Comparison of Selection Schemes for Machine Layout Design

Jung-Jib Kim and Byoung-Tak Zhang

Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
{jjkim, btzhang}@scai.snu.ac.kr

Abstract. Selection is an essential component of evolutionary algorithms, playing an important role especially in solving optimization problems. We compare three most popular selection schemes: proportional, ranking, and tournament selection. Experiments have been performed in the context of the machine layout design problem. This problem provides a useful benchmark due to its scalability of problem complexity and its having both discrete and continuous optimization. Our empirical results suggest that ranking and tournament selection be, in general, more effective in solution quality and computational costs for optimization than proportional selection.

Keywords: Evolutionary optimization, selection schemes, machine layout design.

1 Introduction

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

The main difference between these evolutionary algorithms lies in their representation and genetic operators [2]. 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 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. GA and GP use proportional selection. ES has developed various ranking-based selection schemes. EP is used usually with tournament selection. Characterization of the distinctive features of these methods is a prerequisite for effective application of evolutionary algorithms and for the design of new algorithms.

Goldberg and Deb [7] provide an analysis on the convergence time and growth ratio of several selection schemes. But, they considered only the effect of selection, ignoring the effect of other genetic operators such as crossover and mutation. Baeck and Hoffmeister [1] observed that small takeover time is connected with quickly decrease iin genotypic diversity even for a genetic algorithm with crossover and mutation enabled. Blickle and Thiele [3] made a mathematical analysis of tournament selection to allow a prediction of the fitness values after selection. They show that for the same selection intensity tournament selection has the smallest loss of diversity and the highest selection variance.

In this paper, we compare performances of various selection schemes in a realistic setting. The machine layout problem was chosen as a testbed for this comparative study. The algorithm halts either it reaches the maximum generation or there is no performance improvement in a fixed number of consecutive generations.

The paper is organized as follows. Section 2 defines the machine layout problem. Section 3 describes the methods in comparison. Numerical results are given in Section 4. Section 5 summarizes the results and their implication.

2 Machine Layout Design

Machine layout design (MLD) is concerned with arrangement of physical machines in a certain area. This problem can be formulated as an optimization problem in which the best facility layout is sought by optimizing some measure of performance, such as material handling costs, subject to some constraints [5]. Because of the combinatorial nature of facility layout problems, heuristic techniques are the most promising approach for solving layout problems of practical size.

Heragu and Kusiak have formulated a multiple-row machine layout problem as a two-dimensional continuous space allocation problem, which belongs to the continuous optimization approach [8]. However, in many practical problems, the machines are arranged along well-defined rows because in most cases the separation between rows can be predetermined according to the features of material handling system; that is, this problem can be viewed as discrete in one dimension and continuous in another dimension [6].

Symbol	Description
n	number of machines
m	number of rows
f_{ij}	frequency of trips between machines i and j
c_{ij}	handling cost per unit distance traveled between machines i and j
ℓ_i	length of machine i
ℓ_o	separation between two adjacent rows,
d_{ij}	minimum 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

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

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

$$z_{ik} = \begin{cases} 1, \text{ if machine } i \text{ is allocated to row } k \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

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

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

$$s.t. \quad |x_i - x_j| z_{ik} z_{jk} \geq \frac{1}{2}(\ell_i + \ell_j) + d_{ij}, \quad i,j = 1, ..., n \tag{3}$$

$$y_i = \sum_{k=1}^{m} \ell_0 (k-1) z_{ik}, \quad i = 1, ..., n \tag{4}$$

$$\sum_{k=1}^{m} z_{ik} = 1, \quad i = 1, ..., n \tag{5}$$

$$\sum_{i=1}^{n} z_{ik} < n, \quad k = 1, ..., m \tag{6}$$

$$x_i, y_i \geq 0, \quad i = 1, ..., n \tag{7}$$

$$z_{ik} = 0, 1, \quad i = 1, ..., n, \quad k = 1, ..., m \tag{8}$$

where the symbols are explained in Table 1.

The objective is to minimize the total cost involved in making the required trips between the machines. Constraint (3) ensures that no two machines 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.

Recently, several researchers have applied evolutionary algorithms to machine layout design. Tate and Smith used genetic algorithms to shape-constrained unequal area facility layout problems [11]. Cheng and Gen [4] studied genetic algorithms for solving multi-row machine layout problems.

For r-row layout problem of n machines, the chromosome a_k is represented as

$$a_k := [(s_1^k, s_2^k, ..., s_{r-1}^k), (m_{i_1}^k, m_{i_2}^k, ..., m_{i_n}^k), (\delta_{i_1}^k, \delta_{i_2}^k, ..., \delta_{i_n}^k)] \tag{9}$$

where m_j^k represents the machine in the jth position, and δ_{ik} denotes the neat clearance between machines m_{j-1} and m_j. The separators s_j^k, generally $1 < s_j^k < n$ and $s_j^k < s_{j+1}^k$, denote the cutting position to separate the list into r parts according to the r-row requirement. A machine in the jth part is assigned to the corresponding row.

3 Selection Methods

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.

The initial population is created by randomly generating the separator list, the machine permutation list, and the neat clearance list. The separator is simply a random integer within the open region $(1, n)$. A new machine list is generated by a permutation of machine symbols. 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.

The fitness of individuals is evaluated in terms of total cost and penalty to illegality:

$$f(a_k) = \frac{1}{F_k + \lambda_k K}, \quad k = 1, 2, ..., \mu \tag{10}$$

where F_k is the total cost, K is a positive penalty value, and μ is the population size. The total cost for the kth chromosome is given by

$$F_k = \min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} (|x_i^k - x_j^k| + |y_i^k - y_j^k|). \tag{11}$$

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 λ_k is calculated as follows:

$$\lambda_k = \begin{cases} 0, & \text{if } L_k^u - L \leq 0 \\ L_k^u - L, & \text{otherwise.} \end{cases} \tag{12}$$

After the fitness of chromosomes evaluated, fitter individuals are selected into the next generation. We compared three selection schemes: proportional selection, tournament selection, and ranking selection.

In proportional selection, the individuals are selected according to their relative fitness values. The selection probability of individuals is defined as

$$p_s(a_i^t) = \frac{f(a_i^t)}{\sum_{j=1}^{\mu} f(a_j^t)}. \tag{13}$$

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 [3] is performed by choosing parents randomly and reproducing the best individual from this group. When the number of parents are q, this is called the q-tournament selection. This process is repeated μ times to produce the next generation of individuals. The selection probabilities for q-tournament selection are given by

$$p_s(a_i^t) = \frac{1}{\mu^q}\left((\mu - i + 1)^q - (\mu - i)^q\right). \tag{14}$$

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 (μ, λ) uniform ranking [9], the best μ individuals are assigned a selection probability of $\frac{1}{\mu}$, while the rest are discarded:

$$p_s(a_i^t) = \begin{cases} \frac{1}{\mu}, & 1 \leq i \leq \mu \\ 0, & \mu < i \leq \lambda. \end{cases} \tag{15}$$

4 Empirical Studies

Experiments have been performed for two-row machine layout problems. The number of machines was $N = 10, 20$. Problem instances were generated at random. For each problem instance, one run was made for each selection method. For each problem size, 100 runs were made and the best and average performance were measured. Optimization terminates when 50 consecutive search steps

Method	MaxFit	AvgFit	StanDev
Proportional	*9.993*	*6.508*	*0.010060*
Tournament	*9.649*	*6.743*	*0.010409*
Ranking	*9.932*	*6.743*	*0.010154*

Table 3. *Comparison of solution qualities ($\times 10^{-2}$) for $N = 10$.*

Method	MaxFit	AvgFit	StanDev
Proportional	11.719	8.218	0.001404
Tournament	11.880	9.159	0.001021
Ranking	11.935	9.129	0.001019

Table 4. *Comparison of solution qualities ($\times 10^{-3}$) for $N = 20$.*

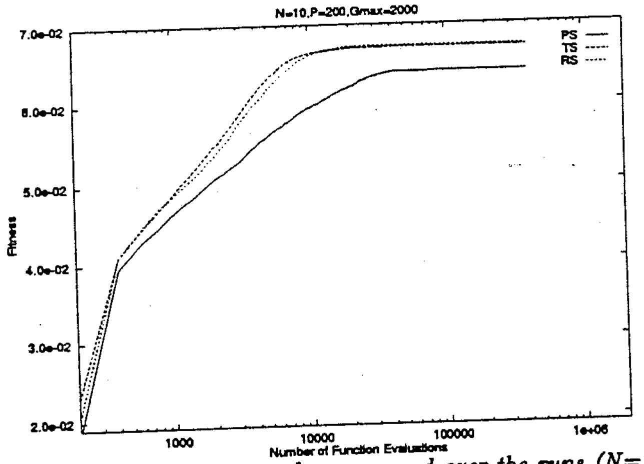

Fig. 1. *Evolution of fitness values averaged over the runs (N=10).*

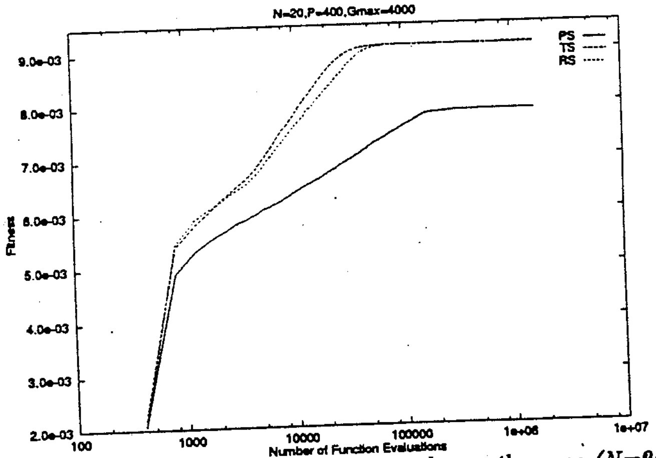

Fig. 2. *Evolution of fitness values averaged over the runs (N=20).*

achieve no improvement or when the maximum number of generations allowed is reached.

Table 2 summarizes the parameters used for the experiments. For larger size problems, we used a larger population size and a greater number of generations allowed to improve the search capability. We made some preliminary experiments to find out best parameter values for specific selection schemes. The tournament size was $q = 2$. The truncation rate for ranking selection was 0.6.

Parameter	Value
Number of machines	10, 20
Population size	200, 400
Max generation	2000, 4000
Crossover rate	0.4
Mutation rate	0.2
Selection scheme	PS, TS, RS

Table 2. *Parameter values used for experiments.*

Tables 3 and 4 summarize the results for $N = 10$ and $N = 20$, respectively. For $N = 10$, the tournament selection (TS) and the ranking selection (RS) obtained comparable performance in the average fitness which is significantly better than the proportional selection (PS). The same is true for $N = 20$.

Figures 1 and 2 compare the evolutionary curves for three methods. Shown are the fitness values averaged over 100 runs made for each method. The curves clearly show the difference in speed and quality of solutions for three selection schemes. To summarize, the tournament selection and ranking selection outperformed the proportional selection both in solution quality and optimization speed and there was, in general, no significant difference between the first two.

5 Conclusion

Three different selection schemes have been compared in the context of the machine layout problem. The performance measures used were solution quality and computational costs. The tournament selection and ranking selection outperformed, in general, the proportional selection with respect to both measures. There was no significant difference between tournament selection and ranking selection. In conclusion, tournament selection and ranking selection seem more suitable for solving optimization problems than proportional selection.

Acknowledgments

This research was supported in part by the Korea Science and Engineering Foundation (KOSEF) under grant 981-0920-350-2.

References

1. Bäck, T. and Hoffmeister, F. Extended selection mechanisms in genetic algorithms, In Belew, R.K. and Booker, L.B. (eds.), *Proc. of 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, pp. 92-99, 1991.
2. Bäck, T. *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.
3. Blickle, T. and L. Thiele, A mathematical analysis of tournament selection, In: *Proc. of Sixth Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, pp. 9-16, 1995.
4. Cheng, R. and M. Gen, Genetic algorithms for multi-row machine layout problem, *Engineering Design and Automation*, 1996.
5. Francis, R., L. McGinnis, and J. White, *Facility Layout and Location: An Analytical Approach*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ. 1992.
6. Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design*, Wiley, 1997.
7. Goldberg, D. E. and Deb, D., A comparative analysis of selection schemes used in genetic algorithms, In Rawlins, G.J.E. (ed.) *Foundations of Genetic Algorithms*, pp. 69-93, 1991.
8. Heragu, S. and A. Kusiak, Machine layout problem in flexible manufacturing systems, *Operations Research*, vol. 36, pp. 258-268, 1988.
9. Schwefel, H.-P., *Evolution and Optimum Seeking*, Wiley, New York, 1995.
10. Tam, K., Genetic algorithms, function optimization, facility layout design, *European Journal of Operations Research*, vol. 63, pp. 322-346, 1992.
11. Tate, D. and A. Smith, Unequal-area facility layout by genetic search, *IIE Transactions*, vol. 27, pp. 465-472, 1995.
12. Zhang, B. T., P. Ohm, and H. Mühlenbein, Evolutionary neural trees for modeling and predicting complex systems, *Engineering Applications of Artificial Intelligence*, 10(5): 473-483, 1997.