Journal of Global Optimization **23:** 171–190, 2002.

A novel metaheuristics approach for continuous global optimization

THEODORE B. TRAFALIS* and SUAT KASAP
School of Industrial Engineering, University of Oklahoma, Norman, Oklahoma 73019, USA
**Corresponding author. e-mail: ttrafalis@ ou.edu. e-mail: suat@ou.edu.*

Abstract. This paper proposes a novel metaheuristics approach to find the global optimum of continuous global optimization problems with box constraints. This approach combines the characteristics of modern metaheuristics such as scatter search (SS), genetic algorithms (GAs), and tabu search (TS) and named as hybrid scatter genetic tabu (HSGT) search. The development of the HSGT search, parameter settings, experimentation, and efficiency of the HSGT search are discussed. The HSGT has been tested against a simulated annealing algorithm, a GA under the name GENOCOP, and a modified version of a hybrid scatter genetic (HSG) search by using 19 well known test functions. Applications to Neural Network training are also examined. From the computational results, the HSGT search proved to be quite effective in identifying the global optimum solution which makes the HSGT search a promising approach to solve the general nonlinear optimization problem.

Key words: Genetic algorithms, global optimization, metaheuristics, scatter search, tabu search.

1. Introduction

In this paper the following problem known as the *global optimization* problem is considered: Find a point x^* such that $f(x^*) \leqslant f(x) \forall x \in D$, where D is a convex set more specifically a polytope that is defined by box constraints in R^n.

The solution methodologies to solve the above problem are classified into two categories as deterministic and stochastic methods respectively. Floudas and Pardalos (1992), Horst et al. (1995), Reeves (1993), Rinnooy et al. (1989), and Törn and Zilinskas (1989) provided excellent surveys of those methods. Some of these methods have been proposed by Becker and Lago (1970), Branin (1972), Dixon and Szegö (1978a, 1978b), Garcia and Gould (1980), Goldstein and Price (1971), Price (1978), Shubert (1972), and Törn (1978). Recently, modern metaheuristics have been proposed such as simulated annealing (Corana et al., 1987; Goffe et al., 1994; Kirkpatrick et al., 1983), genetic algorithms (GAs) (Androulakis and Venkatasubramanian, 1991; Goldberg, 1989; Holland, 1992), evolutionary algorithms by Michalewicz (1996a, 1996b), tabu search (TS) (Al-Sultan and Al-Fawzan, 1997; Battiti and Tecchiolli, 1994, 1996; Cvijovic and Klinowski, 1995; Glover and Laguna, 1993) and scatter search (SS) (Fleurent et al., 1995; Glover, 1994b, 1995; Trafalis and Al-Harkan, 1995).

Deterministic methods attempt to generate trajectories that eventually converge to points which satisfy the criteria of local optimality. They are beneficial only when the starting point belongs to the region of attraction of the global optimum. This infers that any deterministic method could be attracted by the local optimum instead. Stochastic methods attempt to reasonably cover the whole search space, so that all local and global optima are identified. The main difference between deterministic and stochastic algorithms is that in the stochastic methods, points that do not strictly improve the objective function can also be created and take part in the search process.

The main objective of this paper is to apply tabu search and scatter search in continuous optimization problems. Specifically, we modify and extend an evolutionary approach that was proposed by Androulakis and Venkatasubramanian (1991) and Trafalis and Al-Harkan (1995 and forthcoming) by introducing notion of memory that is originated from TS. Our approach, a hybrid scatter genetic tabu (HSGT) search, can be considered as a hybrid approach which attempts to find the global optimum of a general nonlinear function. The proposed HSGT search combines the characteristics of metaheuristics such as scatter search, genetic algorithms, and tabu search. The proposed HSGT search starts with a randomly generated starting point and search directions to construct a collection of solutions. Then, it computes the weighted center of gravity using these solutions and the weight assigned to each solution. Next, a new weighted center of gravity is accepted or rejected according to its tabu status. Subsequently, a new set of search directions using the old search directions are generated either randomly or using the GA operators. If the new center of gravity is tabu, then the new directions are randomly generated. Otherwise, the GA operators are used to construct the new search directions. At this stage, a complete iteration of the HSGT search is performed. The procedure is repeated until the stopping criterion is satisfied. Then, the final weighted center of gravity is the solution to the problem.

The organization of this paper is as follows. In the following section, descriptions of the basic building blocks of the HSGT approach are given. A full description and implementation of the HSGT search is presented in Sections 3 and 4, respectively. The computational results are given in Section 5. An application to neural networks training is examined in Section 6. The conclusion and recommendations are given in Section 7. In Appendix A, the 19 test functions which were used for testing the proposed HSGT search are given.

2. Description of basic building blocks of the HSGT search

Some characteristics of modern metaheuristics such as SS, GA, and TS are combined in order to design the proposed HSGT search for continuous global optimization problems. In this section, methods that are the building blocks of the HSGT search will be reviewed to give a better understanding. The following sections will be based on the following methods.

```
Begin
    initialize a population of chromosomes
    evaluate each chromosome
    while (not termination-condition) do
        select chromosomes for reproduction
        implement GA operators
        evaluate new chromosomes
        replace the old chromosomes with the new
        chromosomes
    end
end
```

Figure 1. The general genetic algorithm.

2.1. GENETIC ALGORITHMS (GA)

This subsection is devoted to describing the GAs which were developed by Holland in 1975. The theory and the application of the GAs have been reported by several researchers (Davis, 1991; Goldberg, 1989; Holland, 1992; Michalewicz, 1996a, 1996b; Srinivas and Patnaik, 1994). In these studies, the GAs were shown to be successfully applied to several optimization problems. For example, they have been applied to routing, scheduling, adaptive control, game playing, cognitive modeling, transportation problems, traveling salesman problems, optimal control problems, and database query optimization among others. Since GAs are adaptive and flexible, they have attracted several researchers from different fields such as computer science and engineering, operations research, business, and social science.

The GAs are stochastic search techniques whose search algorithms simulate natural phenomena (biological evolution). One of the strengths of GAs is that they use past information to direct their search with the assumption of improved performance. The general GA procedure is given in Figure 1 and can be described as follows: a population of binary or non-binary chromosomes is initialized, then, each chromosome is evaluated using the fitness function. Next, a set of chromosomes is selected to reproduce new chromosomes. The production process is accomplished by applying the genetic operators (crossover and mutation) on the chromosomes selected. Then, each new chromosome is evaluated to complete one iteration.

2.2. SCATTER SEARCH (SS)

The SS approach was introduced by Glover (1977). Originally it was introduced as a heuristic to obtain a near optimal solution to an integer programming problem. Also, it was used to generate both starting solutions and trial solutions. Recently the SS approach was refined and used for both discrete and continuous optimization problems (Fleurent et al., 1995; Glover, 1994b, 1995). The SS approach is given in Figure 2 and it is described as follows.

```
begin
    while (not termination-condition) do
        generate a set of reference points
        generate a weighted center of gravity for the reference
        points
        record the function evaluated
    end
end
```

Figure 2. The general scatter search approach.

The SS generates sequences of coordinated initializations, which are performed to ensure the exploration of the various parts of the solution space. The SS begins with a set of reference points which can be obtained by applying either heuristic procedures or random methods. Then, a weighted center of gravity of the reference points is determined using a linear combination of the reference points and their weights. Next, subsets of the initial reference points and the weighted center of gravity are used to define new sub-regions as a foundation for generating subsequent points. Then, these points are evaluated and are used as the new set of reference points. This process is repeated until the stopping criteria are satisfied. More detailed descriptions are given in Glover (1977, 1994b, 1995).

2.3. TABU SEARCH (TS)

The TS approach is a heuristic to solve combinatorial optimization problems. Recently, the TS approach has been applied to solve continuous global optimization problems (Battiti and Tecchiolli, 1994, 1996; Cvijovic and Klinowski, 1995; Glover, 1994a; Glover and Laguna, 1993). The TS approach is given in Figure 3.

The basic idea of the TS approach is to imposing restrictions on the search process to guide it to investigate difficult regions. The TS approach starts its procedures with an initial solution. Then, the TS constructs a neighborhood from this solution to identify adjacent solutions. Next, the objective function associated with each adjacent solution is determined. Before determining the best move, the TS approach determines the set of admissible moves first. When the set of admissible moves has been constructed, a best move can be selected. There are other strategies which can be used to filter the admissible moves and select the best move. These strategies are known as diversification and intensification. These strategies are explained in depth in Glover and Laguna (1993).

3. Description of the hybrid scatter genetic tabu (HSGT) search

The HSGT search combines the features of the above methods. Specifically, we extend the hybrid scatter genetic search that was proposed by Trafalis and Al-Harkan (1995) to TS by introducing the notion of memory to explore the solution space

```
begin
    generate an initial solution for the problem
    evaluate the objective function value
    while (not termination-condition) do
        construct a set of adjacent solutions from the current solution
        evaluate each solution constructed
        construct a set of admissible solutions from the set of adjacent solutions
        select the best solution from the admissible set and set it as the current
        solution
    end
end
```

Figure 3. The general tabu search approach

more extensively. Our approach, a hybrid scatter genetic tabu (HSGT) search, can be considered as a hybrid approach which attempts to find the global optimum of a general nonlinear function. The HSGT search is given in Figure 4 and explained as follows:

The HSGT search begins with an initial solution, X^k. This solution can be considered as the best solution X_{best} for Step 1. Then, at Step 2, reference points X_j^k are generated by using random search directions d_j and a step size μ_j to construct a collection of feasible solutions. The number of reference points, m, needed to attempt to cover the whole solution space, is expressed in terms of the dimensionality of the problem as $m = Cn$, where n is the dimension of the problem and C is an integer constant. After generation of reference points, function evaluations $f(X_j^k)$ are performed for each of the reference points X_j^k at Step 3. In Step 4, a weight W_j is assigned to each point generated according to its function value $f(X_j^k)$. This weight depends on how good that particular point is. For the minimization problems, the largest weight is given to the point with the smallest objective function value. In Step 5, a new solution X^{k+1} is determined as a weighted center of gravity of the reference points X_j^k generated with corresponding weights W_j. The weighted center of gravity is computed as a convex combination of all reference points so that the new solution becomes feasible. The HSGT search includes the features of the SS approach by using the concepts of trajectory and clustering methods combined. The identifications of the consecutive weighted centers of gravity emulate the trajectory method. The groups of similar reference points, which form new reference points that will be explained later in this section, behave similarly like in clustering methods. One can see that, up to this step, the HSGT search is similar to SS approach.

After this step some characteristics of TS and GAs are embedded to the HSGT search to test the status of the new solution X^{k+1} and to update the search directions D_j, respectively. The characteristics of the TS are included in the HSGT search by testing the tabu status of every new center of gravity generated and by implementing the classical aspiration criterion. For each new center of gravity or the new

Step 1: Generate an initial solution, X^k and set $X_{best} = X^k$
Step 2: Generate reference points, $X_j^k = X^k + \mu_j D_j$, where $j=1, \ldots, m$, μ_j is a step length, and D_j is a normalized random search direction, d_j.
Step 3: Evaluate the objective function $f(X_j^k)$ for each of the reference points X_j^k.
Step 4: Assign a weight W_j to each of the reference points according to $f(X_j^k)$.
Step 5: Compute a new solution, X^{k+1} as a weighted center of gravity by using convex combination of reference points X_j^k and weights W_j.
Step 6: If $f(X^{k+1}) < f(X_{best})$ then go to Step 8.
Step 7: Check tabu status of X^{k+1}. If X^{k+1} satisfies the tabu conditions then do not accept X^{k+1} as a solution, go to Step 2 and generate new reference points X_j^k by generating new random search directions d_j.
Step 8: $X_{best} = X^{k+1}$, update search directions D_j^{k+1} using GA operators.
Step 9: If maximum number of iterations are satisfied, HSGT search stops with the best solution as X_{best}
Step 10: Go to Step 2 and use D_j^{k+1} to generate new reference points.

Figure 4. The HSGT Search.

solution X^{k+1}, the tabu status is defined by how close it is to the previous solution, by how much it changes the objective function, and by how much it destroys or improves the objective function. The aspiration criterion used in the HSGT search is activated, when a move that was tabu is attempted and results in a solution that is better than any solution so far.

The HSGT search also combines the characteristics of the GAs through crossover and mutation of the existing search directions. In the HSGT search, the population size is the number of the search directions and the directions are the chromosomes in our population. The binary tournament is used to select directions from the old directions to produce new directions that will be in the new generation. The binary tournament is performed by first selecting randomly two directions from the population. Then, the GA operators are applied to these two directions. Next, the best of the two produced directions will be selected and allowed to enter the pool of the potential directions for the next generation. These procedures will be repeated until a new generation of chromosomes or directions is produced. Once we produce a new generation of chromosomes or directions, we go to Step 2 to generate new reference points by using them. This whole process is repeated until the desired number of iterations are satisfied. When we reached to this desired number of iterations, the HSGT search (Figure 4) stops with X_{best} as the solution of the problem.

4. Implementation of the hybrid scatter genetic tabu (HSGT) search

In this section, some implementation issues of the HSGT search are discussed. It is possible to offer alternative approaches to the implementation of the HSGT.

4.1. GENERATION OF AN INITIAL SOLUTION, X^k

The initial solution, X^k, is randomly generated by using the uniform distribution between the upper and the lower bounds of each variable to satisfy feasibility.

4.2. GENERATION OF REFERENCE POINTS, X_j^k

These reference points are generated as follows. First, a set of m search directions d_j is randomly generated according to the normal distribution with zero mean and unit variance. The generated directions are normalized. A step size μ in each direction is required to be computed to generate this set of m reference points. The step size, μ, is computed using two methods which are selected randomly. We use these two methods together in order to make the implementation more robust. The first method is the line search technique that was proposed by Press et al. (1992). The second method is a fixed step size technique. In this method, the step length, μ, is computed as follows: $\mu = \min\{\alpha, \beta\}$, where:

$$\alpha = \min\left\{\frac{X_U^i - X^i}{d^i} : \text{if } d^i > 0\right\}$$

$$\beta = \min\left\{\frac{X_L^i - X^i}{d^i} : \text{if } d^i < 0\right\}$$

X_U^i: is the upper bound for the i^{th} component of X_U.
X_L^i: is the lower bound for the i^{th} component of X_L.
s^i: is the i^{th} component of the direction investigated.
X^i: is the i^{th} component of the projection point.
Where X_U, X_L, d, and X are in R^n

Then, the value of μ is adjusted according to the number of iterations that have been performed so far as follows:

$$\mu = \left\{\begin{array}{ll} 0.75\mu, & \text{if iteration number} \leqslant 0.7^* \text{ maximum iteration} \\ 0.25\mu, & \text{if iteration number} > 0.7^* \text{ maximum iteration} \end{array}\right\}$$

The above adjustment, means that the HSGT search will have bigger steps at the beginning of the search and then it will have smaller steps when near to the end of the search.

It should be noted that, if the current solution point, X^i, is on the boundary of the solution space then the search direction d_j is likely to bring a step size equal to zero depending on the sign of d_j. In this case probability of trapping on the boundary or local solution is very high. To reduce this probability, m search directions are used but that does not guarantee the global solution all the time.

4.3. ASSIGNING A WEIGHT W_j

Assign a weight, W_j, to each of the reference points, X_j according to its value $f(X_j)$ as follows:

$$W_j = Cj/\sum_{i=1}^{m} C_i \text{ where } C_j = 1/f(X_j)\forall j$$

and

$$\sum_{j=1}^{m} W_j > 0.$$

If $f(X_j) = 0$, the point X_j will be given a very large C_j. If there is a mixture of positive and negative function values, $f(X_j)$ then the function values needs to be shifted to positive values. This procedure can be accomplished by defining $S = \min(f(X_j))$, $j = 1, \ldots, m$. Then, update the $f(X_j)$ as follows:

$$f(X_j) = f(X_j) + |S| + 1, j = 1, \ldots, m.$$

It should be noted that, for the best reference point that has the best objective value, the highest weight value is assigned.

4.4. TS ELEMENTS

This subsection describes the elements of the TS that are used in the HSGT search. A move is tested to check whether it is tabu or not. In the HSGT search, there are the following three criteria which are used to determine if a move is tabu.

(1) $\|X^k - X^{k+1}\|$, which is the total distance moved.

(2) $|f(X^k) - f(X^{k+1})|$, which is the total change in the objective function.

(3) $\frac{f(X^k)-f(X^{k+1})}{f(X^{k+1})} * 100$, which is the percentage improvement or destruction that will be accrued if the new move is accepted.

Thus, a move is considered tabu if it is both closely located to the previous point and the change in its objective function is very small. Hence, if a move satisfied the tabu conditions then the previous center of gravity is reserved and a new set of m directions is generated. In Step 7, a check is made to see if the aspiration criterion is satisfied. The aspiration criterion used in the HSGT search is activated, when a move that was tabu is attempted and results in a solution that is better than any visited solution so far. The tabu list size is chosen to be dynamic and it was generated randomly according to the uniform distribution between two integer numbers. In our implementation the tabu list size is chosen from the uniform distribution between $a = 6$ and $b = 13$.

4.5. GA OPERATORS

The GA operators implemented in the HSGT search are the whole arithmetical crossover and the general mutation. The whole arithmetical crossover is implemented by performing a convex combination of two search directions as follows:

$$D_K^{k+1} = \nu D_K^k + (1-\nu)D_L^k \text{ and } D_L^{k+1} = \nu D_L^k + (1-\nu)D_K^k, \text{ where } 0 \leqslant \nu \leqslant 1.$$

D_K^k & D_L^k are the two old search directions that are randomly selected.

In Step 8, when the two new search direction are formed, one of them will be selected according to how good that particular direction is. Specifically, using X_k, D_K^{k+1}, and D_L^{k+1}, a move can be made in each direction to two new points (where X^k is the current center of gravity). Then, the objective function for each point is computed. Next, the direction that gives the smallest $f(X^k)$ for those two new points is selected as a new direction and put in the pool of the potential directions for the next generation. The general mutation is to alter randomly one of the coordinates of the search directions. The general mutation is performed by first selecting a direction randomly. Then, a coordinate of the above search direction is selected. Next, the value of this coordinate is replaced with a random value that is generated according to the normal distribution with zero mean and unit variance.

It is clear that the HSGT search is stochastic in nature which makes the theoretical analysis considerably difficult. Hence, the convergence of the HSGT search can not be proven analytically. However, the behavior of the developed HSGT search will be determined computationally through a series of experiments which will be discussed in the following section.

5. Computational results

As mentioned earlier, the behavior of the HSGT search is determined by an extensive computational experimentation by using 19 well-known global optimization test problems listed in Table 1. The mathematical representations of these problems are presented in the Appendix A. These problems are well-established test problems in the literature and commonly used by other researchers to test their search algorithms (Al-Sultan and Al-Fawzan, 1997; Battiti and Tecchiolli, 1996; Cvijovic and Klinowski, 1995; Dixon and Szegö, 1978a, 1978b; Duan et al., 1993). Problems numbered between 6 and 19 are introduced by Koon and Sebald (1995) for testing the global convergence of evolutionary programming strategies. Most of these test problems have low dimensionality (2–4) but with several local and global optima and large flat regions enclosing the global optima. These characteristics make the selected test functions difficult to find their global optima.

To determine the behavior and global convergence of the HSGT a series of experiments was performed. In order to conduct performance analysis of the HSGT approach, three more global optimization methods were tested and compared on these problems. The global optimization methods used for performance analysis

Table 1. Test problems used for performance analysis

Problem Number	Name	Dimension	Number of optima	
			Local	Global
1	Goldstein and Price’s function	2	4	1
2	Himmelblau’s function	2	4	4
3	Rosenbrock’s function	4	1	1
4	Shekel’s family	4	5	1
5	Shekel’s family	2	10	1
6	Powell’s function	4	1	1
7	6-Hump CamelBack function	2	6	2
8	Shubert function	2	$\geqslant 760$	$\geqslant 18$
9	General test function	2	4	1
10	General test function	3	8	1
11	General test function	4	16	1
12	General test function	2	Several	1
13	General test function	2	$\geqslant 3$	2
14	General test function	2	$\geqslant 3$	2
15	General test function	2	$\geqslant 3$	2
16	General test function	2	$\geqslant 3$	2
17	Rastrigin function	2	50	1
18	Branin function	2	$\geqslant 3$	$\geqslant 3$
19	General test function	1	38	7

Table 2. Global optimization methods used for performance analysis

Method	Name	Reference
HSGT	Hybrid scatter genetic tabu search	Trafalis and Kasap, 2001 (this paper)
HSG	Hybrid scatter genetic search	Trafalis and Al-Harkan, 1995
SA	Simulated annealing	Goffe et al., 1994
GENOCOP	Genetic algorithm for numerical optimization with constraints	Michalewicz and Janikow, 1996b

are shown in Table 2. As mentioned earlier HSG is the predecessor of the HSGT. In addition, a simulated annealing and a GA approach are also tested. The HSGT, HSG and SA are coded in FORTRAN. GENOCOP (version 3.0) code written in C++ is downloaded from the author’s website
http://www.coe.uncc.edu/ zbyszek/evol-systems.html.

For the methods that are proposed by us, the following parameters were used for all problems: The parameter C that determines the number of reference points at Step 2 is set to 8. It is expected that, less reference points will increase the speed but decrease the success for finding global optima. The parameter ν that is used for crossover operator at Step 8 is randomly generated at each iteration. The new solution at Step 7 is assumed tabu if the total distance moved at the current iteration is less than 0.1 and the total change in the objective function is less than 0.005 or the percentage of destruction at the objective function is higher than a percentage that is generated randomly between 50 and 75%. The tabu list size is generated randomly between 6 and 13 at each iteration. After setting these parameters, each global optimization method was tested by running 100 trials on each test function. All tests have been executed without any individual tuning of these parameters. Each trial began with an independent randomly generated initial solution. Then, the number of global optima found out of 100 trials is reported as the success rate of the method for each problem. These results are presented in Table 3. The common stopping criteria is the maximum number of iterations that is set to 100 for the methods were proposed by us.

From Table 3, it can be seen that the HSGT search performed significantly better than other methods (with the exception of GENOCOP) and reached the global optimum solution in all hundred runs for 15 test problems. The success rate of the GENOCOP was better for six problems and was worse than our approach for two problems. For other problems both techniques performed in a similar way.

More detailed experiments to investigate the quality of the solution obtained by the HSGT search were conducted. The quality of solutions is determined by the percentage deviation and distance from the global optimum, and CPU time. For these experiments, the first 3 methods namely HSGT, HSG, and SA are tested and compared on these problems. The results of these experiments are shown in Table 4.

From Table 4, for the 15 functions that the HSGT search reached the global optimum, the percentage deviation is zero except for problem 17. For the problems that the HSGT search was not able to find the global optimum 100% such as problem numbers 4, 5, 8, 10, 11, and 13, the HSGT converged to the global optimum more often than the other two approaches with average percentage of deviations of 58.79, 78.36, 0.3, 1.06, 4.6, and 30%, respectively. This implies that the HSGT approach converged to the global optimum more often than the other two approaches for 19 test functions. For functions 4, 5, 8, and 13, the SA achieved a better average in terms of the average percentage of deviations. The HSGT and the SA obtained the same average percentage deviation for function 19 while the HSGT and the HSG obtained the same average percentage deviations for functions 3, 14, and 15 respectively. In terms of the CPU time and the average number of iterations the HSG was the best performer. The average CPU time needed by the HSGT, HSG, and SA ranged between 1.3 and 19.55 s, 0.35 and 8.61 s, and 3.58 and 27.43 s respectively. Also, it can be seen that the HSG and SA approaches were

Table 3. The success rate for each test problem and method

Problem	HSGT	HSG	SA	GENOCOP
1	100	100	100	100
2	100	100	100	100
3	100	100	100	100
4	3	0	1	24
5	1	0	0	56
6	100	100	29	58
7	100	100	100	100
8	61	5	58	100
9	100	52	74	100
10	76	19	7	100
11	36	4	0	74
12	100	100	100	100
13	70	99	100	100
14	100	100	100	100
15	100	100	20	100
16	100	98	5	100
17	100	45	100	93
18	100	100	100	100
19	100	84	100	100

effected by the problem size as shown by their results for functions 4, 5, 6, 10, and 11. However, the HSGT approach was more robust and it obtained better results for these functions.

Another detailed experimentation is conducted to understand the effect of embedding the notion of memory that is originated from TS. In this experimentation, objective function value at each iteration is recorded for the 100 trials for some test problems. Then, the average objective function value for the 100 trials for each iteration is plotted. Corresponding plots are shown in Figs. 5–10.

From these figures, we can see the improvement coming from embedding the TS approach in the HSG approach. In fact, the HSGT search finds the solution in less iterations than HSG for the tested six problems. Note that, these figures also prove the success of the HSGT over HSG. Note that Figs. 6 and 10 imply that, either of the search methods has a difficulty for finding global optimum. On the other hand, Table 3 shows that we had 61 and 100% success with the HSGT on these problems respectively. This contradiction is caused by the sinusoidal nature of the problems and the averaged objective values at each iteration to plot the results. On the other hand, this improvement over HSG increased CPU times of

Table 4. CPU time, distance and percentage deviation from global optimum

Problem	CPU time (s)			Distance from global optimum			Percentage deviation from global optimum		
	HSGT	HSG	SA	HSGT	HSG	SA	HSGT	HSG	SA
1	2.39	1.30	5.89	0.0001	0.0006	0.0135	0.00	0.02	0.45
2	1.90	1.01	4.53	0.0000	0.0005	0.0030	0.00	0.05	0.30
3	1.30	0.47	3.92	0.0000	0.0000	0.0006	0.00	0.00	0.06
4	5.66	5.13	14.88	5.9691	7.1388	4.9984	58.79	70.31	49.23
5	9.10	8.61	27.43	8.2567	8.3029	5.0693	78.36	78.80	48.11
6	3.41	2.41	4.88	0.0001	0.0006	0.2167	0.01	0.06	21.67
7	3.72	1.18	5.42	0.0000	0.0023	0.0011	0.00	0.23	0.10
8	4.45	1.17	5.33	0.5617	67.1338	0.1226	0.30	35.95	0.07
9	1.59	1.11	4.94	0.0017	3.3019	0.0840	0.00	4.22	0.11
10	2.27	2.07	6.14	1.2474	11.4841	4.6599	1.06	9.77	3.97
11	3.48	3.31	7.53	7.2232	22.3759	22.9010	4.61	14.28	14.62
12	1.62	0.88	3.91	0.0000	0.0001	0.0003	0.00	0.01	0.03
13	5.67	1.44	7.09	0.1222	0.0043	0.0016	30.00	1.05	0.40
14	1.91	1.44	6.86	0.0000	0.0009	0.0109	0.00	0.00	0.06
15	2.26	1.38	6.96	0.0000	0.0030	0.4081	0.00	0.00	0.18
16	1.57	1.25	7.07	0.0000	48.5762	1.8642	0.00	2.00	0.08
17	2.40	1.01	4.10	0.0475	0.2346	0.0480	2.37	11.73	2.40
18	4.34	0.97	4.51	0.0000	0.0014	0.0012	0.00	0.35	0.31
19	11.37	0.35	3.58	0.0002	0.0932	0.0000	0.00	2.76	0.00

HSGT. At the development stage of the HSGT search, we were expecting this result since we are increasing the number of the tested solutions for one trial by rejecting some solutions.

6. Applications to neural network training

The supervised neural network training can be considered as a nonlinear programming problem (Haykin, 1994). Specifically, we minimize a nonlinear error function between the calculated output and the given desired output with respect to parameter weights. To illustrate this, the XOR-2 problem will be used. In the XOR-2 problem, the desired output will be 0 if both of the inputs are the same otherwise 1. The error function for XOR-2 problem will be as follows.

$$\text{Minimize } f(x) = \sum_{i=1}^{4} (y_i - \text{target}_i)^2$$

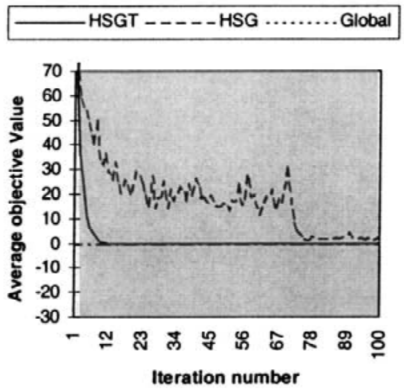

Figure 5. Average objective function value versus iteration number for function 2.

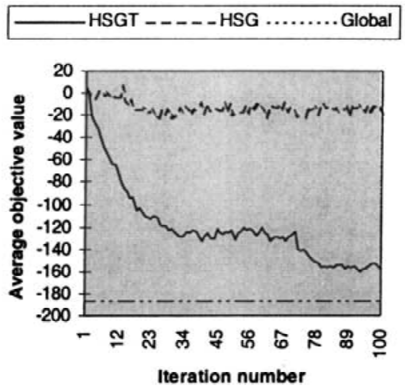

Figure 6. Average objective function value versus iteration number for function 10.

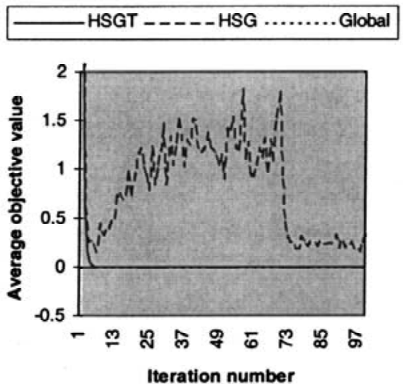

Figure 7. Average objective function value versus iteration number for function 14.

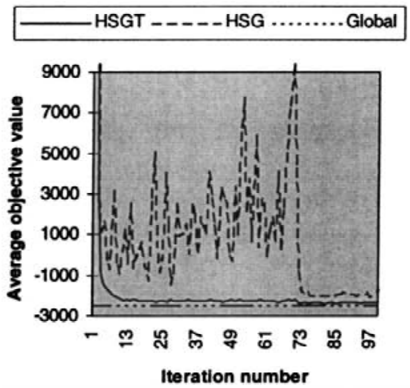

Figure 8. Average objective function value versus iteration number for function 18.

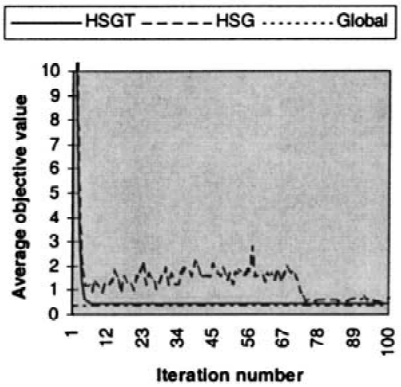

Figure 9. Average objective function value versus iteration number for function 20.

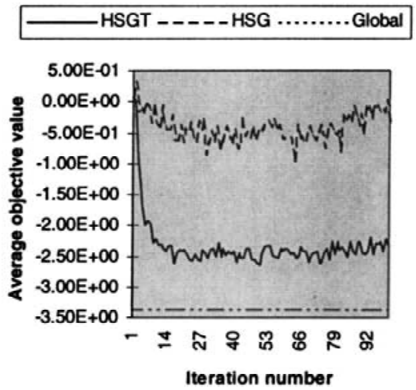

Figure 10. Average objective function value versus iteration number for function 21.

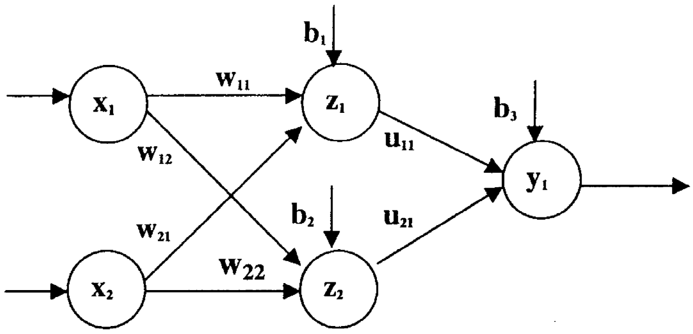

Figure 11. The neural network for XOR-2 problem.

. Results of training for neural networks.

Problem	Dimension	HGST	HGS
XOR-2	9	96	23
XOR-3	16	69	14
XOR-4	25	46	9

The neural network architecture for the XOR-2 problem is shown in Figure 11. The output of this neural network can be calculated as follows:

$$y = f(b_3 + z_1u_{11} + z_2u_{21}) \text{ where } f(x) = (1 + e^{-x})^{-1}$$

Note that, the activation function $f(x)$ is a nonlinear function. Therefore, the error function will have more than one local minimum. It is desirable to find the global optimum for this problem. To find the global optimum to this problem the HSGT search is used. The results of the training of neural network for the XOR-2, XOR-3, and XOR-4 problems are given in Table 5.

7. Conclusion and recommendations

A novel metaheursitics approach to find global optimum of continuous optimization problems with box constraints is proposed. This approach combines the characteristics of modern metaheuristics such as scatter search (SS), genetic algorithms (GAs), and tabu search (TS) and named as hybrid scatter genetic tabu (HSGT) search. The development of the HSGT search, parameter settings, experimentation, and efficiency of the HSGT search are discussed. The HSGT has been tested against a simulated annealing algorithm, a GA under the name GENOCOP, and a modified version of a hybrid scatter genetic (HSG) search by using 19 well known test functions. Applications to Neural Network training are also examined. From the computational results, the HSGT search proved to be quite effective in identifying the global optimum solution which makes the HSGT search a promising approach to solve the general nonlinear optimization problem. Applications to Neural Networks training are also examined. Interior point method versions of the HGST and HGS are also studied (Trafalis and Kasap, 1996, 1998).

The HSGT search does not require any gradient or Hessian matrix calculations. Therefore, it does not suffer from ill-conditioning. This HSGT search is a promising approach to solve the general nonlinear global optimization problem. Therefore, it can be extended to solve general nonlinear constrained optimization problems with convex feasible region using one of the transformation methods (penalty or barrier function approach), and solve nonlinear mixed integer programming problems.

Acknowledgements

The authors are grateful to the referees for numerous useful comments, in particular in some suggestions for improving the presentation of the HSGT search.

Appendix A

In this appendix, the 19 test problems are given. To learn more about these functions, refer to Goldstein and Price (1971), Dixon and Szegö (1978a, 1978b), Reklaitis et al. (1983), Smith et al. (1991), and Koon and Sebald (1995).

Function 1 (2-D): Goldstein and Price's function: $X_i \in [\pm 2, \pm 2] \forall i$. There are four local minima and the global objective function value is 3.

$$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$$

$$[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

Function 2 (2-D): Himmelblau's function: $X_i \in [\pm 6, \pm 6] \forall i$. There are four global minima with an objective function value of 0. $f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
Function 3 (2-D): Rosenbrock's function: $X_i \in [\pm 2, \pm 2] \forall i$. There is a unique global minimum with an objective function value of 0. $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
Functions 4 and 5 (4-D): Shekel's family: $X_i \in [0, 10] \forall i$. Function 4 has 5 local minima and function 5 has 10 local minima. The global objective function value for functions 4 and 5 are -10.1531957 and -10.5362836 respectively.

$$f(x) = -\sum_{i=1}^{m} [(x - a_i)^T (x - a_i) + c_i]^{-1} x = (x_1, \dots, x_n)^T,$$

$$a_i = (x_{i1}, \dots, x_{in})^T, c_i > 0, m = 5 \text{ or } 10.$$

The values for a_i and c_i are given in the following table.

i		a_i			C_i	i		a_i			c_i
1	4	4	4	4	0.1	6	2	9	2	9	0.6
2	1	1	1	1	0.2	7	5	5	3	3	0.3
3	8	8	8	8	0.2	8	8	1	8	1	0.7
4	6	6	6	6	0.4	9	6	2	6	2	0.5
5	3	7	3	7	0.4	10	7	3.6	7	3.6	0.5

Function 6 (4-D): Powell's function: $X_i \in [\pm 3, \pm 3] \forall i$. This function has a unique global with an objective function value of 0.

Function 7 (2-D): Six-Hump CamelBack function: $X_i \in [\pm 5, \pm 5] \forall i$. This function has six local minima and two global minima with an objective function value of -1.031628.

Function 8 (2-D): Shubert function: $X_i \in [\pm 20, \pm 20] \forall i$. This function has more than 760 local minima and more than 18 global minima with an objective function value of -186.7309.

$$f(x) = \left\{ \sum_{i=1}^{5} i \cos[(i+1)x_1 + i] \right\} \left\{ \sum_{i=1}^{5} i \cos[(i+1)x_2 + i] \right\}$$

Functions 9, 10, and 11 (N-D): General test function: $X_i \in [\pm 20, \pm 20] \forall i$. This function has 2^N local minima and one global minimum. The global objective function values are -78.332331, -117.4984, and -156.66466 for N equal 2, 3, and 4 respectively.

$$f(x) = \frac{1}{2} \sum_{j=1}^{N} (x_j^4 - 16x_j^2 + 5x_j), \; N = 2, 3, \text{ or } 4.$$

Function 12 (2-D): General test function: $X_i \in [\pm 5, \pm 5] \forall i$. This function has several local minima with an objective function value of 0. $f(x) = 0.5x_1^2 + 0.5[1 - \cos(2x_1)] + x_2^2$

Functions 13, 14, 15, and 16 (2-D): General test function: $X_i \in [\pm 5, \pm 5] \forall i$ for functions 15 and 16, and $X_i \in [\pm 20, \pm 20] \forall i$ for functions 17 and 18. These functions have more than three local minima and two global minima. The global objective function value for functions 15, 16, 17, and 18 are -0.407461, -18.058697, -227.765747, and -2429.414749 respectively.

$$f(x) = 10^n x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^m (x_1^2 + x_2^2)^4, \; n = -m.$$

Function 17 (2-D): Rastrigin function: $X_i \in [\pm 5, \pm 5] \forall i$. This function has 50 local minima and one global minimum with an objective function value of -2.

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

Function 18 (2-D): Branin function: $X_i \in [\pm 20, \pm 20] \forall i$. This function has more than 3 local and global minima with an objective function value of 0.397887.

$$f(x) = (x_2 - 5.1x_1^2/4\pi^2 + 5x_1/\pi - 6)^2 + 10(1 - 1/(8\pi)) \cos x_1 + 10$$

Function 19 (1-D): General test function: $X \in [\pm 20, \pm 20]$. This function has 38 local minima and 7 global minima with an objective function value of -3.372897.

$$f(x) = -\left\{\sum_{i=1}^{5} \sin[(i+1)x + i]\right\}$$

References

Al-Sultan, K. S., and Al-Fawzan, M. A. (1997), A tabu search Hooke and Jeeves algorithm for unconstrained optimization, *European Journal of Operational Research* 103, 198–208.

Androulakis, I., and Venkatasubramanian, V. (1991), A genetic algorithm framework for process design and optimization, *Computers and Chemical Engineering* 15(4), 217–228.

Battiti, R., and Tecchiolli, G. (1994), The reactive tabu search, *ORSA Journal on Computing* 6 (2), 126–140.

Battiti, R., and Tecchiolli, G. (1996), The continuous reactive tabu search: blending combinatorial optimization and stochastic search for global optimization, *Annals of Operations Research-Metaheuristics in Combinatorial Optimization* 63, 153–188.

Becker, R. W., and Lago, G. V. (1970), Global Optimization Algorithm, *Proceedings of the 8th Allerton Conference on Circuits and Systems Theory.*

Branin, F. H. (1972), Widely convergent methods for finding multiple solutions of simultaneous nonlinear equations, *IBM Journal of Research Developments*, 504–522.

Corana, A., Marchesi, M., Martini, C., and Ridella, S. (1987), Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm, *ACM Transaction on Mathematical Software* 13(3), 262–280.

Cvijovic, D. and Klinowski, J. (1995), Taboo search: an approach to the multiple minima problem, *Science* 267, 664–666.

Davis, L. (1991), *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Dixon, I. C. W. and Szegö, G. P. (1978a), *Towards Global Optimization* 1. North-Holland, New York.

Dixon, I. C. W., and Szegö , G. P. (1978b), *Towards Global Optimization* 2. North-Holland, New York.

Duan, Q. Y., Gupta, V. K., and Sorooshian, S. (1993), Shuffled complex evolution approach for effective and efficient global minimization, *Journal of Optimization Theory and Applications* 76(3), 501–521.

Fleurent, C., Glover, F., Michelon, P., and Valli, Z. (1995), A scatter search approach for unconstrained continuous optimization, Working paper, University of Colorado, Boulder, CO.

Floudas, C. A., and Pardalos, P. M. (1992), *Recent Advances in Global Optimization,* Princeton University Press, Princeton, NJ.

Garcia, C. G., and Gould, F. J. (1980), Relations between several path following algorithms and local global Newton methods, *Siam Review* 22, 263–274.

Glover, F. (1977), Heuristics for integer programming using surrogate constraints, *Decision Sciences* 8/1, 156–166.

Glover, F. (1994a), Tabu search nonlinear and parametric optimization (with links to Genetic Algorithms), *Discrete Applied Mathematics* 49, 231–255.

Glover, F. (1994b), Genetic algorithms and scatter search: unsuspected potentials, *Statistics and Computing* 4, 131–140.

Glover, F. (1995), Scatter search and start-paths: beyond the genetic metaphor, *OR Spectrum* 17, 125–137.

Glover, F. and Laguna, M. (1993), Tabu search. In C. R. Reeves (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, 70–141. John Wiley & Sons, New York.

Goffe, W. Ferrier, G., and Rogers, J. (1994), Global optimization of statistical functions with simulated annealing, *Journal of Econometrics* 60, 65–99.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Goldstein, A. A., and Price, J. F. (1971), On descent from local minima, *Mathematics of Computation* 25, 569–574.

Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*. Macmillian College, New York.

Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA.

Horst, R., Pardalos, P. M. and Thoai, N. V. (1995). *Introduction to Global Optimization*, Kluwer Academic Publishers, New York:

Kirkpatrick, S., Gellat, D. and Vecci, M. (1983), Optimization by simulated annealing, *Science* 220/4598, 671–680.

Koon, G. and Sebald, A. (1995), Some interesting test functions for evaluating evolutionary programming strategies. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel (eds.), *Proceedings of the Fourth Annual Conference on Evolutionary Programming.* MIT Press, Cambridge, MA.

Michalewicz, Z. (1996a), *Genetic Algorithms + Data Structure = Evolution Programs* (3d ed.). Springer, New York.

Michalewicz, Z. and Janikow, C.Z. (1996b), GENOCOP: A genetic algorithm for numerical optimization problems with linear constraints. *Communications of the ACM* 39(12), 175–201.

Press, W. H., Teukosky, S. A., Vetterling, W. T. and Flannery, B. P. (1992). *Numerical Recipes in FORTRAN: The art of Scientific Computing*. Cambridge University Press, New York.

Price, W. L. (1978). A controlled random search procedure for global optimization. In I. C. W. Dixon, and G. P. Szegö (eds.), *Towards Global Optimization* 1, 71–84. North-Holland, New York.

Reeves, C. R. (1993), *Modern Heuristic Techniques for Combinatorial Problems*, John Wiley & Sons, New York.

Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M. (1983), *Engineering Optimization Methods and Applications*. John Wiley and Sons, New York.

Rinnooy Kan, A. H. G. and Timmer, G. T.. (1989), Global Optimization. In G. L. Nemhauser, A. H. G Rinnooy Kan, and M. J. Todd (eds.), *Handbooks in OR@MS*, 631–662. North-Holland, New York.

Shubert, B. O. (1972), A sequential method seeking the global maximum of a function, *SIAM Journal on Numerical Analysis* 9, 379–388.

Smith, S., Eskow, E. and Schanbel, R. (1991), Large adaptive, asynchronous stochastic global optimization algorithms for sequential and parallel computation. In T. F. Coleman, and Y. Li (eds), *Large-Scale Numerical Optimization*. SIAM, Philadelphia.

Srinivas, M. and Patnaik, L. M. (1994). Genetic algorithms: a survey, *Computer* 27(1), 17–26.

Törn, A. L. (1978), A search clustering approach to global optimization. In I. C. W. Dixon, and G. P. Szegö (eds), *Towards Global Optimization* 2, 71–84. North-Holland, New York.

Törn, A. L and Zilinskas, A. (1989), Global Optimization Lecture Notes. In *Computer Science* 350. Springer, Berlin.

Trafalis, T. B. and Al-Harkan, I. (1995), A continuous scatter search approach for global optimization, Extended Abstract. In *Conference in Applied Mathematical Programming and Modeling (APMOD'95)*. London, UK.

Trafalis, T. B. and Al-Harkan, I. A hybrid scatter genetic tabu approach for continuous global optimization. In P.M. Pardalos, A. Migdalas and R. Burkard, (eds.), *Combinatorial and Global Optimization*. World Scientific Publishing Co., forthcoming.

Trafalis, T. B. and Kasap, S. (1996), An affine scaling scatter search approach for continuous global optimization problems, *Intelligent Engineering Systems Through Artificial Neural Networks*, (Ed. C. H. Dagli et al.), 6, ASME Press, 1027–1032.

Trafalis, T. B. and Kasap, S. (1998), An affine scaling genetic scatter tabu (ASGST) search: a hybrid of modern heuristics and interior point methods, *Proceedings of the 2nd International Symposium on Intelligent Manufacturing Systems IMS'98, Volume I*, Sakarya, Turkey, 283–292.