Journal of Global Optimization **14:** 415–436, 1999.

FRACTOP: A Geometric Partitioning Metaheuristic for Global Optimization

MELEK DEMIRHAN[1], LINET ÖZDAMAR[2,*], LEVENT HELVACıOĞLU[2] and ŞEVKET ILKER BIRBIL[1]
[1]*Yeditepe University, Department of Systems Engineering;* [2]*Istanbul Kültür University, Department of Computer Engineering, Istanbul, Turkey*⋆

(Received 3 June 1997; accepted in revised form 3 August 1998)

Abstract. We propose a new metaheuristic, FRACTOP, for global optimization. FRACTOP is based on the geometric partitioning of the feasible region so that search metaheuristics such as Simulated Annealing (SA), or Genetic Algorithms (GA) which are activated in smaller subregions, have increased reliability in locating the global optimum. FRACTOP is able to incorporate any search heuristic devised for global optimization. The main contribution of FRACTOP is that it provides an intelligent guidance (through fuzzy measures) in locating the subregion containing the global optimum solution for the search heuristics imbedded in it. By executing the search in nonoverlapping subregions, FRACTOP eliminates the repetitive visits of the search heuristics to the same local area and furthermore, it becomes amenable for parallel processing. As FRACTOP conducts the search deeper into smaller subregions, many unpromising subregions are discarded from the feasible region. Thus, the initial feasible region gains a fractal structure with many space gaps which economizes on computation time. Computational experiments with FRACTOP indicate that the metaheuristic improves significantly the results obtained by random search (RS), SA and GA.

Key words: FRACTOP, Geometric partitioning, Fuzzy measures

1. Introduction

Despite the advanced computer support we have at hand, optimization problems are still challenging for researchers working in mathematics as well as in operations research. Limited success has been achieved in classifying and identifying global optima in nonlinear programming (NLP). Classical gradient-based algorithms such as the Quasi-Newton method (Fletcher and Powell 1963; Gill and Murray 1972), the modified steepest descent algorithm (Armijo 1966), or the conjugate gradient method (Fletcher and Reeves 1964) may be readily trapped by local optima when the feasible region around the unique optimum is not well-conditioned (Beveridge and Schechter 1970). Consequently, search algorithms which enhance the exploration of the feasible region through the use of a population of solutions such as Multi Level Single Linkage (MLSL) (Kan and Timmer 1984; Törn and Viitanen

⋆ Corresponding author. Address: Gayrettepe Emekli Subay Evleri 23/5, Istanbul, Turkey (e-mail: lozdamar@hotmail.com)

1994), Genetic Algorithms (GA) (Goldberg 1989; Michalewicz 1994), and probabilistic search techniques such as Simulated Annealing (SA) (Kirkpatrick et al. 1983) have been developed for global optimization to quit local optima.

GAs conduct a search on a population of solutions (chromosomes) and they involve asexual reproduction, crossover (or sexual reproduction) and mutation. Through crossover, new chromosomes are added to the population while unfit old chromosomes die out. In MLSL, numerous solutions are generated randomly among which some of them are selected as a starting solution for local search. The selection of solutions is made geometrically, that is, if a solution is the only minimum within a specified radius, then it is selected for the application of local search. On the other hand, SA is a single solution search method and it converges to the optimum by randomly generating a neighbour solution to the incumbent solution, and then, guiding the search with a cooling scheme which enables it to refuse or accept a disimproving solution according to an acceptance probability. SA permits disimproving moves in order to leave behind local optima. The search methods discussed above have both strong and weak points. The GA is given a lot of freedom to generate new chromosomes using crossover and mutation operators. However, it may take a long time to converge to the global optimum. Furthermore, the GA may end up in a local optimum solution, because the crossover operator reshuffles the genes already existing in the previous population and new genes are introduced by the mutation operator only. Thus, the GA's performance may depend on the random population generated as the starting population. MLSL is a practical method with probabilistic guarantees of convergence. However, when the function optimized consists of a large number of local optima, it might be computationally inefficient to generate the number of solutions implied by the termination criterion on which the convergence probability is based. As for SA, besides depending on the starting solution, it has the additional disadvantage of working on a single solution and lacks the diversity of a GA.

Here, we develop a metaheuristic which shares some of the concepts in interval subdivision method (Caprani et al. 1993; Csendes and Pinter 1993) for nonlinear function optimization with simple bounds on the variables. However, here, rather than discarding the regions unlikely to contain the global optimum, the metaheuristic is focussed on locating the narrow subspace where the global optimum lies. The metaheuristic partitions the feasible region systematically into nonoverlapping subspaces to provide smaller sweeping spaces for any search method activated for detecting any local/global optimum solution encompassed by each subspace. Then, the sample of solutions contained in each subspace are compared against other regions' samples and the most promising subspace is selected for re-partitioning in the next level (the selected subspace is subjected to a more intensified search, since its size is smaller) while some of the inferior subspaces are discarded and some are preserved within their current bounds. The selection of the subspace for re-partitioning and the discarding action are executed according to a fuzzy measure. Due to the discarded subspaces the feasible region assumes a fractal nature with several space gaps of varying volumes. The metaheuristic is from now on called Fractal Optimization Approach (FRACTOP) due to the fractal characteristic of the remaining feasible region where the search continues.

In the following sections, the mechanics of FRACTOP are discussed in detail and several evidence collection methods used in this study are described. Next, the possible extensions to FRACTOP to enable its utilization in other areas of optimization, such as constrained NLP, mixed integer NLP (MINLP) and combinatorial optimization, are discussed. Finally, computational results are provided on thirteen nonlinear functions selected from the literature. The improvement in performance provided by FRACTOP is demonstrated by imbedding into FRACTOP, random search (RS), SA and GA and comparing the results with the corresponding stand alone applications.

2. Description of FRACTOP

FRACTOP is based on the strategy of 'divide and conquer'. The search is conducted by partitioning first the initial closure, $\mathcal{G}$, of the feasible region in $\Re^n$ into 2^n identical subregions, where n is the number of variables. These subregions are obtained by bisecting the lower and upper bounds of every variable and they constitute the surface faced at the first level of the search tree.

In the next step, a number of solutions, s, and their functional evaluations are collected (randomly or by using a metaheuristic such as SA or GA) from each subregion and it is assumed that the sample of solutions, A_{it}, collected from a subregion α_{it} (i and t represent the subregion and tree level indices, respectively)represents the behaviour of the function optimized within that subregion.

The evidence provided by each sample (each taken from the available subregions) is mapped onto a zero-one real valued interval through a fuzzy evidence measure, $m(A_{it})$, defined below.

$$m(A_{it}) = 1/s \sum_{j=1}^{s} [(F_j/F^*) \exp(1 - F_j/F^*),$$

where F_j and F^* are, respectively, the functional evaluation of solution j in sample A_{it}, and the best functional value obtained so far. $m(A_{it})$ tends to smooth out the relative differences between the functional evaluations. This measure is used against traps set by local optima so that if an influential local optima is located at a certain subregion, its contribution to the evidence measure is not accentuated. Otherwise, the search would immediately pick up that particular subregion and re-partition it into narrower and narrower subregional areas. As the areas get smaller, solutions which are very close to the local optimum are also identified and their contributions affect significantly the evidence measure. Thus, the chance of backtracking to the region where the global optimum lies diminishes and the search may terminate at a given tree level without finding the global optimum.

The belief that the global optimum $\mathbf{x}^*$ lies in a subregion α_{it} is denoted as Bel(α_{it}). The subregion $\alpha^*{}_{it}$ with the maximum belief is selected for repartitioning into identical subregions in the next level of the search tree whereas the lower and upper bounds of all variables in the remaining undiscarded subregions are preserved at their current values.

Bel(α_{it}) is a function that retains both the sample information gathered at current tree level t and the genetic information related to α_{it} acquired during the previous level. Bel(α_{it}) is expressed as follows:

$$Bel(\alpha_{it}) = \gamma Bel(\alpha'_{it}) + (1 - \gamma)m(A_{it}), \quad \text{for } 0 \leq \gamma \leq 1,$$

where α'_{it} (A'_{it}) denotes the parent subregion (sample obtained from the parent subregion), that is $\alpha_{it} \subset \alpha'_{it}$ in terms of size.

In order to carry out a global search rather than a local one (our fuzzy measure does not provide 100% guarantee that the optimal point is included in the selected territory), the subregions which are not re-partitioned are also subjected to re-sampling in the next level without being split into further smaller subregions. The sample size, s, can be fixed throughout the search or it may vary in proportion to the size of the subregion. However, s is not permitted to drop below a minimum number. Consequently, after a few partitioning iterations, subregions of different sizes are subjected to the same sample size and the smaller subregions are subjected to a more intensified search. Thus, in each level of the search tree simultaneous search is conducted on different regional partitions but in varying intensity.

Given that k_t subregions preserve their bounds at level t and $2^n - 1$ new subregions emanate from the selected subregion, the number of total subregions in level $t + 1$ becomes $k_t + 2^n - 1$. The sample collection procedure is then repeated in each subregion on the surface of the tree. However, before moving on to a next level in the search tree, some subregions may be discarded from the search based on the criterion of ‘Bel(α_{jt})/Bel ($\alpha^*{}_{it}$) $< \rho$’ where ρ is a constant less than one. Furthermore, if a subregion’s size is reduced below a certain small value, then it is also discarded. The elimination of some subregions j reduces the computational burden of the search while the feasible region assumes a fractal structure with some disjoint feasible subspaces. The resulting nonconvexity in the feasible region is not a problem for FRACTOP, because the search covers disjoint subregions independently anyway.

In Figure 1, we assume n=1 without loss of generality and demonstrate how the search may jump from one subregion to the next according to the support of gathered evidence. The partitioning iterations end according to a stopping criterion defined by the user. The best solution obtained so far is reported.

The geometric diversification and intensification described in Figure 1 may lead to a deeper search within a territory which is geometrically remote from the subregion selected for re-partitioning in the current level. In this respect, FRACTOP closely resembles the breadth-first branch and bound technique (B&B) often used in combinatorial optimization. The crucial difference is that in FRACTOP each

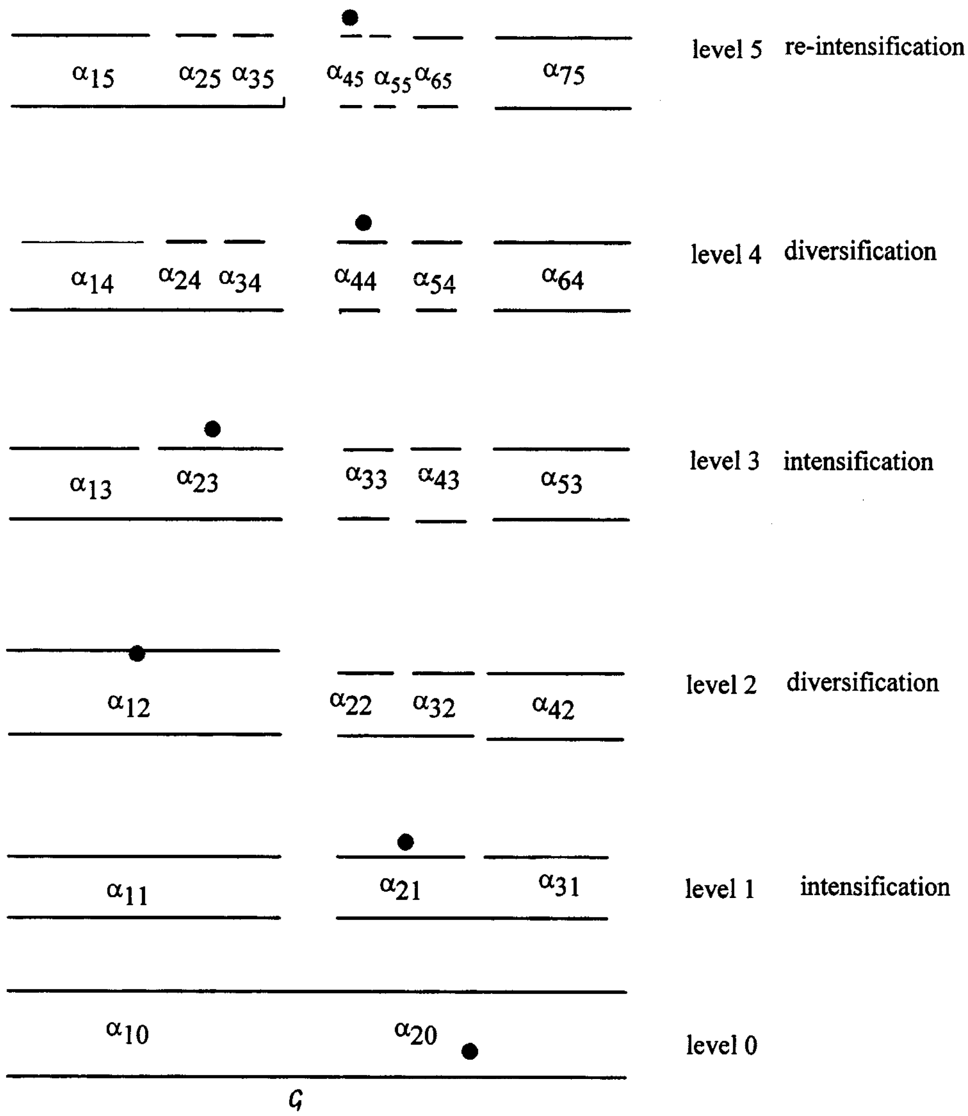

Figure 1. Visualisation of the search executed by FRACTOP in one dimension (○: subregion having the best belief measure for level t).

branch represents a portion of the feasible space rather than a singular partial solution. Similar to the B&B, some areas are chopped off and the areas which do not seem to be hopeless, are further investigated while the most promising one is searched more intensively.

Figure 2 sets an example for a FRACTOP search path in three dimensions. White blocks are discarded during the search which has been diverted from the first region to the second, when the comparison among the deeper partitions in the

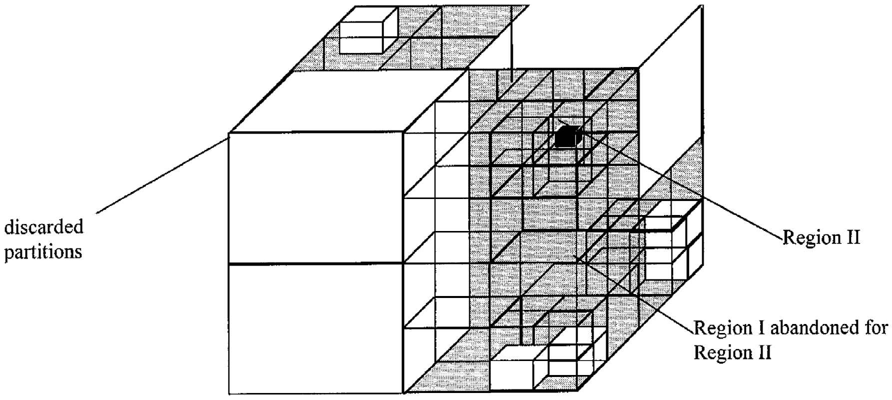

Figure 2. An example for the search path adopted by FRACTOP in three dimensional feasible space.

first region and the re-investigated second region lead to the conclusion that the first region no longer holds the promise it had before deeper partitioning. The darkest region is the one where FRACTOP has finally converged.

The systematic partitioning of the feasible region into disjoint subregions helps the search to overcome the premature convergence issue common to many search methods. Diversification and intensification are achieved naturally throughout the search process by diverting the search from an area which loses its promise once a deeper search is executed, to an area which becomes comparatively attractive after new evidence is gathered. The fuzziness in the evidence measure diminishes as more and more samples are taken from the corresponding subregion. It is important to note that unattractive areas are not totally abandoned. Rather, they are subjected to a less intensive search as compared to the area where the search is focussed.

FRACTOP economizes on computation time by applying search in nonoverlapping regions and discarding considerably unfavorable subspaces. Apart from stochastic search methods, classical gradient-based methods may also benefit from geometric partitioning. In the most simple case, for a given convex function with a valley around the unique optimum, gradient-based methods may converge rather slowly, zigzagging all the time. Smaller subregions where the gradient search (activated for a limited number of times) may eliminate a lot of zigzagging while discarded subregions may eliminate many gradient calculations taking place far from the region where the optimum lies.

Due to the geometric partitioning approach in FRACTOP, revisits to the same regions occurring in other non-geometric search methods (SA, GA) are eliminated considerably and additionally, the search achieves the ability to deal with feasible regions originally defined as fragmented disjoint regions (e.g., regions described by either/or constraints). In FRACTOP the search is confined to the specified boundaries of each subregion. Consequently, the search in each subregion is directed towards the local (global) optimum existing within that subregion. Non-geometric search methods have the problem of re-visiting the areas close to the solutions previously visited. In Tabu Search (Glover 1989), long and short term memory tabulists which forbid the search from returning to previously visited solutions are used in order to eliminate repetitive visits. In SA, the search may return to the area around the same local optimum after wandering about unfavorable feasible regions (in fact, this situation occurs frequently and efforts are made through tabulists imbedded in SA to avoid this situation; see Bozyel and Özdamar 1997; Özdamar and Bozyel 1998). The same issue arises in GAs where many chromosomes in the population acquire the same genes after converging to a local minimum and mutation is applied to prevent the trap set up by the local minimum. In an attempt to prevent premature convergence, Özdamar and Birbil (1998) permit infeasible chromosomes in the population and then occasionally apply a SA procedure incorporating tabulists to a number of chromosomes randomly selected from the population. In the latter reference, diversity is also achieved by migrating parallel populations.

In FRACTOP, although the number of subregions increase exponentially at each level t with the number of variables, n,it is possible to reduce the complexity by variable reduction techniques applied in each subregion (e.g., regression) or by utilizing other linear or low polynomial partitioning approaches. As presented here, the re-partitioning scheme of FRACTOP emphasizes each dimension on an equal scale which is the ideal case. Furthermore, FRACTOP can be executed on parallel processors where search in different subregions may be carried out independently. The global data to be exchanged between the host and the slave processors include the bounds of existing subregions, F^* and F_j belonging to A_{it} taken from the subregions at the current level t. The search tasks are carried out independently.

3. FRACTOP: A fuzzy optimization approach

FRACTOP is a fuzzy approach, because as long as every solution in the feasible region is not evaluated in an exhaustive manner (since the search is in real domain), the boundaries (maximum and minimum values) of the function optimized over each subregion are fuzzy. (Reviews of fuzzy theory is given by Klir and Folger (1988) and Zimmermann (1991) among others.)

The type of uncertainty involved here is close to the concept of fuzziness called non-specificity by Yager (1983) and resolutional uncertainty by Pal and Bezdek (1994). In the case of our metaheuristic, although the function can be evaluated exactly for a given solution, the manner in which we collect samples from each subregion is not guaranteed to discover the lowest (highest) functional value existing over the subregion. That is, ‘the uncertainty arises from the limitations of the evidence gathering system’ (Pal and Bezdek 1994). A review of uncertainty measures for evidential reasoning is provided by Pal et al. (1992, 1993).

In FRACTOP, the search tree is managed by Bel(α_{it}) which incorporates genetic evidence as well as fresh evidence collected. Bel(α_{it}) behaves like autoregressive models in time series and the effects of genetic evidence gathered from levels prior to level t decrease exponentially. Although the effect of parental evidence dies out exponentially, it still has an influence on the selection of the most promising subregion. In our experiments, which were conducted with γ ranging from 0.0 to 1.0, we observed that the best results are obtained with a value lying within the interval 0.1 and 0.2. Genetic evidence is used for the following reason. Since the evidence gathering procedure has no guarantee of locating the best solution in each subregion, a region which seemed to be promising in a previous level, may turn out to be totally unfavourable during the re-sampling process at the current level of the search. The genetic evidence then becomes influential and may prevent the elimination of a potential container of the global optimum.

In FRACTOP, the management of the search tree and the imbedded geometric partitioning mechanism accord with the following three axioms stated by Ross (1995, p. 559), for belief measures. Here, we demonstrate how these axioms fit in FRACTOP's method of conducting the search.

Suppose that subregion α_{it} at level t and its parent region α'_{it} contain $\mathbf{x}^*$.

(i) Bel(ϕ) = 0 and Bel(S) =1, where ϕ and S represent the case of no evidence and the case of complete evidence, respectively. Thus, the first axiom is satisfied.

(ii) $\text{Bel}(\alpha'_{it}) \leq \text{Bel}(\alpha_{it} \cup \alpha'_{it})$, where $\text{Bel}(\alpha_{it} \cup \alpha'_{it})$ denotes the information obtained from the united samples ($A_{it} \cup A'_{it}$) and Bel(α'_{it}) denotes the information obtained from the sample A'_{it}. Since α_{it} covers a smaller region around $\mathbf{x}^*$, the information gathered is relatively more certain than that of the parent region's, α'_{it}. Thus, the second axiom holds, because $A'_{it} \subseteq (A_{it} \cup A'_{it})$. In Figure 3, we observe the inverse relationship between the size of subspaces and fuzziness.

(iii) $\text{Bel}(\alpha'_{it} \cup \alpha_{it}) = \text{Bel}(\alpha'_{it}) + \text{Bel}(\alpha_{it}) - \text{Bel}(\alpha'_{it} \cap \alpha_{it})$ holds, because $\text{Bel}(\alpha'_{it} \cup \alpha_{it}) = \text{Bel}(\alpha_{it})$ and $\text{Bel}(\alpha'_{it} \cap \alpha_{it}) = \text{Bel}(\alpha'_{it})$ due to the second axiom.

If the search is terminated at a given level of the tree, then each region's belief value represents the evidence of containing the global optimum. From this point

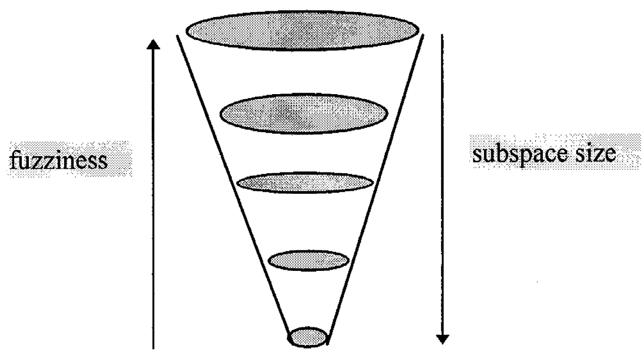

Figure 3. The relationship between fuzziness and subspace size.

on, the user may continue his search for the global optimum by restricting the feasible region with the bounds of the region(s) with the highest belief value(s) and start applying a procedure which utilizes information embedded in the function optimized, such as the gradient. Thus, FRACTOP might be used as a preliminary fast method to locate the whereabouts of the global optimum.

4. Methods of evidence collection

There are numerous methods to gather evidence from each subregion. For demonstration purposes, here we utilize RS, SA and GA in evaluating subregions. However, the latter three methods are not the only possible evidence gathering tools. Gradient-based methods or MLSL could also be used to provide evidence about a geometric region. Furthermore, a different method can be selected to evaluate each subregion, because the belief measure involves functional evaluations on an average basis. For instance, in the initial level of the search, MLSL can be used to evaluate large chunks of the feasible space. Then, medium-sized subregions could be handled by the GA and then finer regions could be investigated by SA or the steepest descent method. Here, once we specify the evidence gathering method, we stick to it until the end, so that we can have a fair comparison of each method in stand alone mode versus being imbedded in FRACTOP. In the following we indicate briefly how we apply RS, SA and GA.

Random sampling (RS). Each subregion is evaluated by randomly taking s solutions within its bounds and using their functional evaluations in the belief measure.

Simulated annealing (SA). As in RS, each subregion is evaluated by randomly taking s solutions within its bounds. However, these solutions are regarded as the initial starting points from which SA begins neighbourhood search. To every random starting point, a prespecified number of SA moves (which is reduced according to the size of the subregion) are applied. In a SA move, first, a dimension in which the solution will be perturbed is selected randomly. Then, a stepsize (positive/negative) which will not push the current solution out of regional bounds is selected randomly. The new solution is evaluated. If there is an improvement in the current best functional evaluation, the solution is accepted. Else, the solution is accepted according to a probability of acceptance, PA, defined by

$$\mathrm{PA}(F\mathrm{Dif}, t\mathrm{SA}) = \exp(-F\mathrm{Dif}/(F_c t\mathrm{SA}))$$

where, F_c is the functional evalution of the current solution, FDif is the difference between the functional evaluations of the new solution and the latest solution, and tSA is the temperature. If a randomly generated number between zero and one turns out to be less than PA, then the deteriorating move is carried out.

tSA depends on the number of times a deteriorated cost has been obtained consecutively. Initially tSA is equal to one, but after each non-improving move, the temperature, tSA, is reduced as follows.

$$t\mathrm{SA} \leftarrow t\mathrm{SA}/(1 + \beta t\mathrm{SA}),$$

where β is a nonnegative constant less than one (preferably between 0.05 and 0.1).

Once all SA moves have been carried out for an initial random solution, the best solution found during these iterations replace the initial starting solution and becomes one of the sample solutions to represent the subregion in the belief measure.

Genetic algorithm (GA). Each subregion is evaluated by randomly taking s solutions within its bounds. These solutions constitute the initial population for GA. A float encoding used in the GA, that is, each chromosome has n genes, each consisting of a real number which indicates the value of the nth variable.

The reproduction opeator works as follows. Each chromosome is assigned a reproduction probability equal to its relative functional value, RF_j $(= (F^* - F_j)/F^* - F_{\text{worst}})$ in the population. Then, during a single pass over all chromosomes, a chromosome is picked up for the next generation randomly if a random number less than one turns out to be less than RF_j.

After reproducing the parent population, a single pass is executed over all chromosomes in order to identify the set of chromosomes, C, to undergo crossover. A chromosome's crossover probability is again equal to RF_j. Note that RF_j depends on both the chromosome's performance and the population's performance. Adaptive crossover and mutation rates such as RF_j are demonstrated to work well with multi-modal functions (Srinivas and Patnaik 1994) and they have been also tested in a difficult combinatorial optimization problem (Özdamar 1998). Next, a pair of chromosomes are randomly selected from C to result in an offspring. The offspring's gene values result from the convex combinations of the corresponding parent genes. Offspring are generated until the set C becomes empty. The offspring replace the chromosomes with the worst functional values in the previous generation.

Since the population always remains within the convex hull specified by the boundary limits of the initial random population, mutation is applied to all the genes of the chromosomes selected for mutation to maintain diversity in the population. The mutation probability of a chromosome is equal to RF_j which is recalculated after crossover. A single pass over the new population identifies the chromosomes that are to be mutated. In mutation, every gene's value is either decreased or increased (which is a random decision) by a randomly determined amount which respects the bounds of each dimension in the considered subregion.

4.1. AN EXAMPLE

The following example demonstrates how FRACTOP converges to an optimum solution. Consider the function $f(x, y) = \sin(x)\sin(y)$ in Figure 4. This function has multiple optimal solutions in the intervals, $1 \leq x \leq 5$ and $-2 \leq y \leq 6$, at coordinates ($\pi/2$, $\pi/2$), ($3\pi/2$, -$\pi/2$) and ($3\pi/2$, $3\pi/2$), or approximately, at (1.57, 1.57), (4.71, −1.57) and (4.71, 4.71), respectively, each with $f(x, y) = 1$.

Figure 5 demonstrates the contours of the objective function with a projection to the coordinate system. An optimum solution is reached with an accuracy of

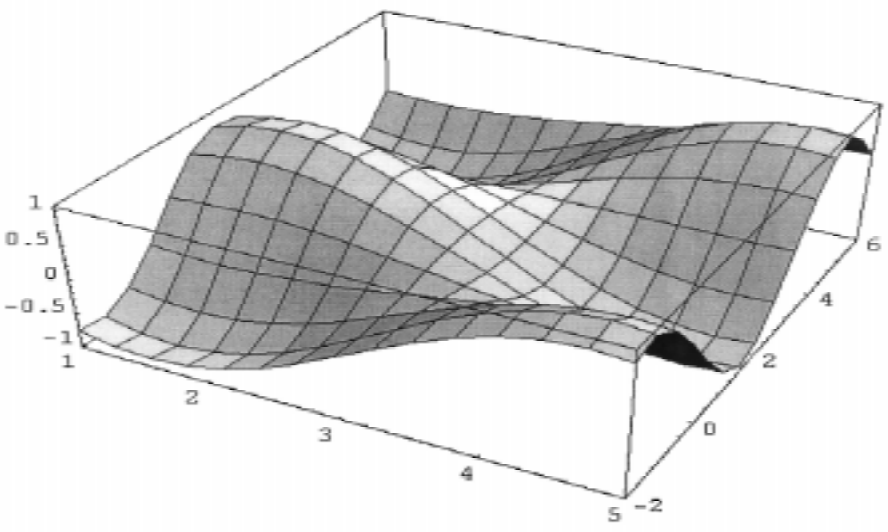

Figure 4. Example function.

10^{-6}at the fourth level. The region that contains the coordinates ($\pi/2$, $\pi/2$) is selected repeatedly at each level of the tree until the size of the selected region becomes quite small, and the search converges to a region whose boundaries are (1.5, 1.625)$\times$(1.5, 1.75). Note that the crossed out regions in Figure 5 are discarded at the first level, since their belief values are small with respect to that of the re-partitioned region. The other two regions (containing the other two minima) that are not re-partitioned, are not discarded. Therefore, once the region containing ($\pi/2$, $\pi/2$) becomes too narrow, the search will locate the other minima if permitted to go on. Naturally if the user wishes a quicker convergence he/she may use a discarding ratio of higher value and will end up with a smaller number of evaluations.

5. Possible extensions of FRACTOP

Constrained optimization. FRACTOP is easily adaptable to constrained optimization. Similar to the case of simple bounded variables discussed here, FRACTOP would be initiated by identifying a closure of the feasible region, $\mathcal{G}$. However, the closure would not be minimal. Identifying the minimal closure of an n-dimensional feasible space specified by a set of nonlinear equations $g(\mathbf{x})$, is a research subject in itself and therefore, a simple closure identification scheme is adopted here. If the search is conducted in the n-dimensional space, for each variable x_i, 1, ... , n, i = 1, ... , n, each equation in $g(\mathbf{x})$ is solved where the remaining n-1 variables x_j are set to 0, $j \neq i$. The lower and upper bounds (LB_i & UB_i) are thus calculated for x_i. A natural outcome of this method is that infeasible solutions are included in the closure. Infeasibility can be dealt with using different approaches. The simplest approach is to ignore the infeasible solutions in the sample and calculate the belief

Table 1. The iterations of FRACTOP.

Level (t)	Region $\alpha(i,t)$	Regional bounds $(x)\times(y)$	Belief of regions Bel(α_{it})	Best value obtained (F^*)	Decision on regions	No. of evaluations
0	1,0	(1,3)×(2,6)	0.286203	0.967555	preserved	40
	2,0	**(1,3)×(−2,2)**	**0.504116**		**selected**	
	3,0	(3,5)×(2,6)	0.192677		preserved	
	4,0	(3,5)×(−2,2)	0.365606		preserved	
1	1,1	(1,3)×(2,6)	0.147354	0.984023	discarded	70
	2,1	**(1,2)×(0,2)**	**0.745949**		**selected**	
	3,1	(1,2)×(−2,0)	0.150126		discarded	
	4,1	(2,3)×(0,2)	0.428105		preserved	
	5,1	(2,3)×(−2,0)	0.178990		discarded	
	6,1	(3,5)×(2,6)	0.266319		preserved	
	7,1	(3,5)×(−2,2)	0.220976		preserved	
2	1,2	(1,1.5)×(1,2)	0.869372	0.996453	preserved	70
	2,2	(1,1.5)×(0,1)	0.462551		preserved	
	3,2	**(1.5,2)×(1,2)**	**0.896172**		**selected**	
	4,2	(1.5,2)×(0,1)	0.537004		preserved	
	5,2	(2,3)×(0,2)	0.459380		preserved	
	6,2	(3,5)×(2,6)	0.303752		preserved	
	7,2	(3,5)×(−2,2)	0.291175		preserved	
3	1,3	(1,1.5)×(1,2)	0.919032	0.998634	preserved	100
	2,3	(1,1.5)×(0,1)	0.435450		preserved	
	3,3	**(1.5,1.75)×(1.5,2)**	**0.944974**		**selected**	
	4,3	(1.5,1.75)×(1,1.5)	0.919015		preserved	
	5,3	(1.75,2)×(1.5,2)	0.905079		preserved	
	6,3	(1.75,2)×(1,1.5)	0.896526		preserved	
	7,3	(1.5,2)×(0,1)	0.435912		preserved	
	8,3	(2,3)×(0,2)	0.421746		preserved	
	9,3	(3,5)×(2,6)	0.394266		preserved	
	10,3	(3,5)×(−2,2)	0.315258		preserved	
4	1,4	(1,1.5)×(1,2)	0.741136	0.999997	preserved	130
	2,4	(1,1.5)×(0,1)	0.545157		preserved	
	3,4	**(1.5,1.625)x(1.5,1.75)**	**0.982319**		**selected**	
	4,4	(1.5,1.625)×(1.75,2)	0.930914		preserved	
	5,4	(1.625,1.75)x(1.5,1.75)	0.932722		preserved	
	6,4	(1.625,1.75)x(1.75,2)	0.976587		preserved	

Table 1. (*Continued*)

Level (t)	Region $\alpha(i,t)$	Regional bounds $(x)\times(y)$	Belief of regions Bel(α_{it})	Best value obtained (F^*)	Decision on regions	No. of evaluations
	7,4	(1.5,1.75)x(1,1.5)	0.926896		preserved	
	8,4	(1.75,2)x(1.5,2)	0.775833		preserved	
	9,4	(1.75,2)x(1,1,5)	0.877272		preserved	
	10,4	(1.5,2)x(0,1)	0.599215		preserved	
	11,4	(2,3)x(0,2)	0.482069		preserved	
	12,4	(3,5)x(2,6)	0.221382		preserved	
	13,4	(3,5)x(−2,2)	0.347852		preserved	

Example information: Number of samples taken from each region: 10 (fixed throughout the search) Genetic information value: 0.25 Region discarding criterion: Bel(α_{ij})/ Bel(α^*_{ij}) < 0.25 Number of levels permitted in the search tree: 4 Total number of evaluations: 410.

measure using the feasible data only. If the evidence collection method does not identify any feasible solutions within a subregion, the search might give that subregion another chance in the next level of the tree or discard it immediately. This approach can be supported by corrective actions applied to an infeasible solution once it is detected. If a solution involves a small degree of infeasibility, corrective actions might push the solution into the feasible region by slightly perturbing the values of some variables.

A nice property of geometric partitioning is that it is easy to treat soft constraints. The right-hand sides of the constraints may be expanded as required and a penalty based on fuzzy membership values of the right hand sides may be added to the objective function value of each solution.

Discrete variables. The system to be optimized may include binary (0–1) or general integer discrete variables in the objective function as well as in the constraints. The resulting model then becomes a nonlinear mixed integer (binary) model (MINLP). The constraints may be nonlinear or linear. The re-classification of the variables does not affect the procedural structure of FRACTOP. There would simply be an additional classifier in the database for each variable. All evaluation techniques are based on an initial random sample and in this case, some variables would be subjected to an integer restriction while being generated. Then, the feasibility of the discrete variables would be preserved in further applications (SA, GA, etc.).

In some practical combinatorial optimization problems which involve both real and binary variables and linear constraints (Özdamar and Birbil 1998; Tempelmeier and Dersdorf 1996) it is not possible to conceive a polynomial algorithm for identifying a feasible solution from which a search procedure can start (not only the identification of the optimal solution is NP-Complete, but also the constraint satis-

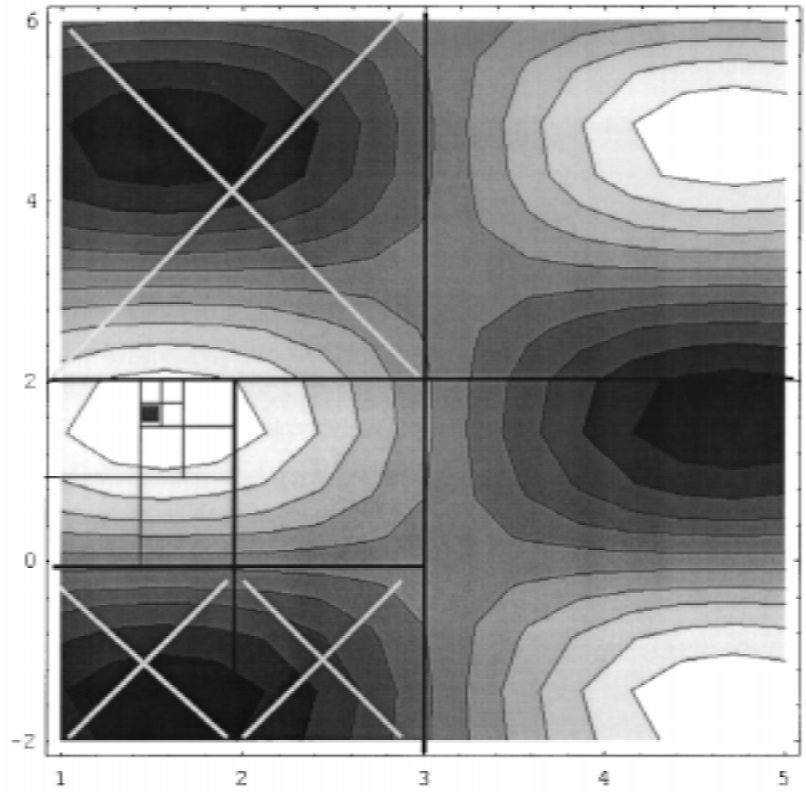

Figure 5. Objective function contours of the example function and regional partitions carried out by FRACTOP.

faction problem). The geometric partitioning structure in FRACTOP facilitates the issue of constraint satisfaction in hard problems, because it is a systematic method which sweeps over the initial (partly infeasible) closure.

6. Computational experience with FRACTOP

Thirteen nonlinear functions are selected for the computational experiments. The first six functions (the fifth one has two variants, the first one with three variables and the second one with four) are selected by Androulakis and Vrahatis (1996) for comparing the convergence rates of different gradient-based numerical search methods. The next three functions (the third one has three variants with the number of variables equal to 2, 4 and 5, respectively, with each variant having increased epistacity) are used by Srinivas and Patnaik (1994) in order to compare their adaptive GA, AGA (which involves adaptive crossover and mutation rates), with simple GA, SGA. The last function is a spiky function used in Michalewicz (1994) to show the mechanics of a GA. The mathematical expressions for all functions and their corresponding references are given in the Appendix.

In order to demonstrate the improvement in the performance of the search methods imbedded in FRACTOP, we compare the results obtained by FRACTOP + search method with the ones obtained by the stand alone application of the specific search method. An equal number of evaluations are permitted for both applications. We compare the results by indicating the average (standard deviation) objective function value obtained in 100 runs, each carried out with a different random seed. The best (worst) objective function values obtained in 100 runs are also reported. FRACTOP is permitted to continue until at most the tenth level of the tree or until no more improvement is observed as compared to the preceding level. The stand alone search method to which the results of FRACTOP are to be compared is permitted to execute the same number of functional evaluations as FRACTOP. In Table 2, all performance measures are provided for FRACTOP+RS and stand alone RS, as well the average number of functional evaluations executed by each of the 100 runs. The results for FRACTOP+GA and GA, and the results for FRACTOP+SA and SA are given in Tables 3 and 4, respectively. The number of samples, s, collected from each region is 100 at level zero in FRACTOP+RS and FRACTOP+GA (s indicates population size in GA). In the next levels, s is decreased in the proportion of the size of the considered subregion to the size of the initial closure $\mathcal{G}$. However, s is not permitted to be below 30. Regardless of the size of the subregion, the GA runs for 20 generations. As for FRACTOP+SA, $s = \min\{2^n, 12\}$ and is constant at all levels of the tree. However, the number of SA moves are decreased in proportion of the size of the subregion to the size of the initial closure, starting at 100 at level zero. In all applications, a subregion is discarded if ratio of its belief measure to the maximum belief measure is less than 80%.

The computational experiments demonstrate that FRACTOP improves considerably all three search methods imbedded in it. The performance of FRACTOP+SA is somewhat inferior to that of SA in the last two functions and their variants (#10..13) which require a higher accuracy in the size of the subregions over which the search is conducted. Namely, the sinusodial movement of these functions take place in a very narrow range and FRACTOP+SA requires to execute more partitioning levels with reduced sample sizes. On the other hand, stand alone SA ploughs its way through ‘spikes’ because the spikes have a positive trend towards the upper corner (with coordinates 12.1 and 5.8) of the feasible region.

In this experiment, we are not particularly interested in the search method which provides the best results. Rather, the emphasis lies on the fact that FRACTOP, as a geometric partitioning metaheuristic, disposes of the numerous disadvantages pertaining to the search metaheuristics previously proposed for global optimization and improves their results.

A further remark on FRACTOP is on its convergence to the small area containing the global optimum. For instance, in Table 4, in functions no. 1, 4, 5, 6,

Table 2. Comparison of results (FRACTOP+RS and stand alone RS).

		FRACTOP+RS				RS			
Func #	# of func. evaluations	Average	Standard dev.	Worst	Best	Average	Standard dev.	Worst	Best
1	1240	0.000052	–	0.0008	0	0.0169	0.0002	0.0685	0.00007
2	1112	0.00588	0.0002	0.0943	0	0.0291	0.001	0.1552	0.0006
3	1306	0.0194	0.007	0.688	0.00002	0.3375	0.1296	2.3321	0.0002
4	2330	0.478	0.24	2.905	0.0004	1.5262	0.9132	4.0427	0.0312
5	3388	1.0005	0.000001	1.0061	1.00001	1.0134	0.0002	1.0843	1.0002
6	6931	1.0024	0.000159	1.1153	1.00006	1.1336	0.0099	1.4212	1.0072
7	4606	0.0514	0.0101	0.2951	0.0001	0.4001	0.523	1.2231	0.0057
8	1808	0.0102	0.000002	0.1866	0.0048	0.0711	0.0022	0.1964	0.0097
9	1163	0.87	0.05	1.3791	0.159	2.1937	0.3732	3.8263	1.0142
10	1000	0.0038	0.000004	0.0101	0.00005	0.5895	0.1775	1.9124	0.0476
11	4523	0.0157	0.00003	0.0338	0.0041	0.1012	0.0007	0.1825	0.0317
12	12324	0.00452	0.000026	0.0191	0.0012	1.3794	0.8624	2.015	0.253
13	2941	38.18	0.178	37.15	38.849	37.76	0.1832	36.69	38.69

Standard dev. = ’-’ means standard deviation is less than 10^{-6}.

Table 3. Comparison of results (FRACTOP+GA and stand alone GA).

		FRACTOP+GA				GA			
Func #	# of func. evaluations	Average	Standard dev.	Worst	Best	Average	Standard dev.	Worst	Best
1	100137	0.000001	–	0.00004	0	0.0011	0.000001	0.0061	0.000007
2	105063	0.000003	–	0.00009	0	0.0024	0.000005	0.0113	0.00001
3	105825	0.000025	–	0.0005	0	0.0136	0.00025	0.0791	0.0003
4	204750	0.0078	0.0436	0.3024	0.000001	0.0986	0.0061	0.4444	0.0037
5	336334	1.00005	–	1.0009	1	1.0003	–	1.0017	1.000001
6	493555	1.0047	0.00009	1.0559	1.000001	1.0078	0.000028	1.0318	1.00019
7	356099	0.0152	0.0020	0.2495	0	0.0202	0.00014	0.0548	0.0021
8	102622	0.00038	-	0.0035	0.000001	0.0055	0.000014	0.0097	0.000004
9	92743	0.0643	0.0009	0.1474	0.0152	0.2173	0.00777	0.4732	0.0612
10	97745	0.00005	-	0.0003	0.000003	0.0027	0.000002	0.0076	0.0003
11	393452	0.0006	0.000001	0.0060	0.00002	0.0100	0.00002	0.0231	0.0020
12	801364	0.005	0.00007	0.0492	0.0001	0.0123	0.000045	0.0350	0.0024
13	173072	38.82	0.0189	38.51	38.85	38.79	0.00348	35.54	38.849

Table 4. Comparison of results (FRACTOP+SA and stand alone SA).

		FRACTOP+SA				SA			
Func #	# of func. evaluations	Average	Standard dev.	Worst	Best	Average	Standard dev.	Worst	Best
1	4866	0	0	0	0	0.00002	-	0.0001	0
2	4484	0.000008	–	0.0004	0	0.0046	0.0001	0.067	0.000001
3	4888	0.0004	0.000005	0.0166	0	0.0008	0.000002	0.0062	0.000003
4	3281	0	0	0	0	1.978	0.842	2.8231	0.00008
5	3082	1	0	1	1	1.0006	0.000001	1.0042	1.000007
6	6802	1	0	1	1	1.0037	0.00001	1.0139	1.00008
7	37836	0.0002	–	0.0021	0.000009	0.00001	–	0.0002	0
8	2115	0	0	0	0	0.0192	0.0002	0.0784	0.0007
9	866	0.00004	-	0.0005	0	0.9251	0.119	2.0318	0.2381
10	7357	0.0004	-	0.0017	0.00004	0.0003	–	0.0012	0.00001
11	54796	0.0036	0.00002	0.018	0.0002	0.0002	–	0.0007	0.00004
12	101964	0.0131	0.00024	0.068	0.001	0.0003	–	0.0007	0.00008
13	10165	38.75	0.0177	38.52	38.85	38.80	0.0016	38.73	38.85

8, FRACTOP has converged to the exact location of the global optimum in 100% of the 100 runs. In the spiky function (function no.13), although FRACTOP does not seem to improve the performance of stand alone SA, the bounds of the region that FRACTOP finally converges to, includes first variable of the global optimum value x_1^*=11.62523 in 100 out of 100 runs, and in 91 out of 100 runs, it includes the global solution's second variable value x_2^*=5.725082. The boundaries of the converged subregion are very tight (0.2 and 0.1, on the average, for first and second variables, respectively).

7. Conclusion

We develop a geometric partitioning metaheuristic, FRACTOP, which is an expansive geometric method in which any search approach suggested for global optimization can be embedded. The aim of FRACTOP is to eliminate some of the disadvantages existing in stochastic search techniques, such as eliminating revisits to the same area of the feasible region and providing the correct sequence of diversification and intensification required for converging to the global optimum. Intelligent guidance to lead the search into promising regions is provided by evaluating and comparing nonoverlapping regions using fuzzy measures.

Although the current partitioning scheme used in FRACTOP is exponential in the number of variables, this issue does not pose a serious problem, because any linear partitioning scheme might have been used. Future work will include the development of an efficient and intelligent linear partitioning scheme as well as applying FRACTOP to constrained optimization and mixed integer nonlinear problems.

Appendix

1. Complex Function (Press et al. 1992), $n = 2$

$$f_1(x_1, x_2) = (x_1^3 - 3x_1x_2^2 - 1)^2 + (3x_1^2x_2 - x_2^3)^2, \quad -2 \le x_i \le 2, \quad i = 1, 2.$$

This function has three minima $\mathbf{x}_1^*$ = (1,0), $\mathbf{x}_2^*$ = $(-\frac{1}{2}, \sqrt{1/2})$ and $\mathbf{x}_3^*$ = $(-\frac{1}{2}, \sqrt{3/2})$ with f($\mathbf{x}_i^*$) = 0, i = 1,2,3.

2. Stenger Function (Stenger 1975), n = 2

$$f_2(x_1, x_2) = (x_1^2 - 4x_2)^2 + (x_2^2 - 2x_1 + 4x_2)^2, -1 \le x_i \le 4, i = 1, 2.$$

This function has two minima $\mathbf{x}_1^*$=(0, 0) and $\mathbf{x}_2^*$ = (1.695415, 0.7186082) with $f(\mathbf{x}_i^*)$ = 0, i= 1, 2

3. Himmelblau Function (Botsaris 1978), n = 2

$$f_3(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2, -6 \le x_i \le 6, i = 1, 2.$$

This function has four minima $\mathbf{x}_1^*$= (3,2), $\mathbf{x}_2^*$=(−2.805118, 3..131312), $\mathbf{x}_3^*$=(3.584428, −1.848126) and $\mathbf{x}_4^*$=(−3.779310, −3.283186) with $f(\mathbf{x}^*)$ = 0

4. Hellical Valley Function (More et al. 1981), n = 3

$$f_4(x_1, x_2, x_3) = 100(x_3 - 10\theta(x_1, x_2)^2 + 100(\sqrt{x_1^2 + x_1^2} - 1)^2 + x_3^2$$
$$-2 \le x_1 \le 4, \quad \text{and} \quad -2 \le x_i \le 2, \quad i = 2, 3.$$

$$Q(x_1 x_2) = \begin{cases} \frac{1}{2\pi} \arctan(\frac{x_2}{x_1}) & \text{for } x_1 > 0 \\ \frac{1}{2\pi} \arctan(\frac{x_2}{x_1}) + 0.5 & \text{for } x_1 < 0 \end{cases}$$

This function has one minimum: $\mathbf{x}^* = (1, 0, 0)$ with $f(\mathbf{x}^*) = 0$

5. Brown Almost Linear Function, for n=3 (More et al. 1981)

$$f_5(x_1, x_2, x_3) = \sum_{i=1}^{n} g_i^2(x_1, x_2, x_3), \qquad -2 \le x_i \le 4, \quad i = 1, 2, 3,$$

where

$$g_i(x_1, \ldots, x_n) = x_i + \sum_{j=1}^{n} x_j - (n+1), \quad i = 1, \ldots n-1,$$
$$\text{and } g_n(x, \ldots, x_n) = \left(\Pi_{j=1}^{n} x_j\right) - 1$$

The minimum of this function is at $\mathbf{x}_i^* = 1$ for i = 1,2, . . . ,n with f($\mathbf{x}^*$) = 1.

6. Brown Almost Linear Function, for n=4 (More et al. 1981) has the same optimum.

7. Extended Kearfott Function, n=4 (Kearfott 1979; Vrahatis 1988)

$$f_7(x_1, x_2, x_3, x_4) = (x_1^2 - x_2^2)^2 + (x_2^2 - x_3^2)^2 + (x_3^2 - x_4^2)^2 + (x_4^2 - x_1^2)^2,$$
$$\text{where } -3 \le x_i \le 10, i = 1, 2, 3, 4.$$

This function has two minima $\mathbf{x}_1^*$ = (1,1,1,1) and $\mathbf{x}_1^*$ = (0,0,0,0) with $f(\mathbf{x}_i^*)$ = 0, i=1,2.

8. The Sine Envelope Sine Wave Function, n = 2

$$f_8(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{1.0 + 0.001(x_1^2 + x_2^2)]^2}, \quad -100 \le x_i \le 100, \quad i = 1, 2$$

the optimum value is $\mathbf{x}^*$ = (0,0) with $f(\mathbf{x}^*)$ = 0.

9. Function of Davis (1987), n=2

$$f_9(x_1, x_2) = (x_1^2 + x_2^2)^{0.25}[sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1.0]$$
$$-100 \le x_i \le 100, i = 1, 2$$

the optimum value is $\mathbf{x}*$ = (0,0) with $f(\mathbf{x}^*)$ = 0.

10. Epistacity Test Function by Srinivas and Patnaik (1994), n=2

$$f_{10}(x_i) = \sum_{i=1}^{2} \frac{\sin(\pi k x_i)}{(\pi k x_i)}, \quad -0.5 \le x_i \le 0.5, \quad i = 1, 2$$

the optimum value is $\mathbf{x}^* = (0,0)$ with $f(\mathbf{x}^*) = 0$.

11. Epistacity Test Function by Srinivas and Patnaik (1994), n=4, has the same optimum as (#10).

12. Epistacity Test Function by Srinivas and Patnaik (1994), n=5, has the same optimum as (#10).

13. The Spiky Function (Michalewicz 1994)

$$f_{13}(x_1, x_2) = 21.5 + x_1 \sin(4\Pi x_1) + x_2 \sin(20\Pi x_2) \qquad -3 \leq x_1 \leq 12.1, \quad 4.1 \leq x_2 \leq 5.8.$$

The optimum value is $\mathbf{x}^*$=(11.62523, 5.72082) with $f(\mathbf{x}^*)$ 38.85.

References

Androulakis, G.S. and Vrahatis, M.N. (1996), OPTAC: a portable software package for analyzing and comparing optimization methods by visualization, *Journal of Computational and Applied Mathematics* 72: 41–62.

Armijo, L. (1966), Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific J. of Mathematics* 16: 1–3.

Beveridge, G. and Schechter, R. (1970), *Optimization: Theory and Practice*, Mc-Graw Hill, New York.

Botsaris, C.A. (1978), A curvilinear optimization method based upon iterative estimation of the eigensystem of the hessian matrix, *J. of Mathematical Analysis and Applications* 63: 396–411.

Bozyel, M.A. and Özdamar, L. (1997), A heuristic approach to the capacitated lot sizing and scheduling problem (with parallel facilities), Working Paper, Istanbul Kültür University, Dept. of Computer Engineering, Istanbul.

Caprani, O., Gothaab, B. and Madsen, K. (1993), Use of real-valued local minimum in parallel interval global optimization, *Interval Computations*, 3: 71–82.

Csendes, T. and Pinter, J. (1993), The impact of accelerating tools on the interval subdivision algorithm for global optimization, *European Journal of Operations Research* 65, 314–320.

Davis, L. (ed.) (1987), *Genetic Algorithms and Simulated Annealing*, London, Pitman.

Fletcher, R. and Powell, M. (1963), A rapidly convergent descent method for minimization, *Computer Journal* 6: 163–168.

Fletcher, R. and Reeves, C. (1964), Function minimization by conjugate gradients, *Computer Journal* 7: 149–154.

Gill, P. and Murray, W. (1972), Quasi-Newton methods for unconstrained optimization, *J. Institute of Mathematics and its Applications*, 9: 91–108.

Glover, F. (1989), Tabu search – Part I, *ORSA Journal of Computing* 1: 190–206.

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

Kan, A.H.G. and Timmer, G.T. (1984), Stochastic methods for global optimization, *American J. of Mathematical Management Science* 4: 7–40.

Kearfott, B., An efficient degree-computation method for a generalized method of bisection, *Numerische Mathematics* 32: 109–127.

Kirkpatrick, A., Gelatt, Jr., C.D. and Vechi, M.P. (1983), Optimization by simulated annealing, *Science* 220: 671–680.

Klir, G. and Folger, T. (1988), *Fuzzy Sets Uncertainty and Information*, Prentice Hall, Englewood Cliffs, N.J.

Michalewicz, Z. (1994), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, Berlin.

More, B.J., Garbow, B.S. and Hillstrom, K.E. (1981), Testing unconstrained optimization, *ACM Trans. Math. Software* 7: 17–41.

Özdamar, L. (1998), A genetic algorithm approach for the multi-mode resource-constrained project scheduling problem under general resource categories, to appear in *IEEE Trans. On Systems, Man and Cybernetics*.

Özdamar, L. and Birbil, Ş.I. (1998), A hybrid genetic algorithm for the capacitated lot sizing and loading problem, to appear in *European Journal of Operations Research*.

Özdamar, L. and Bozyel, M.A. (1998), Simultaneous lot sizing and loading of product families on parallel facilities of different classes, *Int. J. Production Research*, 36, 1305–1324.

Pal, N.R. and Bezdek, J.C. (1994), Measuring fuzzy uncertainty, *IEEE Transactions on Fuzzy Systems* 2, 107–118.

Pal, N.R., Bezdek, J.C. and Hemasinha, R. (1992), Uncertainty measures for evidential reasoning I: A review, *Int. J. of Approximate Reasoning* 7: 165–183.

Pal, N.R., Bezdek, J.C. and Hemasinha, R. (1993), Uncertainty measures for evidential reasoning II: A review *Int. J. of Approximate Reasoning*, 8, 1–16.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992), *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, New York.

Ross, T.J. (1995), *Fuzzy Logic with Engineering Applications*, Mc-Graw Hill.

Schaffer, J.D. (1989), A study of control parameters affecting online performance of genetic algorithms for function optimization, *Proc. of the Third Int. Con. on Genetic Algorithms* 51-60.

Srinivas, M. and Patnaik, L.M. (1994), Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics* 24: 656–667.

Stenger, F. (1975), Computing the topological degree of a mapping in $\mathcal{R}^n$, *Numerische Mathematics* 25: 23–38.

Tempelmeier, H. and Derstroff, M. (1996), A Lagrangean based heuristic for dynamic multi item multi level constrained lot sizing with setup times, *Management Science* 42: 738–757.

Törn, A. and Viitanen, S. (1994), Topographical global optimization using pre-sampled points, *Journal of Global Optimization* 5: 267–276.

Vrahatis, M.N. (1988), Solving systems of nonlinear equations using the nonzero value of the topological degree, *ACM Trans. Math. Software* 14: 312–329.

Yager, R. (1983), Entropy and specificity in a mathematical theory of evidence, *Int. J. of General Systems* 9: 149–260.

Zimmermann, H. (1991), *Fuzzy Set Theory and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands.