Multiobjective Simulated Annealing: A Comparative Study to Evolutionary Algorithms

Dongkyung Nam and Cheol Hoon Park

Abstract

As multiobjective optimization problems have many solutions, evolutionary algorithms have been widely used for complex multiobjective problems instead of simulated annealing. However, simulated annealing also has favorable characteristics in the multimodal search. We developed several simulated annealing schemes for the multiobjective optimization based on this fact. Simulated annealing and evolutionary algorithms are compared in multiobjective NK model. The preliminary results of the simulated annealing developed show that simulated annealing method performs well and sometimes better than evolutionary algorithms. More systematical analyses to the various problems are discussed as further researches.

Keywords: Multiobjective Optimization, Evolutio- nary Algorithms, Simulated Annealing, Pareto Optimality, NK model

1. Introduction

The multiobjective optimization problem has a rather different aspect to scalar-objective one. Instead of finding one global optimum, which is a general aim in scalar-objective optimization, multiobjective optimization must find a set of solutions, which is called the *Pareto set*, or *Pareto optimal frontier*, as all the Pareto solutions are equivalently important and all of them are the global optimal solutions. As many engineering and economical problems are often complex and have this multiple objectives characteristic, which must be optimized simultaneously, conventional optimization techniques, such as the steepest-descent method, conventional simplex method, many conventional evolutionary algorithms, and the simulated annealing method, have difficulties in extending themselves to the multiobjective case because they are not originally designed to find multiple solutions. Typically multiobjective problems are often solved with conventional single-objective optimization methods by using penalty or weighted sum methods [4,13,22,33,36]. However, the penalty and weighted sum methods also have difficulties in selecting proper penalty functions and weighting factors respectively. The other problem of using the weighted sum method is it cannot find a solution in a concave region [6]. To solve this problem, many researches for multiobjective optimizations have been suggested and new concepts introduced [9,10]. One of these concepts, *Pareto optimality*, is widely used in many multiobjective optimization algorithms including evolutionary algorithms.

Evolutionary algorithms (EAs) have many interesting properties and have been widely used in various optimization problems from combinatorial problems such as job shop scheduling to real valued parameter optimization [2,3]. Also many evolutionary algorithms for solving the multiobjective problem have been suggested [19,20]. The success of evolutionary approaches in multiobjective optimization is mainly based on the population concept with the ability of finding multiple optima simultaneously, which matches the idea of multiobjective optimization. However, the simulated annealing method, which is reported to give good performance a many single-objective problems, has been seldom used for the multiple objectives problems. The main reason is that simulated annealing usually finds only one solution instead of set of solutions and this is a critical handicap in multiobjective optimization [30].

There are four important properties for a good algorithm in multiobjective optimization.

1) *Searching precision.* The algorithm must find the Pareto optimal solutions, which are global optima in multiobjective optimization. When this is hard to achieve because of problem complexity, it must find the possible near solutions to the optimal solutions set.
2) *Searching time.* It must find the optimal set efficiently.
3) *Uniform probability distribution over the optimal set.* The solutions found must be widely spread, or uniformly distributed over the real Pareto optimal set instead of converging to one point because every solution is important in multiobjective optimization.
4) *Information about Pareto frontier*. The algorithm must give as much information as possible about the Pareto frontier.

Simulated annealing has been applied for multiobjective optimization by using the weight sum method in limited applications. Whidborne used the simulated annealing to solve a problem formulated as the

Corresponding Author: Cheol Hoon Park is with Department of Electrical Engineering, KAIST
373-1, Kusong-dong, Yusong-gu, Taejon 305-701,
Republic of Korea
E-mail: chpark@ee.kaist.ac.kr

method of inequalities (MOI) [38]. The objective of this paper is to construct a simulated annealing method to find all the Pareto solutions, which satisfies the above properties. First, simulated annealing method, which is suggested in this paper, uses the concepts of Pareto optimality and domination to achieve high searching precision. The main drawback of simulated annealing is searching-time, as it is generally known that simulated annealing takes long time to find the optimum. Though it is also reported that long searching-time is not always needed, the second property remains as a main problem of simulated annealing. Simulated annealing has an interesting advantage in its uniform probability distribution property as it is mathematically proved that it can find each of the global optima with the same probability in a scalar finite-state problem [12,29]. Considering that evolutionary algorithms generally use additional algorithms such as fitness sharing, niche induction for spreading the solutions, simulated annealing can have a more simple and compact structure. The last property comes from the difference between the properties of scalar-objective and multiobjective optimization. In solving scalar-objective problems, there is no need to find all the global optima except some special cases because every global optimum has the same value. The only thing needed is the optimal cost and parameters with which the cost is evaluated. However, the situation is different in multiobjective case. As all the Pareto solutions have different cost vectors that have a trade-off relationship, a human or a decision-maker must select a proper solution from the found Pareto solution set or sometimes by interpolating the found solutions.

The rest of this paper is organized as follows. Section 2 formulates the multiobjective optimization problem including the concept of Pareto optimality and domination, and describes some previous works about evolutionary algorithms and the simulated annealing method. Section 3 shows the idea of multiobjective simulated annealing method and preliminary results from it. Comparison results to an evolutionary approach are presented. Further research directions are discussed in the section 4 and section 5 summarizes the simulated annealing method in multiobjective optimization and discusses the comparisons with the evolutionary approach.

2. Survey of Stochastic Multiobjective Optimization Algorithms

2.1 Multiobjective optimizations

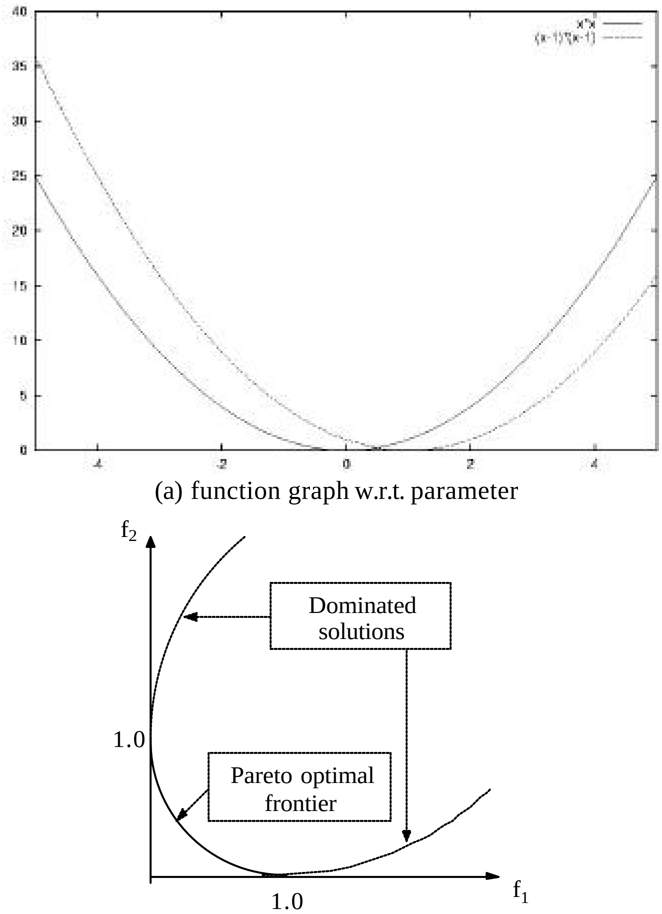
For most multiobjective problems, there exists a set of non-dominated solutions that have a trade off relationship each other, and one of the multiple objectives of each solution cannot be improved without sacrificing any of others. This concept is known as the *Pareto optimality* [29].

Definition 1 *Consider, without loss of generality, the minimization of the n components f_k, $k = 1,\ldots, n$, of a vector function **f** of a vector variable **x** in a universe* $\mathbf{A}$, *where*

$$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}),\ldots, f_n(\boldsymbol{x})).$$

Then a decision vector $\boldsymbol{x}_u \in \mathbf{A}$ *is said to be* Pareto optimal *if and only if there is no* $\boldsymbol{x}_v$ *for which* $\boldsymbol{v} = \boldsymbol{f}(\boldsymbol{x}_v) = (v_1,\ldots,v_n)$ dominates $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}_u) = (u_1,\ldots,u_n)$, *that is, there is no* $\boldsymbol{x}_v \in \mathbf{A}$ *such that*

$$v_i \le u_i \quad \forall i \in \{1,\ldots,n\} \text{ and } v_i < u_i \quad \exists i \in \{1,\ldots,n\}$$



(a) function graph w.r.t. parameter

(b) function space graph

Figure 1. The concept of the Pareto optimal frontier

The set of all Pareto-optimal decision vectors is called the Pareto optimal set, efficient set, admissible set or the Pareto frontier of the problem. The corresponding set of objective vectors is called the non-dominated set. In practice, however, it is not unusual for these two terms to be used interchangeably to describe solutions of a multiobjective optimization problem. The notion of Pareto optimality is only a first step towards the practical solution of a multiobjective problem, which usually involves the choice of a single compromise solution from the non-dominated set according to some preference information. Figure 1 shows the concept of the Pareto optimal set clearly. Considering the specified multiobjective optimization, figure 1 (a) shows each function value with respect to the parameter x and figure 1 (b) is a plot in which the x-axis is f_1 and the y-axis is f_2. The middle solid curved segment of figure 1(b) is the Pareto optimal frontier - non-dominated set and the two outer dashed curved segments are dominated solutions.

The simultaneous optimization of multiple, possibly competing, objective functions deviates from scalar-objective optimization. Instead of finding one perfect solution, multiobjective optimization problems tend to be characterized by a family of alternatives that must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others.

Therefore, the first objective in multiobjective optimization is to find the Pareto set, and the next is to select a proper solution from the found Pareto solution set.

2.2 Evolutionary algorithms

Many multiobjective optimization problems have been successfully solved using traditional mathematical optimization procedures, such as linear programming, integer programming, and nonlinear programming. However, many real-world problems involve complex and nonlinear properties that do not fit readily into one of these traditional frameworks. Recently, non-gradient, stochastic based search techniques such as simulated annealing and evolutionary algorithms have been successfully employed to solve real-world optimization problems. There have been four important multiobjective search criteria in the history of evolutionary algorithms [9].

Plain aggregating approaches: as conventional evolutionary algorithms can solve the problem when it has single objective, it is required to make scalar fitness functions on which to work. In most problems, where no global criterion directly emerges from the problem formulation, scalarization of the objective function has been achieved by aggregating the multiple objectives with weighting factors. Several applications of evolutionary algorithms in the optimization of aggregating functions have been reported from the beginning of 1990s, and almost every multiobjective problem used this method in the first era of the multiobjective evolutionary history [13,22,36]. There are two advantages of using this method. The first is, of course, the simplicity of this method. There is no need to change the algorithm itself except making a single objective by using the weighted sum method. The second is, there is no need for post-processing such as decision-making because there is only one solution already. Even though this algorithm is still widely used, the difficulty in selecting proper weights, and the inability to find solutions in a concave Pareto region are the main drawbacks of this algorithm.

Population-based non-pareto approaches: Schaffer was probably the first to recognize the possibility of exploiting populations to treat multiple, conflicting objectives separately and search for multiple non-dominated solutions concurrently in a single run [33]. In this case, his algorithm uses the concept of speciation instead of Pareto optimality. The entire population is divided into several sub-populations (speciation) and the divided sub-population was selected using a selection mechanism which considered only one objective function for each sub-population. The selected speciation makes a new population (next generation) which is divided into sub-populations again after mutation and crossover operations. Schaffer suggested this algorithm – Vector Evaluated Genetic Algorithm (VEGA) and his simulations showed good results in multiobjective optimization. As this algorithm is also very simple, uses the concept of population well, and is able to find Pareto solutions including the concave region just in one run, there have been many researches based on the population concept [11,17,25]. However, there are also weaknesses in this algorithm. One is the biasing phenomenon: final solutions have a tendency to be located on the edge of the Pareto frontier. Also the performance of this algorithm is severely affected by the objective values because selection is determined according to one of the values in the objective vector not the domination relationship.

Pareto-based approaches: Goldberg suggested a multiobjective optimization algorithm using the concept of Pareto optimality in 1989 [13]. This search algorithm, which considers all the objectives simultaneously and selects the non-dominated solutions with a high probability, can find a good Pareto optimal frontier by the Pareto ranking technique. Soon many research results about the algorithms based on the Pareto concept were published [10,19,35]. The advantage of this algorithm is its ability to find the Pareto optimal frontier indifferent to the parameter value, that is, this algorithm works well when there is large difference of average or variance between objectives. The second possible advantage of this Pareto ranking approach is that solutions that exhibit good performance in many objective dimensions are more likely to be produced by recombination.

Niche induction techniques: this algorithm uses the niche and sharing concepts to spread the searching agents uniformly over the Pareto optimal frontier. Also this method has a tendency to prohibit the genetic drift phenomenon by forcing the searching agents not to converge to one point from the beginning of the search. Though it is very helpful for the decision-making to spread out the solution uniformly, this algorithm has a weakness also. As the sharing technique is affected by the scale difference severely, spreading out the solution is generally dominated by the objective function with the largest variation. This property seems to be opposite to the philosophy of Pareto optimality and domination. Therefore, it is necessary to control the scales of each parameter before search but it is generally difficult. There have been many promising results from this algorithm by the many researchers in the 1990s [10,17,19,35].

2.3 Simulated annealing

Simulated annealing (SA) is one of the stochastic search algorithms, which is designed using a spin glass model by the Kirkpatrick [24]. It has been used in wide areas from the combinatorial problems to the real world problems because it performs well on most of optimization problems, especially on complex problems [1,26,30].

The powerfulness of SA originates in the good selection scheme and annealing technique. Generally SA used two kinds of selection scheme. One is the Metropolis algorithm and the other is the logistic selection algorithm [27]. Originally any kind of selection that satisfies the detailed balance equation can be used as a selection scheme because the detailed balance equation guarantees the convergence of SA [29]. Another reason why SA performs well is annealing, that is, the gradual temperature reducing technique. As the temperature and the cost difference mainly determine the amount of mutation in generating the next searching point, SA can do local fine-tuning towards the end of the search to give finer results. The disadvantage of SA is, as is well known, the long annealing time. There are, of course, many algorithms to compensate for this such as fast simulated annealing (FSA), very fast simulated re-annealing (VFSR), new simulated annealing (NSA) [21,37,41].

However, there is little research into using the simulated annealing method for multiobjective optimization. The first and most significant problem is that SA uses only one search agent. As solving multiobjective problem generally requires finding all the solutions at the same time, using many search agents will be effective in general. Though SA was designed originally to use only one search agent, there have been also many techniques for using multi search agents or for parallelization [1]. To use a population in SA, however, has a possibility to lose the merits of SA a little because those kinds of methods usually entail redundant search.

3. Multiobjective Simulated Annealing

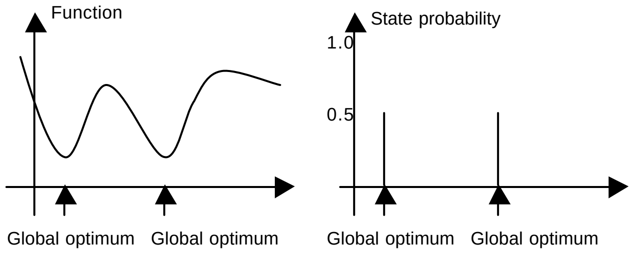
EAs have been widely used in various static optimization problems from combinatorial optimization to real parameter optimization as a powerful and robust optimization technique. There have been a lot of researches showing that EAs are good optimization methods, which has resulted in fast enlargement of their application areas [7,8,18,32,34]. Many EA researchers have been trying to characterize EAs' mechanisms and landscape. One result of this research is simulation results with Royal road functions [28]. Though the Royal road function was designed in favor of crossover operations, evolutionary search do not always outperform variations of the hill climbing method and a well-designed hill climbing method shows better performance than evolutionary algorithms. In 1995, Wolpert and Macready published the No Free Lunch theorem and the theorem showed mathematically that all algorithms perform equally well over all the functions in the finite search space [39,40]. According to this theorem, discussion about the performance between different algorithms can be meaningless as they perform equally from an average point of view. However, the situation is different in treating real world problems, as there are general tendencies in ordinary problems. Droste, Jansen and Wegener showed that a particular algorithm performs better over a subset of the entire function set in their paper [5]. It means that there can be a better algorithm to solve restricted problems.

Many researchers also have found that EAs are very promising algorithms for solving multiobjective optimization problem as they can find many good solutions (the Pareto set) in one simulation. However the SA algorithm has been hardly used for multiobjective optimization because SA was originally constructed to use only one searching agent. This is known to be a critical weakness of SA as it betrays the philosophy of multiobjective optimization – searching for all the Pareto solutions instead of only one solution. As the result of this weakness, SA has remained as one of the improper or not favorable algorithms for multiobjective optimization.

It is, however, a question whether SA cannot be used at all for multiobjective optimization though it performs well and sometimes better than EAs in solving single objective optimization problems [31]. In this section, a possible method for SA is suggested to solve multiobjective optimization and its advantages and disadvantages are shown by simulation results.

3.1 Extension from Single-objective to Multiobjective

Multiobjective SA (MOSA) uses the domination concept and the annealing scheme for efficient search. The main obstacle for SA in multiobjective optimization is its inability to find multiple solutions. However, SA can do the same work by repeating the trials as it converges to the global optima with a uniform probability distribution in the single objective optimization. Figure 2 shows this characteristic of SA. When there are two global optima, it is proved that SA can find each optimum with probability 0.5 [29]. When this fact is also true in multiobjective optimization, SA has advantages over EAs because it does not need large memory to keep the population; nor does it use additional algorithms to spread the solutions over the Pareto frontier. Additionally MOSA can find a small group of Pareto solutions in a short time with the demand of urgent simulation and then find more solutions by repeating the trials for detailed information about the Pareto frontier. The mathematical tasks of showing the uniform convergence to the Pareto frontier of MOSA is not completed and remains as a future work. In this paper, simulation results on a simple test-bed will be presented to show this property.



(a) (b)

Figure 2. Uniform distribution property in SA

a) the graph of the objective function; b) State probability of the Markov chain as time goes to infinity. As there are only two global optima, SA finds each global optimum with the same probability 0.5

```
s=s0
T=T0
Repeat
    Generate a neighbor s'=N(s)
    If C(s') dominates C(s)
            move to s'
    else if C(s) dominates C(s')
            move to s' with transition probability
         Pt(C(s), C(s'), T)
    else if C(s) and C(s') do not dominate each other
            move to s'
    endif
    T=annealing(T)
Endrepeat (until the termination are satisfied)
```

Figure 3. Pseudo-code of multiobjective simulated annealing

General scheme:

The general SA algorithm involves the following three steps. First, the objective function corresponding to the energy function must be identified. Second, one must select a proper annealing scheme consisting of decreasing temperature with increasing of iterations. Third, a method of generating a neighbor near the current search position is needed. In single objective optimization problems, the transition probability scheme is generally selected by the Metropolis and logistic algorithms [27,29]. However, the situation is different in multiobjective optimization and choosing a proper transition probability is difficult. This problem will be treated detail in the transition probability paragraph. The algorithmic description of the MOSA is outlined in figure 3 where s represents the current search position (or the current state in a finite state search problem) and T is the temperature parameter, which is gradually decreased as time goes on. A new search position s' is generated by the N(s) function, its cost is evaluated and compared with the previous cost. When it is determined to be a good solution by the domination test, the new state is accepted. Even when the new position is not proper (meaning the new position is dominated by the current state), it is accepted with some acceptance probability. When there is no superiority between the current state and the next state, the new state is accepted instead of the current one because moving in the non-dominated situation helps increase the spread performance and evade local optima. This fact will be shown with simulation results. When whether to move or stay is determined, the algorithm repeats its loop with lower temperature until termination conditions are satisfied.

Neighbor generating and annealing:

In finite state problems like combinatorial problems (TSP, QAP, NK-model), a general neighbor generating method is the permute operation (or the bit flip operation in a binary problem), which must satisfiy the reachability and symmetry conditions. The following geometric cooling is widely employed for the annealing scheme in this kind of problem.

$$T_k = a^k T_0 \tag{1}$$

where $0 < a < 1$ is a cooling rate. For combinatorial problems (including the NK-model), it is usual to generate a neighbor by flipping one bit at a random position and use the geometric annealing scheme.

Transition probability:

General transition rules such as the Metropolis or logistic method cannot be applied directly to the multiobjective problems because they support only a scalar cost function. The suggested transition rule in this paper is very similar to the Metropolis method except that they used a different cost criterion for the multiobjective cost function. The transition probability from state i to j is,

$$P_t(i, j) = \min\{\exp(-c(i, j)/T), 0\} \tag{2}$$

where $c(i, j)$ is the cost criterion for transition from state i to j, and T is the annealing temperature.

Six criteria for MOSA are suggested and evaluated. The schemes are as follows:

Minimum cost criterion

$$c(i, j) = \min_k (c_k(j) - c_k(i)) \tag{3}$$

where $c_k(i)$ is k^{th} cost value in the objective vector of i^{th} state.

Maximum cost criterion

$$c(i, j) = \max_k (c_k(j) - c_k(i)) \tag{4}$$

Random cost criterion

$$c(i, j) = \sum_{k=1}^{D} a_k (c_k(j) - c_k(i)) \tag{5}$$

where D is the dimension of the objective vector and a_k is a random variable with uniform probability distribution.

Self cost criterion

$$c(i, j) = \sum_{k=1}^{D} c_k(i) \tag{6}$$

Average cost criterion

$$c(i, j) = \frac{\sum_{k=1}^{D} (c_k(j) - c_k(i))}{D} \tag{7}$$

Fixed cost criterion

$$c(i, j) = \text{fixed value} \tag{8}$$

We tested the above six criteria on the simple test-beds and found that the random, average, fixed criteria generally show good performance. The performances of the minimum, maximum, self cost criteria change greatly dependent on the test-beds. In the following simulations, we used the average cost criterion. The main problem with using the weighted sum method – the inability to find a concave region – does not occur in the suggested MOSA as it uses the domination test first.

Move or Stay in non-dominated situation:

When the new state is the same level of value as the current state, there can exist two schemes – move to the new state or stay in the current state. The analysis of this problem shows that the move scheme is better than the stay one. With the stay scheme, search will end on both edges of the Pareto frontier not entering the middle of the frontier. However, with the move scheme, search will be continue into the middle part of the frontier, move freely between non-dominated states like a random walk when the temperature is low and eventually will be distributed uniformly over the Pareto frontier as time goes to infinity.

3.2 EA techniques – the niche induction algorithm

The specifics of the Niche Pareto algorithm are localized to the implementation of selection - the use of Pareto domination tournaments, where two candidates for selection are compared against each individual in the comparison set. In tournament selection a set of individuals is randomly chosen from the current population and the best of this subset is chosen to be represented in the next population. In order to obtain a Pareto optimal surface, tournament selection must be altered to use multiple objectives. Selection pressure is mainly determined by the size of the comparison set t_{dom}; if the size of the comparison set is large, there is high selection pressure which possibly lead to local optima in many cases, if the size is small, there is low pressure which make the convergence of population slow. The Pareto rank is the number of elements in the comparison set dominated by the candidate. For example, if the candidate dominates three elements of the comparison set of size ten, the Pareto rank of the candidate is 3 [13]. Although the Pareto rank scheme encourages the exploration in the direction of non-dominated individuals, they have a tendency to converge on one point as time goes on and will suffer from population drift because this is the property of most conventional evolutionary algorithms. To find uniformly distributed solutions along the Pareto frontier, Goldberg and Richardson, Deb, Goldberg have incorporated fitness sharing method by the niche scheme [14,15,16]. Fitness sharing degrades the individual fitness by a sharing function as f_i / m_i, dividing the objective fitness f_i, by the niche count m_i, which reflects the neighborhood crowding around an individual. In this paper, this scheme was adopted in a simplified form. Instead of recalculating the fitness function by the sharing method, the niche count of one candidate is directly compared to the niche count of the other. As the sharing function encourages the candidate that is located in the sparse space, gradually all the solutions of the population become uniformly distributed as the algorithm goes on.

3.3 Comparison on the NK fitness model

In this section we discuss whether or not simulated annealing is a promising tool for solving to solve hard optimization problems by comparing its performance with evolutionary algorithms on the multidimensional version of Kauffman's NK fitness landscape model [23]. The NK-model of fitness landscapes can be regarded as combinatorial optimization problems defined on the binary space $\{0,1\}^N$, where N is the length of binary string. The fitness function, $f:\{0,1\}^N \rightarrow R$ is defined by the average of fitness contributions of all bits as shown in equation (9)

$$f = \frac{1}{N}\sum_{i=1}^{N} f_i \tag{9}$$

where the fitness contribution f_i of the *i*-th bit is determined by a random number drawn from uniform distribution in the interval [0,1], depending on the values of itself and K other bits. That is, f_i has $2^{(K+1)}$ different random numbers. In the NK model, K is the most important parameter that influences the statistical property of the NK fitness landscape. K is used to tune the ruggedness of the landscape. For example, when K = 0, the landscape has a unique global optimum but as K increases (up to N-1), it becomes more rugged with an increasing number of local optima. As the NK model was originally designed to construct to single objective landscape, we extend it to a multiobjective one. In multiobjective NK model (NKD model), there is one more parameter D that determines the dimension size of multiple objectives.

In what follows, the parameters of simulated annealing are described:

Initial temperature value: The initial temperature is chosen to be 500 by heuristics from simulation results.

Annealing scheme: For the NKD model, the geometric annealing method is used. $T_{k+1} = a T_k$ $(a = 0.995)$.

Neighbor generation: Generating a new search position is done by flipping one bit of parameter string.

Chain length: The chain length represents the number of allowable transitions before the temperature changes its value. The length of the string (N) is used for the chain length.

Termination condition: The algorithm finished its calculation after pre-defined iteration. In this simulation, we set its value to 5000.

Agent number: We used 100 independent searching agents simultaneously. That is, 100 agents search for the Pareto optimal without exchanging of information between them.

The parameters for the evolutionary algorithm are as follows:

Population size: The population size is set to 100 for all the simulations.

Genetic operation: Conventional one point crossover is used with crossover probability 0.1 and standard mutation per bit is used with the mutation rate 0.3.

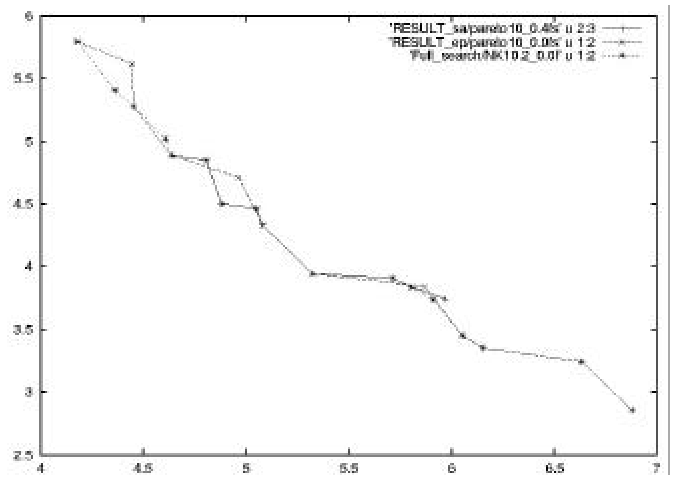
Selection: Pareto based tournament selection is used with a comparison set size of 5 (5% of the population). As this parameter determines the selection pressure, it must be chosen carefully. However, as there is no systematic method for choosing this by considering the landscape, this value is chosen based on the empirical simulation results. It is an open problem to choose a proper selection power according to each problem and remains as future work.

Niche size: The niche size that determines the size of the hyper sphere around the candidate is set to 0.1.

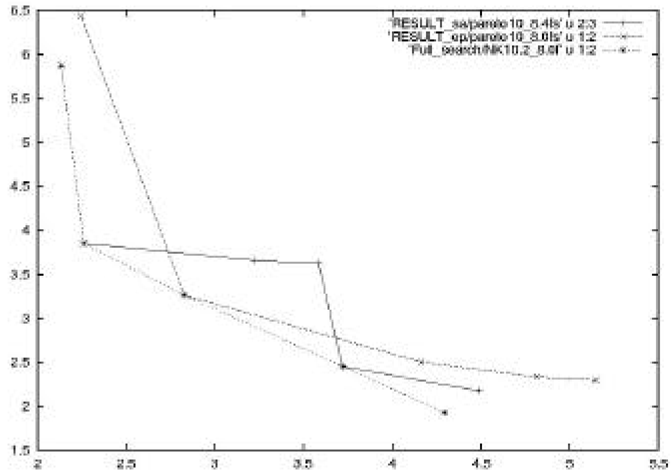
Termination condition: The evolutionary algorithm was designed to use the same number of iterations for comparison with simulated annealing. The simulation runs over 5000 iterations.

The first four tests are conducted on the small size landscape model where the exact Pareto frontier can be found by exhausitive search. The length of NKD model is 10, the epistatic parameter K changes its value to 0, 2, 4, 8 and the objective dimension is 2. The figure 4 shows a (pseudo) Pareto optimal frontier that each algorithm found. As it is difficult to show the performance of multiobjective optimization except by showing the Pareto frontier, a randomly chosen typical graph is presented as an example. In the small size landscape model, the simulated annealing method can find the Pareto frontier more precisely at many times, and each solution spreads widely over the Pareto frontier in spite of the fact that the simulated annealing method does not use any sharing method. However, it is true that the evolutionary approach with the sharing technique has a tendency of spreading more than simulated annealing. We can see this tendency more easily with the large size landscape model.

We conducted the second tests to examine the performance of the two algorithms in a large landscape. By changing N to 20, 40, 80 with K, to 2, 8, the simulated annealing and evolutionary algorithms have been simulated and compared. Figure 5 shows the comparison results of simulation. The conclusion is that the evolutionary algorithm shows better performance as the size of landscape becomes large with better searching ability and better spreading characteristics. However, simulated annealing also showed satisfactory results from another point of view when considering that simulated annealing does not use any additional algorithm and it can be used independently.



(a) N=10, K=0, D=2 (Low epistatic problem)



(b) N=10, K=8, D=2 (Highly epistatic problem)

Figure 4. Simulation results on the small size NKD-model

4. Discussion and Future Work

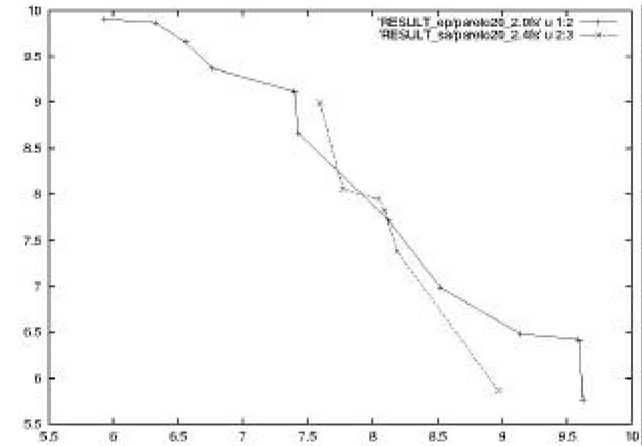
4.1 Niche induction simulated annealing

Though the suggested simulated annealing method gives satisfactory simulation results in multiobjective optimization over a finite state optimization, the NKD model, it is sometimes observed that simulated annealing has difficulty in searching the Pareto optimal with uniform distribution. That is, multiobjective simulated annealing can find the solutions in the easier and non-complex problem, but the performance is degraded in complex problems with much randomness like highly epistatic NKD models. One possible approach for increasing the performance of simulated annealing is to use the population information efficiently. The niche induction method was reported as a powerful technique in multiobjective evolutionary algorithms [19]. However, using the information of the population like niche induction should be designed carefully because it may harm the advantages of the simulated annealing method.

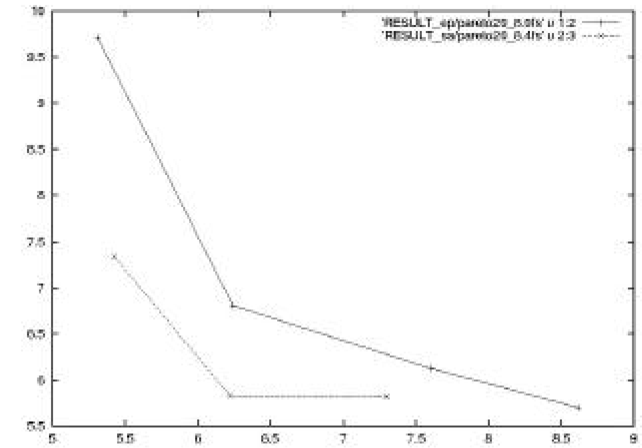
4.2 Performance measures for multiobjective optimization

One difficulty in comparing the algorithms in the multiobjective test-beds is that there is no systematic criterion to measure the performance of each algorithm. This is mainly due to the fact that in multiobjective optimization, the objective value itself does not have a significant meaning. Instead, the configuration of objective values is more important. Therefore, the conventional measure is only the plotting of the Pareto set, but it is impossible to draw the graph when the dimensions of objectives are larger than three. (Even for three-dimensional graph it is not so easy to determine which is the better Pareto set) Even if it is possible to plot the graph for more than three objectives, it is not a good measure as there is no quantitative information. A good performance measure for comparison must have the following properties.

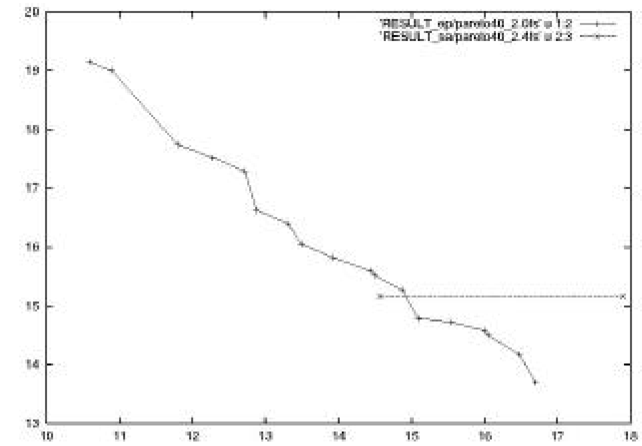
1) It must measure the closeness to the real Pareto frontier in numeric value.
2) The uniformity of the distribution of solutions over the Pareto frontier must be measured.
3) Additional information, e.g. separated frontiers number, must be also measured.



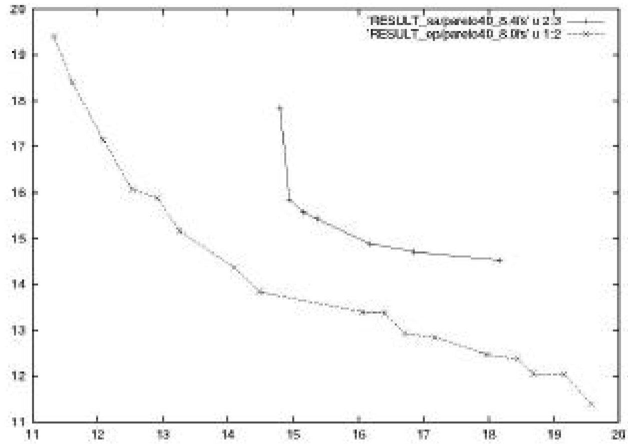
a) N=20, K=2, D=2



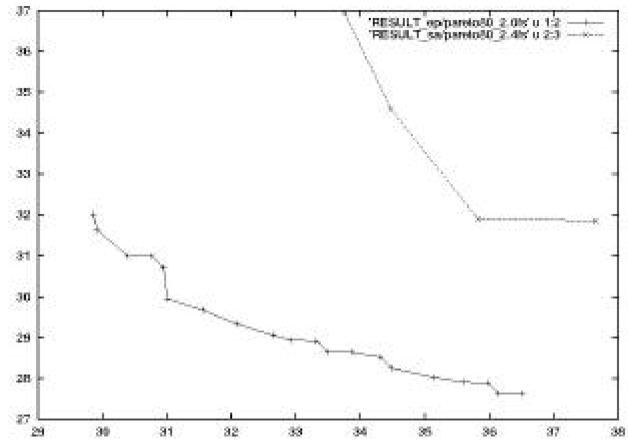
b) N=20, K=8, D=2



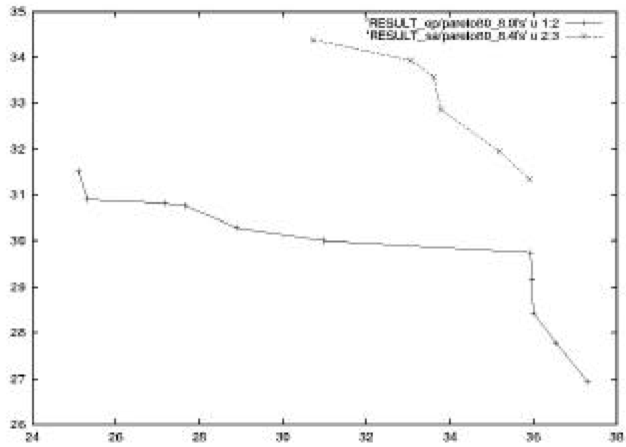
c) N=40, K=2, D=2



d) N=40, K=8, D=2



e) N=80, K=2, D=2



f) N=80, K=8, D=2

Figure 5. Simulation results on the large size NKD-model

4.3 Mathematical Analysis

The most favorable property of simulated annealing is that there is a complete convergence proof for it. The ideal annealing and neighbor generating schemes are deduced from the mathematical analysis of convergence. Even though these ideal schemes have little meaning from a practical point of view, for example, the conventional simulated annealing uses the log-like annealing scheme from the mathematical result and it takes enormous simulation time for the algorithm to converge, guaranteeing the convergence is a fundamental step for constructing an algorithm. Unfortunately, mathematical analyses about simulated annealing in multiobjective optimization have seldom been studied and remain as an open problem.

There are two main properties to be considered mathematically: one is the convergence proof and the other is uniformity of distribution. As the conventional simulated annealing method satisfies the detailed balance condition, it is guaranteed to have pseudo-stationary probability and the global convergence probability is represented as a simple equation [29]. However, in the multiobjective case, finding a proper acceptance probability criterion, which satisfies the detailed balance condition, is difficult. Therefore, even though the pseudo-stationary probability exists, it is not easy to find the probability as an equation form.

It is also unclear whether the independent simulated annealing algorithm gives uniformly distributed solutions over the Pareto frontier or not. Though proving uniform distribution over the connected Pareto set is clear and easily explained by the random walk property, the problem is not so easy when the Pareto set is not a connected one. This problem also remains as further work.

5. Conclusion

There have been many researches into using evolutionary algorithms to solve multiobjective problems and many efficient algorithms have been developed. However, though simulated annealing is also a very powerful searching algorithm and has given many good results in various optimization fields, it has been seldom used for the multiobjective optimization because it conventionally uses only one search agent, which makes the search for all solutions in the Pareto set difficult.

With the idea that simulated annealing has a uniform probability distribution over global optima, a multiobjective simulated annealing method is suggested. The preliminary results of the developed algorithms are compared with an evolutionary algorithm and show that simulated annealing also has good properties in multiobjective optimization. The first test with finite state test-beds shows that independent simulated annealing have a tendency of finding the solutions in the Pareto set with uniform probability. This property was tested over a more complex combinatorial problem – the multidimensional NK model. When the size of problem is small, simulated annealing showed good performance compared to the evolutionary algorithm. However, the evolutionary algorithm outperforms simulated annealing when the problem size and the epistatsis become large. Experimental results suggest that simulated annealing has much potential in the multiobjective optimization field also. Parallelizing techniques and using population information will be good approaches for increasing the performance of MOSA. Also, finding efficient parallelizing techniques and performance measures for multiobjective optimization remains as future work.

References

[1] R. Azencott, *Simulated Annealing: Parallelization Techniques*, Reading, John-Wiley & Sons, Inc., 1992.

[2] T. Back, U. Hammel and H.-P. Schwefel, "Evolutionary Computation: Comments on the History and Current State," *IEEE Trans. on Evolutionary Computation*, pp.3-17, 1997.

[3] T. Back, F. Hoffmeister, and H. Schwefel, "A Survey of Evolutionary Strategies," In *Proceedings of the Fourth ICGA*. pp.1-10, Morgan Kauffman, San Mateo, CA, 1993.

[4] L. Davis, and M. Steenstrup, "Genetic Algorithms and Simulated Annealing: an Overview," In L. Davis (Ed.) *Genetic Algorithms and Simulated Annealing, Research Notes in Artificial Intelligence*, pp.1-11, London:Pitman.

[5] S. Droste, T. Hansen, and I. Wegener, "Perhaps not a Free Lunch but at least a Free Appetizer," *Proceedings of the Genetic and Evolutionary Computations Conference*, pp.833-839, 1999.

[6] P. J. Fleming, and A. P. Pashkevich, "Computer Aided Control System Design Using a Multiobjective Optimization Approach," *Proceedings of the IEE Control'85 Conference*, pp.174-179, London, UK: IEE, 1985.

[7] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. on Neural Networks*, vol.5, pp.3-14, 1994.

[8] D. B. Fogel, *Evolutionary Computation*, IEEE Press, 1995.

[9] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiojective Optimization," *Evolutionary Computation*, vol.3, no.1, pp.1-16, 1995.

[10] C. M. Fonseca and P. J. Fleming, "Genetic Algorithm for Multiobjective Optimization: Formulation, Discussion and Generalization," *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp.416-423, 1993.

[11] M. P. Fourman, "Compaction of Symbolic Layout Using Genetic Algorithms," In J. J. Grefenstette (Ed.), Genetic Algorithms and Their Applications: *Proceedings of the First International Conference*

on Genetic Algorithms, pp.141-153, Hillsdale, NJ:Lawrence Erlbaum, 1985.

[12] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribtions, and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.PAMI-6, no.6, November 1984.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[14] D. E. Goldberg, K. Deb, and J. Horn, "Massive Multimodality, Deception, and Genetic Algorithms," *IlliGAL Report No. 92005*. Urbana-Champaign, IL: Department of General Engineering, University of Illinois at Urbana-Champaign, 1992.

[15] D. E. Goldberg, K. Deb, and J. Horn, "Massive Multimodality, Deception, and Genetic Algorithms," *Parallel Problem Solving From Nature*, vol.2, pp.37-46, 1992.

[16] D. E. Goldberg and J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization," *Proceedings of the Second International Conference on Genetic Algorithms*, pp.41-49, 1987.

[17] P. Hajela, and C. -Y. Lin, "Genetic Search Strategies in Multicriterion Optimal Design," *Structural Optimization*, vol.4, pp.99-107, 1992.

[18] J. Holland, *Adaptation in Natural and Artificial systems*. MI: University of Michigan Press, 1975.

[19] J. Horn and N. Nafpliotis, "Multiobjective Optimization Using the Niched Pareto Genetic Algorithm," *IlliGAL Report No. 93005*. Urbana-Champaign, IL: Department of General Engineering, University of Illinois at Urbana-Champaign, 1993.

[20] J. Horn, N. Nafpliotis and D. E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," In *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, pp.82-87, Piscataway, NJ. IEEE Service Center, 1994.

[21] L. Ingber, "Very Fast Simulated Re-annealing," *Mathematical and Computer Modelling*, vol.16, no.11, pp.87-100, 1992.

[22] G. Jones, R. D. Brown, D. E. Clark, P. Willet and R. C. Glen, "Searching Databases of Two-Dimensional and Three-Dimensional Chemical Structures Using Gentic Algorithms," In S. Forrest (Ed.) Genetic Algorithms: *Proceedings of the Fifth International Conference*, pp.597-602, San Mateo, CA:Morgan Kaufmann, 1993.

[23] S. Kauffman, "Adaptation on Rugged Fitness Landscapes," in *Lecture in the sciences of complexity* (D. L. Stein, ed.), pp.527-618.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol.220, pp.671-680, 1983.

[25] F. Kursawe, "A Variant of Evolution Strategies for Vector Optimization," In H,-P, Schewefel and R. Männer (Eds.), *Parallel problem solving from nature*, 1st workshop, proceedings, volume 496 of *Lecture notes in computer science*, pp.193-197, Berlin: Springer-Verlag, 1991.

[26] P. J. M. V. Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands, D. Reidel, 1987.

[27] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbloth, A. H. Teller, and E. Teller, "Equation of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, vol.21, pp.1087-1092, 1953.

[28] M. Mitchell, *An Introduction to Genetic Algorithms*, Reading: MIT press, 1996.

[29] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, "Convergence and Finite-time Behavior of Simulated Annealing," *Advanced Applied Probability*, vol.18, pp.747-771, 1986.

[30] L. J. Park, "Hybrid Evolutionary Algorithms with Heuristic Operators for Combinatorial Optimization Problems," *Ph. D Thesis*, 1997.

[31] L. J. Park, C. H. Park, C. Park, and T. Lee, "Application of Genetic Algorithms to Parameter Estimation of Bioprocesses," *Med. & Biol. Eng. & Comput.*, vol.35, pp.1-3, Jan. 1997.

[32] I. Rechenberg, *Evolutions Strategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*. Stuttgart Frommann-Holzboog, 1973.

[33] J. D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," In J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pp.93-100. Lawrence Earlbaum, 1991.

[34] J. D. Schaffer, D. Whiteley, and L. J. Eshelman, "Combination of Genetic Algorithms and Neural Networks: A Survey of the State of Art," *Proceeding of COGANN-92*, pp.1-37, 1992.

[35] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computations*, vol.2, no.3, pp.221-248, 1995.

[36] G. Sysweda, and J. Palmucci, "The Application of Genetic Algorithms to Resource Scheduling," In R. K. Belew and L. B. Booker (Eds.) Genetic Algorithms: *Proceedings of the Fourth International Conference*, pp.502-508, San Mateo, CA:Morgan Kaufmann, 1991.

[37] H. H. Szu, and R. L. Hartley, "Fast Simulated Annealing," *Physics Letters A*, vol.122, pp.157-162, 1987.

[38] J. F. Whidborne, D.-W. Gu, and I. Postlethwaite, "Simulated Annealing for Multiobjective Control System Design," *IEE Proceedings of Control Theory Application*, vol.144, no.6, November 1997.

[39] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Search," *Technical Report SFI-TR-95-02-010*, Santa Fe Institute, Santa Fe, 1995.

[40] D. H. Wolpert and W. G. Macready, “No Free Lunch Theorems for Optimization,” *IEEE Trans. on Evolutionary Computation*, vol.1, no.1, pp.67-82, 1997.
[41] X. Yao, “A New Simulated Annealing Algorithm,” *International Journal of Computer Mathematics*, vol.56, pp.161-168, 1995.

Dongkyung Nam received the B.S. degree in 1994, and M.S. degree in 1996 from the Seoul National University, Seoul, Korea, all in electrical engineering.

He is currently pursuing the Ph.D. degree in Electrical Engineering at Korea Advanced Institute of Science and Technology. His research interests include evolutionary algorithm, artificial neural network, artificial life, and multiobjective optimization.

Cheol Hoon Park He received the B.S. degree in Electronics Engineering with the best student award from Seoul National University, Seoul, Korea, in 1984 and the M.S. and Ph.D. degrees in Electrical Engineering from California Institute of Technology, Pasadena, California, in 1985 and 1990, respectively. He joined the Department of Electrical Engineering at the Korea Advanced Institute of Science and Technology in 1991, where he is currently an Associate Professor. His current research interests are in the area of intelligent systems including intelligence, neural networks, fuzzy logic, evolutionary algorithms, and application to recognition, information processing, intelligent control, and optimization. He is a member of IEEE, INNS, and KITE.