1}{m}, & 1 \leq i \leq m \\ 0, & m < i \leq I. \end{cases} \quad (19)$$

In (m, I) linear ranking (Baker, 1985), the individuals are assigned selection probabilities which are proportional to their ranking:

$$p_s(a_i^t) = \begin{cases} \frac{1}{m} \left(h_{\max} - 2(h_{\max} - 1) \frac{i-1}{m-1} \right), & 1 \leq i \leq m \\ 0, & m < i \leq I. \end{cases} \quad (20)$$

The parameter h_{\max} controls the selection pressure.

Another selection method we compared is the Genitor selection (Whitley, 1989) which is a steady-state selection method. The original version works individual by individual. Each time one individual is chosen according to linear ranking and then the worst individual in the population is replaced. We use here a modified version of this. Instead of choosing one individual we choose two individuals which undergo crossover and mutation to generate two offspring which then replace two worst individuals in the current population.

We also study the performance of standard optimization methods augmented with genetic operators, i.e. random-mutation hill-climbing and simulated annealing. The hill-climbing algorithm is a greedy method that generates one alternative and accepts it deterministically if it is better than current one. Otherwise, it generates another state. The selection probability of individuals is given as:

$$p_s(a^t) = \begin{cases} 1, & \text{if } f(a^t) > f(a^{t-1}) \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

Simulated annealing is a stochastic greedy method that chooses deterministically the new configuration a^t if the probability of the state a^t is greater than the probability of the current state a^{t-1} . Otherwise, the state a^t is chosen with the probability given by $\exp\{-(f(a^t)-f(a^{t-1}))/kT\}$. More formally, the probability of each new state is given as

$$p_s(a^t) = \begin{cases} 1, & \text{if } \exp\left(-\frac{f(a^t)-f(a^{t-1})}{kT}\right) \geq 1 \\ \exp\left(-\frac{f(a^t)-f(a^{t-1})}{kT}\right), & \text{otherwise.} \end{cases} \quad (22)$$

Here k is the Boltzmann's constant and T is the temperature parameter that controls the greediness of this method. T is scheduled to start with a large value and then decrease.

It should be noted that the last two algorithms are point-based search methods. For these algorithms, we use the same genetic operators as in other evolutionary search except the selection operator.

4. Experimental Results

Experiments have been performed for two-row machine layout problems. The number of machines was $N=10, 20, 30, 40, 50$. For each problem size, 100 problem instances were generated at random and the selection methods were compared on this set of instances. Optimization process was terminated if 50 consecutive generations achieved no improvement in fitness or when the maximum number of generations allowed was reached. For all selection methods except simulated annealing, we employed the elitist strategy to ensure nondecreasingness in best fitness values. For steady-state and point-based selection schemes, a single generation was considered equivalent to the number of selection steps necessary for generating the population size. For larger size problems, we used a larger population size and a greater number of generations than for small size problems (Table 2).

Some preliminary experiments were made to find out the best parameter values for each selection scheme. Following the results as shown in Figures 2 and 3, we have chosen the tournament size of $q=2$ for tournament selection and the truncation threshold of 0.6 for ranking selection. The parameter for linear ranking was $h_{max} = 1.1$ as recommended in Baker, 1985. Crossover rate was 0.4 and mutation rate was 0.2.

Performance of selection methods was compared in terms of solution quality and convergence time. Table 3 summarizes the fitness values of selection methods at the time the convergence criterion was met. It can be observed that tournament selection (TS), ranking selections (RS and RSL), and Genitor selection (GS) obtained comparable performance in average fitness which is significantly better than that of proportional selection (PS), simulated annealing (SA), and hill-climbing (HC). SA and HC were slightly better than PS, except for $N=10$.

The convergence time of selection methods was compared in Table 4. It can be seen that point-based selection (SA and HC) and ranking selection were best in convergence speed. Ranking selection was 3-4 times faster than tournament selection which was faster than proportional selection by a factor of approximately two. Genitor selection was relatively fast for small size problems ($N=10$), but it was very slow for large size problems ($N=50$). Table 4 also shows the standard deviation of the convergence time. Tournament and Genitor selection had relatively large deviations.

Combined performance of solution quality and convergence speed was depicted in Figures 4 and 5 for $N=20$ and $N=50$, respectively. Shown are the evolution of average fitness values for each method. The curves clearly indicate that the selection methods can be classified into three categories: (TS, RS, GS), (SA, HC), and (PS). Figure 6 shows the distribution of runs for the selection schemes (PS, TS, RS, GS) in the fitness-time space. In this scatter diagram, the points located in the upper left corner represent good runs, achieving high-quality solutions with a small amount of computation time. To summarize, tournament selection and ranking selection outperformed the proportional selection in terms of a combined performance measure of solution quality and optimization speed. They also outperformed SA and HC. Genitor selection achieved sometimes good solutions but usually with a large amount of time.

Table 2 Parameter values used for experiments.

Parameter	Value				
Number of machines N	10	20	30	40	50
Population size P	200	400	600	800	1000
Max generation G_{max}	2000	4000	6000	8000	10000

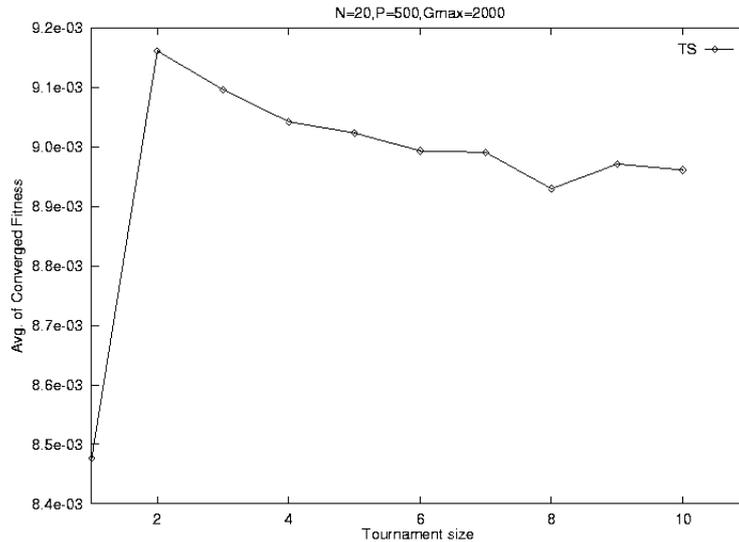


Figure 2 Finding the best tournament size q .

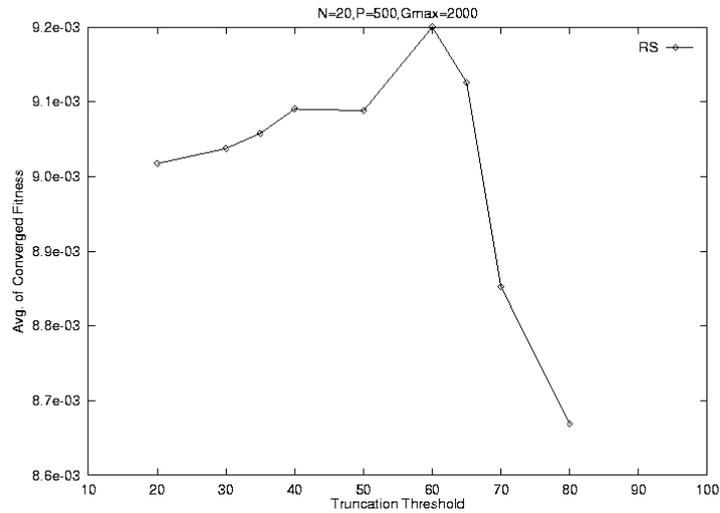


Figure 3 Finding the best truncation rate.

Table 3 Comparison of solution quality (greater is better).

N	unit	PS	TS	RS	RSL	GS	SA	HC
10	$\times 10^{-2}$	6.277 ± 1.159	6.476 ± 1.093	6.465 ± 1.087	6.459 ± 1.095	6.460 ± 1.094	6.096 ± 1.121	6.113 ± 1.097
20	$\times 10^{-3}$	8.394 ± 1.317	9.368 ± 1.018	9.376 ± 1.034	9.384 ± 1.065	9.311 ± 1.062	8.399 ± 1.201	8.615 ± 1.178
30	$\times 10^{-3}$	2.608 ± 0.291	2.858 ± 0.235	2.853 ± 0.229	2.850 ± 0.230	2.838 ± 0.236	2.631 ± 0.330	2.681 ± 0.341
40	$\times 10^{-3}$	1.143 ± 0.115	1.231 ± 0.098	1.232 ± 0.095	1.229 ± 0.095	1.211 ± 0.106	1.186 ± 0.129	1.186 ± 0.132
50	$\times 10^{-4}$	5.958 ± 0.504	6.416 ± 0.496	6.418 ± 0.500	6.412 ± 0.499	6.409 ± 0.525	6.009 ± 0.666	6.131 ± 0.577

Table 4 Comparison of convergence time (number of function evaluations).

N	unit	PS	TS	RS	RSL	GS	SA	HC
10	$\times 10^5$	2.778 ± 0.928	1.510 ± 0.924	0.487 ± 0.254	0.524 ± 0.239	1.756 ± 1.202	0.215 ± 0.130	0.282 ± 0.143
20	$\times 10^5$	8.274 ± 3.504	3.953 ± 3.850	0.967 ± 0.356	0.917 ± 0.334	6.831 ± 7.530	1.033 ± 0.435	1.146 ± 0.589
30	$\times 10^5$	13.097 ± 5.434	6.254 ± 6.072	1.851 ± 0.511	1.603 ± 0.537	13.875 ± 18.994	2.200 ± 0.840	4.084 ± 1.800
40	$\times 10^5$	19.443 ± 5.350	8.052 ± 8.001	2.954 ± 0.605	2.456 ± 0.529	20.745 ± 27.964	4.810 ± 2.128	9.427 ± 3.946
50	$\times 10^5$	27.611 ± 9.206	12.713 ± 11.696	4.475 ± 0.975	4.038 ± 0.959	33.253 ± 44.378	7.231 ± 2.833	4.724 ± 0.303

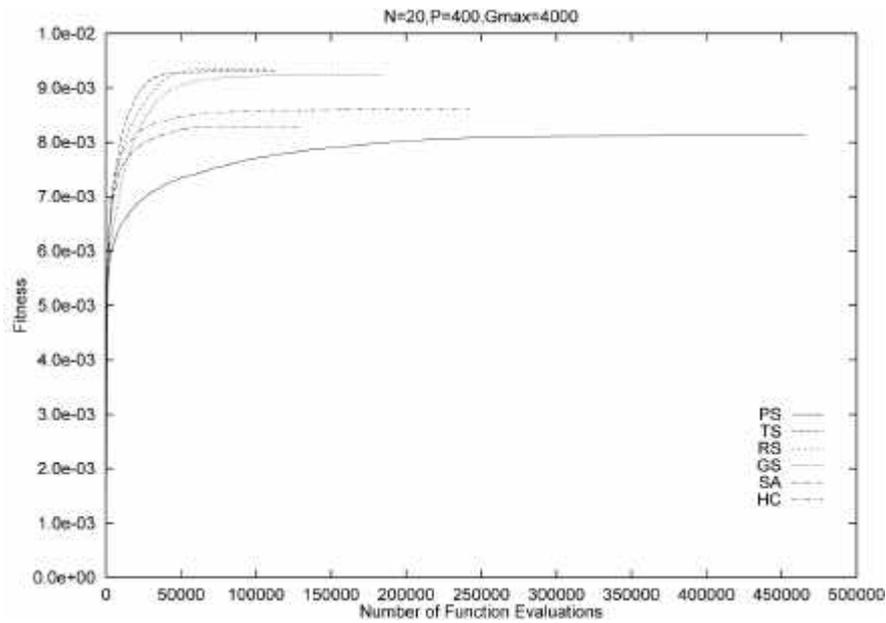


Figure 4 Evolution of fitness values averaged over 100 runs ($N=20$).

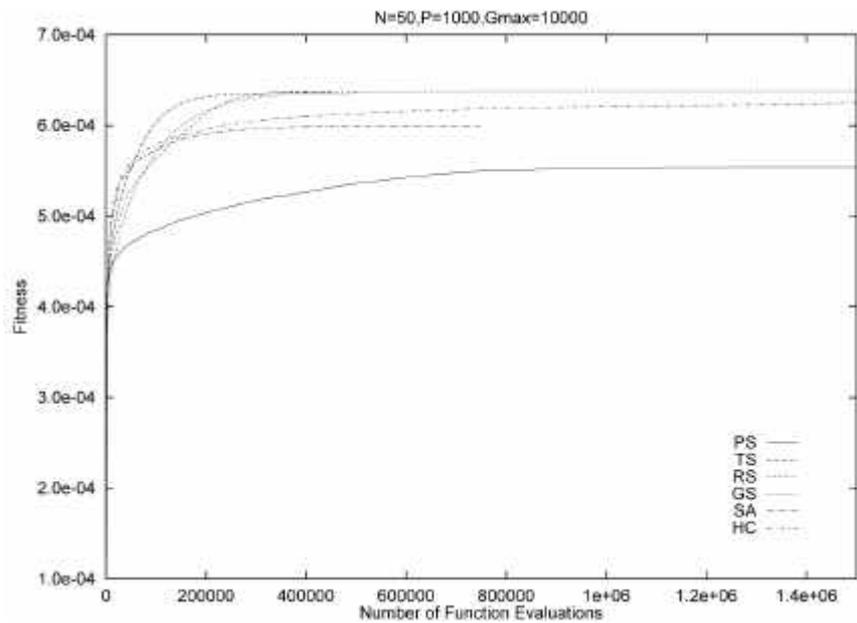


Figure 5 Evolution of fitness values averaged over 100 runs ($N=50$).

5. Analysis

Analysis of selection in evolutionary algorithms can be made at three different levels: the fitness of the best individual (Goldberg and Deb, 1991; Bäck, 1996), the average fitness

of the population (Mühlenbein and Schlierkamp-Voosen, 1993), and the distribution of fitness values in the population (Blickle and Thiele, 1995). In the following, we analyze the experimental results by comparing them with those of theoretical analysis. This analysis is based on the best fitness. We also examine the behavior of selection schemes in terms of the selection differential and the response to selection (Mühlenbein and Schlierkamp-Voosen, 1993) which are measures of selective pressure based on average fitness.

5.1 Convergence Time

Goldberg and Deb, 1991, introduced the concept of takeover time to characterize the effect of selection in the absence of any genetic operators. Takeover time is defined as the number of generations at which the population contains $I - 1$ or more copies of the best individual a_{best}^0 contained in the initial population. Let $B(t)$ denote the set of individuals at t th generation that are the same as a_{best}^0 :

$$B(t) = \{ i \mid a_i^t = a_{best}^0, \quad i = 1, \dots, I \}. \quad (23)$$

The takeover time t^* is then defined as

$$t^* = \min \{ t \mid size(B(t)) \geq I - 1 \}, \quad (24)$$

where $size(B(t))$ is the number of elements in $B(t)$. Small takeover times characterize strong selective pressure while large takeover times impose weak selective pressure on the search. Takeover time is useful for the comparative analysis of convergence speed of several selection schemes. But, this does not take into account the effect of other genetic operators such as crossover and mutation.

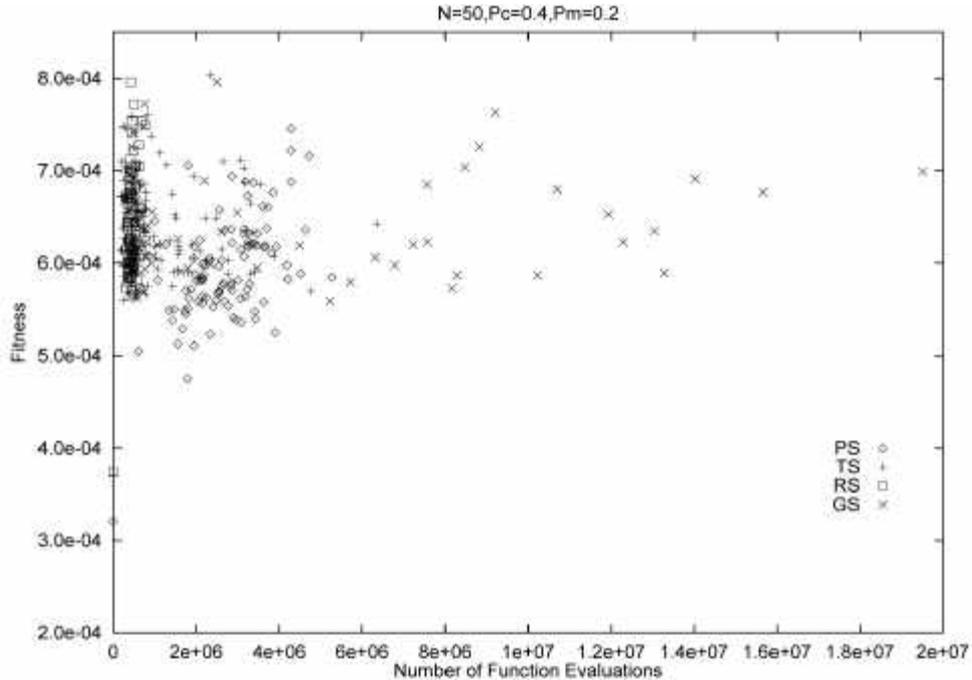


Figure 6 Distribution of final fitness values for various selection schemes ($N=50$).

In our experiments, the convergence time t was measured as follows:

$$t = \min\{t_{emp}, G_{max}\}, \quad (25)$$

where G_{max} is the maximum number of generations allowed and t_{emp} is the empirical convergence time measured as

$$t_{emp} = \min\{t \mid f_{best}(t) = f_{best}(t - \Delta t)\}, \quad (26)$$

Here $f_{best}(t)$ is the best fitness at generation t and $f_{best}(t - \Delta t)$ is the best fitness of Δt generations before t . It should be noted that we check the phenotypic convergence rather than the genotypic as used in takeover time.

Table 5 compares the theoretical takeover times derived in Goldberg and Deb, 1991, with our empirical results in terms of convergence time and solution quality. For clarity, the table shows only relative values in terms of rank and the number of function evaluations (the values in parentheses). As expected, convergence time alone does not suggest much on what is the best selection method for the given problem since it does not take into account the solution quality. For example, in terms of convergence speed PS is better than GS. However, considering the solution quality together, PS is worse than GS as the values in the table indicate.

5.2 Response to Selection

To explain the experimental results we analyzed the dynamics of population fitness using the concepts of the selection differential and the response to selection as suggested by Mühlenbein and Schlierkamp-Voosen, 1993. The response to selection is defined as the difference between the population mean fitness of generation $t+1$ and the population mean of generation t . $R(t)$ measures the expected progress of the population:

$$R(t) = M(t+1) - M(t), \quad (27)$$

where $M(t)$ denotes the average fitness of the population at generation t . Selection differential is defined as the difference between the mean fitness of the selected parents $M_s(t)$ and the mean fitness of the population:

$$S(t) = M_s(t) - M(t), \quad (28)$$

Table 5 Comparison of theoretical and empirical results in convergence time. Empirical results are for the machine layout problem with $N=50$. Shown are the ranks of performance with their actual values in parentheses.

Method	Takeover time	Empirical time	Empir. quality
Ranking selection (RS)	$O(N \ln n)$	1 (4.4)	1 (6.4)
Tournament selection (TS)	$O(\ln n)$	2 (12.7)	1 (6.4)
Proportional selection (PS)	$O(N \ln n)$	3 (27.6)	4 (5.9)
Genitor selection (GS)	$O(\ln n)$	4 (33.2)	1 (6.4)

These measures are useful to make predictions on the behavior of evolutionary algorithms. That is, the designer of the algorithm can try to predict the $R(t)$ values from the $S(t)$ values based on the equation:

$$R(t) = b_r S(t), \quad (29)$$

where b_t is called the realized heritability in quantitative genetics. Assuming b_t is a constant for a certain number of generations, the cumulative response R_s for s generations can be predicted by

$$R_s = \sum_{t=1}^s R(t) = b \sum_{t=1}^s S(t). \quad (30)$$

The response to selection is the product of the heritability and the selection differential.

Figures 7 and 8 depict the average values of $S(t)$ and $R(t)$ over 20 runs for each selection method. In this analysis we compared four methods (PS, TS, RS, and GS) on the layout design problem of size $N=20$. It is interesting to note that there are some clear difference in the selection differential curves (Figure 7), but the response-to-selection curves (Figure 8) converge to more or less similar values as generation goes on. Figure 9 shows the actual values for $S(t) - R(t)$. It can be seen that RS and TS, the two best selection methods in our experiments, have relatively large difference values.

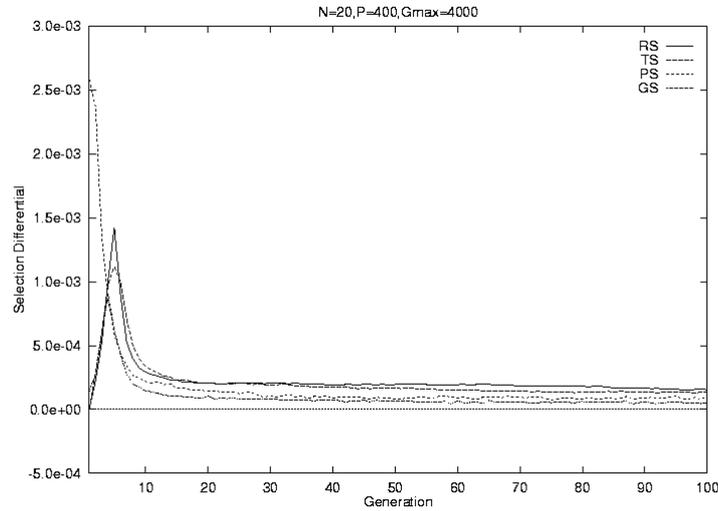


Figure 7 Selection differential curves.

In theory, large $S(t) - R(t)$ values for a fixed $S(t)$ indicate small heritability since

$$S(t) - R(t) = S(t) - b_t S(t) = (1 - b_t) S(t), \quad (31)$$

where we used the relationship (29) and b_t is the realized heritability. Combining this theory with the experimental results, we can conclude that having large selection differential and small heritability at the same time is a characteristic for being a good evolutionary algorithm for optimization.

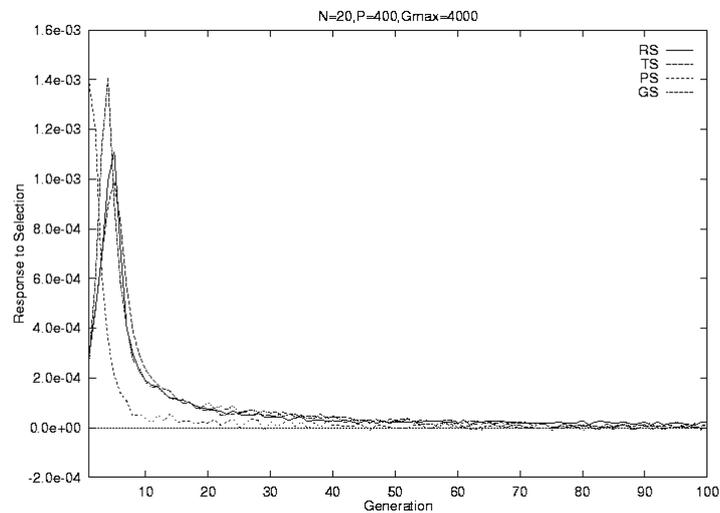


Figure 8 Response to selection curves.

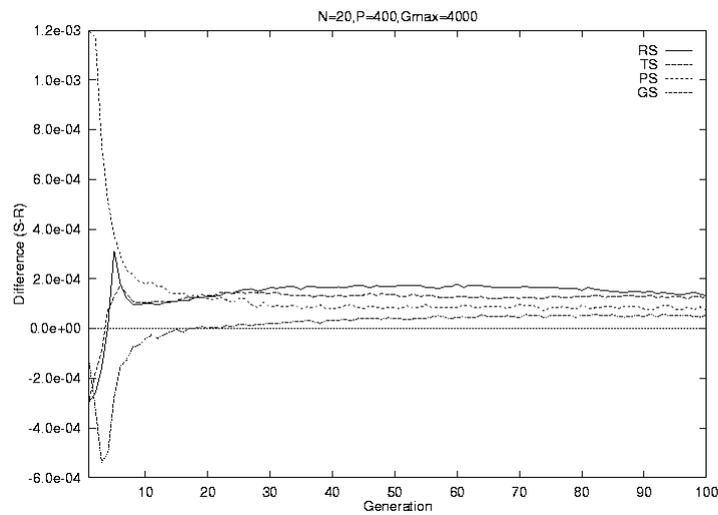


Figure 9 The $S(t)-R(t)$ curves indicating the diversity of selection methods.

6. Conclusion

A set of selection schemes has been compared in the context of real-life industrial optimization problems. We considered the combined effect of selection and other genetic operators on solution quality and convergence speed. The optimization process was considered to be converged if it either reached the maximum generation or achieved no

performance improvement in a fixed number of consecutive generations. This problem setting is more realistic than that of previous studies since practical applications require acceptable solutions within a time limit. Our experimental results show strong evidence that ranking selection and tournament selection are a better choice than proportional selection. The Genitor selection achieved sometimes comparable fitness to that of ranking and tournament selection but was usually much slower.

To see the characteristics of the selection methods we analyzed the dynamics of the evolutionary algorithms with different selection schemes by measuring the selection differential and the response to selection. It turned out that ranking selection and tournament selection had larger selection differential values but more or less similar response to selection values compared to proportional selection and Genitor selection. This suggests that large selection differential combined with small heritability is an important property for effective evolutionary optimization.

Acknowledgments

This research was supported in part by the Korea Science and Engineering Foundation (KOSEF) under grants 96-0102-13-01-3 and 981-0920-107-2.

References

- Bäck T. (1996) *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York.
- Bäck T. & Hoffmeister F. (1991) Extended selection mechanisms in genetic algorithms, In Belew, R.K. and Booker, L.B. (eds.), *Proceedings of Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 92-99.
- Baker J.E. (1985) Adaptive selection methods for genetic algorithms, In *Proceedings of International Conference on Genetic Algorithms and Their Applications*, pp. 100-111.
- Blickle T. & Thiele L. (1995) A mathematical analysis of tournament selection, In *Proceedings of Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 9-16.
- Cheng R. & Gen M. (1996) Genetic algorithms for multi-row machine layout problem, *Engineering Design and Automation*.
- Deb K. & Goldberg D.E. (1989) An investigation of niche and species formation in genetic function optimization, In *Proceedings of Third International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 42-50.
- Francis R., McGinnis L., & White J. (1992) *Facility Layout and Location: An Analytical Approach*, 2-nd ed., Prentice Hall, Englewood Cliffs, NJ.
- Gen M. & Cheng R. (1997) *Genetic Algorithms and Engineering Design*, Wiley.
- Goldberg D.E. (1990) A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing, *Complex Systems*, 4:445-460.
- Goldberg D.E. & Deb D. (1991) A comparative analysis of selection schemes used in genetic algorithms, In Rawlins, G.J.E. (ed.) *Foundations of Genetic Algorithms*, pp. 69-93.
- Grefenstette J. J. (1987) *A User's Guide to GENESIS*, Navy Center for Applied Research in Artificial Intelligence, Washington, D.C.
- Heragu S. & Kusiak A. (1988) Machine layout problem in flexible manufacturing systems, *Operations Research*, 36:258-268.
- Kim J.J. & Zhang B. T. (1998) Comparison of selection schemes for machine layout design, In *Proceedings of Asia-Pacific Conference on Simulated Evolution and Learning (SEAL-98)*, Canberra, Australia.
- Mühlenbein H. (1989) Parallel genetic algorithms, population genetics and combinatorial optimization, In *Proceedings of Third International Conference on Genetic Algorithms*, pp. 416-422.

- Mühlenbein H. & Schlierkamp-Voosen D. (1993) Predictive models for the breeder genetic algorithm: Continuous parameter optimization, *Evolutionary Computation*, 1(1):25-49.
- Schwefel H.P. (1995) *Evolution and Optimum Seeking*, Wiley, New York.
- Tam K. (1992) Genetic algorithms, function optimization, and facility layout design, *European Journal of Operations Research*, 63:322-346.
- Tate D. & Smith A. (1995) Unequal-area facility layout by genetic search, *IIE Transactions*, 27:465-472.
- Whitley D. (1989) The Genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, In *Proceedings of Third International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 116-121.
- Zhang B.T. & Mühlenbein H. (1995) Balancing accuracy and parsimony in genetic programming, *Evolutionary Computation*, 3(1):17-38.
- Zhang B.T., Ohm P. & Mühlenbein H. (1997) Evolutionary induction of sparse neural trees, *Evolutionary Computation*, 5(2):213-236.