The repulsion algorithm, a new multistart method for global optimization

Article *in* Structural and Multidisciplinary Optimization · June 1996

DOI: 10.1007/BF01197028

2 authors:

Abdon Sepulveda
University of California, Los Angeles

Leonardo David Epstein
University of the Andes (Chile)

Structural Optimization 11, 145–152 © Springer-Verlag 1996

J. 15

The repulsion algorithm, a new multistart method for global optimization

A.E. Sepulveda

Department of Mechanical, Aerospace, and Nuclear Engineering, University of California, Los Angeles, CA 90095, USA

L. Epstein

Department of Biostatistics, The Johns Hopkins University, Baltimore, MD 21205, USA

Abstract This paper proposes a new multistart algorithm to find the global minimum of constrained problems. This algorithm, which in this paper is called the repulsion algorithm, efficiently selects initial design points for local searches. A Bayesian approach provides the stopping rules. The method uses information from the previous sampling points and the corresponding sequences generated by local searches to select new initial points. This approach increases the probability of finding all local minima with fewer local searches. Numerical example problems show that compared with traditional multistart methods, the repulsion algorithm reduces significantly the number of local searches required to find the global minimum.

1 Introduction

In structural and mutidisciplinary design optimization as well as in highly nonlinear problems and mechanical design problems, it is common to find either disjoint or nonconvex design spaces. Examples are grillage structures for the static case (Kavlie and Moe 1971, Moses and Onoda 1969) and structures subject to dynamic loads. For the steady state case, Johnson (1976) and Mills-Curran and Schmit (1985) show that near resonance conditions usually produce disjoint feasible regions with large dynamic displacements, which are highly nonlinear functions of the design variables. Cassis and Schmit (1976) and Sepulveda and Jin (1992) present problems that involve transient behaviour constraints with severe nonconvexities and disjoint feasible regions. For these problems, where the objective function values at distinct local minima may be substantially different, an algorithm that converges to a local minimum is not satisfactory.

This article introduces the repulsion algorithm, a new scheme for approximately solving the global optimization problem,

$$\text{find } \hat{x} \quad \text{such that } \hat{f} = \min_{x \in K} f(x), \tag{1}$$

where f is the objective function, and K is the set of the feasible designs.

There are two kinds of algorithms to solve the optimization problem equation (1) states, namely deterministic and stochastic. Deterministic algorithms guarantee convergence to the global optimum only for certain classes of objective functions and/or constraint functions. Stochastic algorithms randomly generate points in the design space to initiate local searches. Stochastic algorithms guarantee convergence to the global minimum only asymptotically, that is roughly speaking, the probability that the algorithm finds the global minimum converges to one as the number of local searches tends to infinity. The repulsion algorithm is of the stochastic kind.

A brief review of the essentials of multistart methods is presented. A description of the repulsion algorithm will follow. Mutistart methods enumerate all local minima with local optimizations initiated from a set of random points distributed uniformly over the region K. The basic steps are as follows (see for example, Rinnooy-Kan and Timmer 1986).

(1) Select x at random from a uniform distribution on K.
(2) Starting from x, use a local optimizer to find the local minimum x^*.
(3) Check stopping rules. If a termination criterion is satisfied, stop sampling and the local minimum with the lowest objective function value estimates the global minimum. If the termination criterion is not satisfied, go to step 1.

The repulsion algorithm is a multistart method, as it tries to find all local minima starting local searches from a number of different points in K. A review of the literature indicates that only the uniform distribution has been used to generate the starting points. In contrast, the repulsion algorithm does not use this distribution, but instead it conveniently changes the distribution after each local search. The sequential change of the sampling distribution attempts to avoid spending local searches in regions where it is unlikely to find a new local minimum, i.e. it attempts to reduce the number of times that step 2 is executed.

In global optimization the number of local minima is usually unknown but assumed to be finite. Multistart methods sequentially generate points in K and use them to start local searches. The problem of deriving stopping rules then arises, that is, of deciding when to stop the sequence of local searches. Multistart methods do not necessarily find the global minimum, but the probabilistic structure provided by the sampling distribution for the starting points allows one to make inferences about the number of local minima. One then uses these inferences to determine when to stop the sequence of local searches. If costs are associated with the local searches, stopping rules may be derived using elements of statistical decision theory.

Two assumptions supporting the repulsion algorithm, and some stopping rules, are that the problem stated in (1) admits a finite number of local minima x_i^*, $i = 1, \ldots, N$, in K and that the regions of attraction form a partition of K. A region of attraction is a subset of the feasible domain associated with a local minimum, such that if a local optimization procedure starts from any point in this subset, then the sequence of points generated converges to that local minimum. More precisely, if x^* is a local minimum, then the region of attraction $R(x^*)$ associated with x^* is the subset of K such that if a local minimization procedure (A) starts from any point in $R(x^*)$, then the sequence of points A generates converges to x^*. With these definitions, if the algorithm A starts from an initial design $x_0 \in R(x^*)$, then the points $x_{i+1} = A(x_i) \in R(x^*), i \geq 0$, i.e. the points generated by the algorithm belong to the same region of attraction $R(x^*)$. It is important to note that for a given problem, the definition of its regions of attraction depends on the optimization procedure.

2 Stopping rules for multistart

Stopping rules developed by Boender and Rinnooy-Kan (1987) and by Betro and Schoen (1987) are reviewed briefly. Whereas the former reference uses inferences about the sizes of the regions of attraction, the latter uses inferences about the values of the objective function at the local minima. With stochastic multistart methods, each local minimum that a local search finds is viewed as an observation from an unknown distribution. For this reason the term "sampling" sometimes refers to a local search, that is, to the process of obtaining a new observation, and sometimes refers to the process of generating starting points for local searches.

After n local searches, the approach of Betro and Schoen (1987) uses the available data $[x_i^*, t_i = f(x_i^*)]$, $i = 1, \ldots, n$, to decide whether to continue with new searches or to stop. They assume that $t_i, i, \ldots, n$, are independent observations from a random variable T whose distribution is unknown. Let F denote this unknown distribution function. The Bayesian approach requires specifying a prior distribution reflecting one's beliefs about F. As the observations t_i, $i = 1, \ldots, n$, become available, the prior distribution is updated to obtain a posterior distribution using Bayes' theorem. Following a Bayesian approach, Betro and Schoen (1987) derive optimal stopping rules for the multistart procedure. In addition to a prior distribution, an optimal stopping rule requires a loss function expressing the cost associated with stopping after n local searches. Betro and Schoen (1987) use the loss function,

$$L(t_1, \ldots, t_n; c) = t_{(n)} + nc\,, \tag{2}$$

where $t_{(n)} = \min_{i=1,\ldots,n} t_i$, and c is the cost of a local search, expressed in units of $f(x)$, the objective function.

The decision rule is derived from minimizing the expected posterior cost. Optimal sequential rules compare the current cost given by (2) with the expected total cost of making more observations. Deriving such stopping rules appears to be very difficult, and therefore practical suboptimal rules are sought. Betro and Schoen (1987) indicate that "one-stage look ahead" (1-sla) rules are adequate in practical situations. The 1-sla rule prescribles to stop the sequence of local searches as soon as

$$L^1(t_1, \ldots, t_n; c) = L(t_1, \ldots, t_n; c)\,, \tag{3a}$$

where

$$L^1(t_1, \ldots, t_n; c) =$$
$$\min\left\{L(t_1, \ldots, t_n; c), E^{T_{n+1}}\left[L(t_1, \ldots, t_n, T_{n+1}; c)\right]\right\}\,, \tag{3b}$$

and $E^{T_{n+1}}(\cdot)$ is the expected value with respect to the distribution of the observation T_{n+1} conditional on the first n observations. That is, this rule terminates the sequential sampling when the present cost is lower than the total expected cost of a new local search. Betro and Scheon (1987) use a class of priors on the distribution functions $F(t)$ originally developed for nonparametric Bayesian inference. They use simple homogeneous processes to generate prior probability measures for the distribution functions $F(t)$ that are easy to update. Within this framework $F(t)$ is a stochastic process whose distribution is characterized by a function $\gamma(t)$ and a positive parameter λ.

For a simple homogeneous process $F(t)$, the prior expectation

$$E\left[F(t)\right] = 1 - e^{-\gamma(t)/\lambda}\,, \tag{4}$$

reflects an initial guess for the unknown $F(t)$. If $F_0(t)$ denotes one's initial guess, then (4) implies that $\gamma(t) = -\lambda \log\left[1 - F_0(t)\right]$.

In addition, condition (3a) reduces to

$$\int_{-\infty}^{-t_{(n)}} \left[1 - \hat{F}_n(t)\right] \mathrm{d}t \leq c\,, \tag{5}$$

where $\hat{F}_n(t) = E\left[F(t)|t_1, \ldots, t_n; c\right]$ is the posterior expectation of $F(t)$ given the observations $t_1, \ldots, t_n$.

Betro and Schoen (1987) suggest taking $\lambda = 1$ and a sensible function $F_0(t)$ that makes the evaluation of (5) easy. For a minimization problem, their suggestion translates to

$$1 - F_0(t) = \begin{cases} \left[1 - \left(1\sqrt{2}\right) e^{(a-t)/b}\right]^2\,, & t \geq a \\ 0.5\, e^{-2(\sqrt{2}-1)(a-t)/b}\,, & t < a \end{cases}\,, \tag{6}$$

where a and $b > 0$ are, respectively, a location, in fact the mode as well as the median, and a scale parameter for the density associated with F_0. This density is not symmetric. The parameter b may be assessed using the fact that the probability of a value lower than $a - 4.07b$ and the probability of a value higher than $a + 4.72b$ are both 0.01. With these choices the 1-sla rule reduces to

for $t_{(n)} > a$

$$\frac{\lambda}{n+\lambda}\exp(-S)\Big\{-a + \left[0.5(1 - 1\sqrt{2})^2 - 2 + \sqrt{2} + 0.5\right]b +$$
$$t_{(n)} + (2 - \sqrt{2})b\ \exp\left[(a - t_{(n)})/b\right] +$$
$$0.5(1 - 1/\sqrt{2})^2 \exp\left[2(a - t_{(n)})/b\right]\Big\} \leq c\,, \tag{7a}$$

for $t_{(n)} \leq a$

$$\frac{\lambda}{n+\lambda}\exp(-S)\frac{1}{4(\sqrt{2}-1)}b\exp\left[-2(\sqrt{2}-1)(a - t_{(n)})/b\right] \geq c, \tag{7b}$$

where

$$S = \sum_{j=1}^{n} \left[\gamma(t_{(j-1)}) - \gamma(t_{(j)})/(m_{j-1}+\lambda)\right], \tag{8a}$$

$$n_j = \#\left\{\text{distinct observations} = t_{(j)}\right\}, \tag{8b}$$

$$m_j = n - \sum_{i\leq j} n_i = m_j + n_j. \tag{8c}$$

In contrast with the previous approach, Boender and Rinnooy-Kan (1987) develop stopping rules on the basis of the sizes of the regions of attraction. Some rules combine the cost of a search with the cost of premature stopping, while others do not consider these costs. Rules that do not use costs are, roughly speaking, of two types. One type compares, at each iteration, the number of different minima that have been detected with an estimate of the total number of minima. The second type uses an estimate of the relative sizes of the regions of attraction that the algorithm has not detected. The rule this paper uses prescribes to stop performing new local searches as soon as the posterior expected relative size of the detected regions,

$$\frac{(n-w-1)(n+w)}{n(n-1)}, \tag{9}$$

exceeds a user specified number ε, where w denote the number of distinct local minima detected by the algorithm after n local searches.

3 The repulsion algorithm

The repulsion algorithm is a multistart method that sequentially generates starting points for local searches. Let z denote any points that the algorithm has previously visited i.e. z is either a starting point for a local search or an iterate that a local search has generated. It is assumed that any local search converges to a local minimum. Therefore, z belongs to the region of attraction of some local minimum. It is reasonable to assume that points near z are in the same region of attraction as z and therefore, one would like to initiate the subsequent local search away from z to increase the probability of starting that search in a different region of attraction. In fact, one would like to initiate subsequent local searches away from any points the algorithm has already visited. This is precisely what the repulsion algorithm does. To generate a starting point, the repulsion algorithm first generates $y \in K$ according to a uniform distribution and then transforms y to a point u to be used as a starting point. The transformation attempts to reach regions of attraction which may not have been detected by earlier samples. To accomplish this, each previously visited point z repels y away. The total repulsion on y is the sum of the individual repulsions. The repulsion z exerts on y is in the direction of $y - z$ and its magnitude, $\Delta(r)$, depends only on the distance from z to y, $r = d(z, y)$. There are many choices for $\Delta(r)$, but only linear repulsions are considered,

$$\Delta(r) = \begin{cases} \alpha - \beta r, & \text{if } \ 0 \leq r < r_0 \\ 0, & \text{if } \ r \geq r_0 \end{cases}, \tag{10}$$

where $\beta = \alpha/r_0$ and $0 \leq \alpha < r_0$.

The point y is repelled to the point u given by

$$u = u(y,z) = y + \Delta(r)\frac{y-z}{r}. \tag{11}$$

The constraints on α and β are desirable for two reasons. First, they guarantee that the repulsion does not exceed r_0. Second, if y' and y'' are two points such that $r' = d(z, y') < r'' = d(z, y'')$ then $\|u(y', z)\| < \|u(y'', z)\|$. In other words, the constraints avoid the point y' jumping over u'' to a point u' further away from z than u''.

Assume now that the algorithm has visited the points $z_1, \ldots, z_v$. The total repulsion on y is the sum of the repulsions due to each of the z_i's, that is, the repulsion algorithm repels y to the point

$$u = y + \sum_{i=1}^{v} \Delta(r_i)\frac{y-z_i}{r_i}, \tag{12}$$

where $\Delta(r_i)$ is given by (10) with $r_i = d(y, z_i)$.

Observe that $d[u(z,y),z] = \Delta(r) + r(1-\alpha/r_0) + \alpha \geq \alpha$. Therefore, if the algorithm has visited only the point z, then with the repulsion given by (10) there is zero probability of sampling the next starting point within a ball of radius α about z. This does not imply that the algorithm will never start a search within such ball. In fact, as the number of points visited by the algorithm increases, the total repulsion away from z will decrease.

4 Numerical examples

The following examples assess the performance of the repulsion algorithm. These examples were chosen because they have known nonconvexities in the design space. The program DOT (Vanderplaats and Hansen 1989), a feasible directions local optimizer, was used for the local searches.

4.1 Problem 1. Levy -Gomez

This example, introduced by Levy and Gomez (1985), can be stated as

$$\begin{aligned} &\text{min} \quad f(x_1,x_2) = 0.1(x_1^2 + x_2^2), \\ &\text{s.t.} \quad g(x_1,x_2) = 2\sin(2\pi x_2) - \sin(4\pi x_1) \leq 0, \\ &\qquad\ -1 \leq x_1, x_2 \leq 1. \end{aligned} \tag{13}$$

This problem has at least 24 local minima and the feasible domain is not connected. There is one global minimum, $x_1^* = x_2^* = 0$ with $f^* = 0$. Levy and Gomez (1989) used the tunneling algorithm and report results for 20 runs. For these 20 runs, the average number of objective function evulations is 625 and the average number of constraint function evualtions is 835. Ratschek and Rukne (1988) solved (13) with an interval arithmetic-based algorithm. Their solution required 415 interval evaluations of the objective function and 271 interval evaluations of the constraint function.

To compare the performance of the repulsion algorithm with the Levy-Gomez (1969) tunneling algorithm and the Ratschek-Rukne (1988) interval arithmetic algorithm, problem (13) was solved one hundred times. Table 1 summarizes the results. Table 1 indicates that for a repulsion with $\alpha = 0.2$ the repulsion algorithm performed better than any of the other algorithms. In fact, it required only about 20% of the function evaluations of multistart with no repulsion, with similar savings when compared with the interval arithmetic-based algorithm. Compared with the tunneling algorithm,

Table 1. Average results for 100 runs: problem 1

	$r_0 = 0.25$ no repulsion	$\alpha = 0.1$	$\alpha = 0.2$
No. successful runs	92	96	96
No. local searches to reach global minimum	1.038	1.020	1.021
No. objective evaluations	345	442	75
No. constraint evaluations	356	448	75
No. gradient evaluations	332	423	73

the repulsion algorithm with $\alpha = 0.2$ required only about 10% of function evaluations.

Table 1 also shows that the choices of α may strongly affect the performance of the repulsion algorithm. If the region of attraction of a local minimum is large, then small values of α may not suffice to repel starting points to regions of attraction that the algorithm has not detected. This might have been the reason for the decreased performance of the repulsion algorithm with $\alpha = 0.1$. This example suggests that if local searches often converge to the same local minimum, then, restarting the method with a higher α may speed the detection of new regions of attraction.

Global minimization methods may also be evaluated on the basis of the probability that they will find the global minimum. With this second criterion, Table 1 shows that, regardless of the value of α, the repulsion algorithm performed better than the multistart with no repulsion, since it increased the percentage of cases in which the global minimum was found from 92% to 96%.

4.2 Problem 2. Grillage structure

The second problem involves the two-beam grillage structure, shown in Fig. 1, subject to distributed static loads. Hajela (1990), Kavlic and Moe (1971), and Sepulveda and Schmit (1993) have discussed this example. The design variables are the cross-sectional areas of the beams (x_1, x_2). These are related to the moments of interia (I_1, I_2) and the section modulus (Z_1, Z_2) by empirical relations given by (Kavlie and Moe 1971):

$$Z_i = (x_i/1.48)^{1.82}\,, \tag{14a}$$

$$I_i = 1.007(x_i/1.48)^{2.68}\,, \quad i = 1, 2\,, \tag{14b}$$

The structure is designed for minimum weight subject to constraints on stresses ($\sigma^* = 20$ ksi) at the centre point and at the location of the maximum bending moment along the span. Figure 2 shows the design space for this problem, where one can clearly see that the design space is nonconvex.

First, the regions of attraction for each local minimum were set to approximately the same size by considering $1 \leq x_1 \leq 30$ and $1 \leq x_2 \leq 50$. Figure 3 plots the average number (over 100 runs) of local searches to reach the global optimum for the first time versus α for different values of r_0, i.e. Fig. 3 shows the effect of the repulsion. From this figure, one observes that the repulsion algorithm effectively reduces this number, and more so when r_0 is larger. Next, this example was solved with $1 \leq x_1 \leq 26$ and $1 \leq x_2 \leq 50$, which reduces the area of the region of attraction of the global minimum to approximately 5% of the area of the regions of attraction of the other two local minima. Figure 4 presents the results for both cases. Observe that the repulsion improves performance, especially for the second case where the area of the region of attraction associated with the global optimum is smaller.

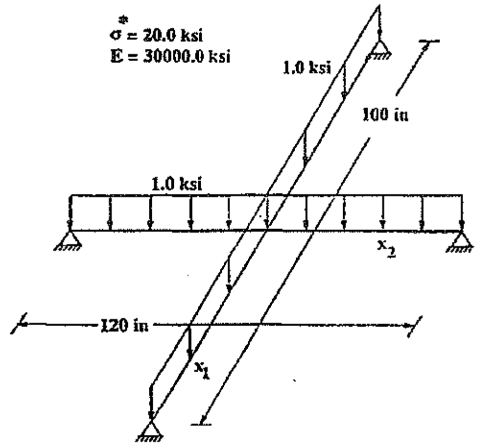

Fig. 1. Example problem 2: grillage structure

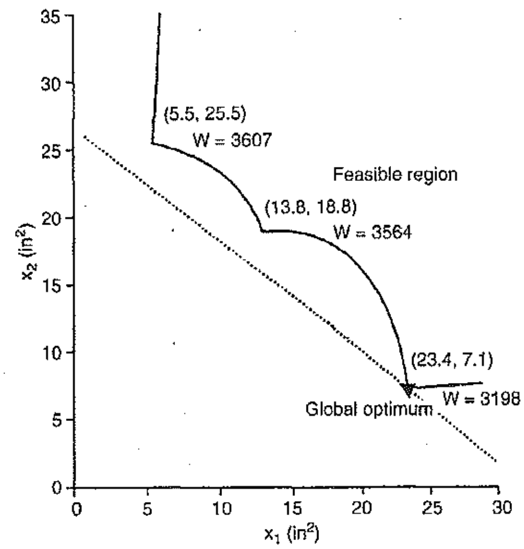

Fig. 2. Design space for problem 2: grillage structure

4.3 Problem 3. Disjoint design space

The third example involves the symmetric beam structure shown in Fig. 5. The beam is simply supported at both ends, has a length of 1000 cm and a squre cross-section (20 cm × 20 cm) with independent flange (t_b) and web (t_h) well thickness. The material properties are $\rho = 2.768 \times 10^{-3}$ and $E = 7.1 \times 10^6$ N/m^2. The beam is subjected to two independent load conditions at the midspan given by

$$f_1(t) = 4000 \text{ N} \sin \Omega_2 t, \quad \Omega_1 = 5 \text{ Hz}$$

and

$$f_2(t) = 5000 \text{ N} \sin \Omega_2 t, \quad \Omega_2 = 13 \text{ Hz}\,.$$

The design variables for this problem are the web and flange thickness (t_h and t_h) of the beam. The side constraints for these variables are 0.5 cm $\leq t_b, t_h \leq$ 10.0 cm. There are behaviour constraints on the midspan vertical displacement

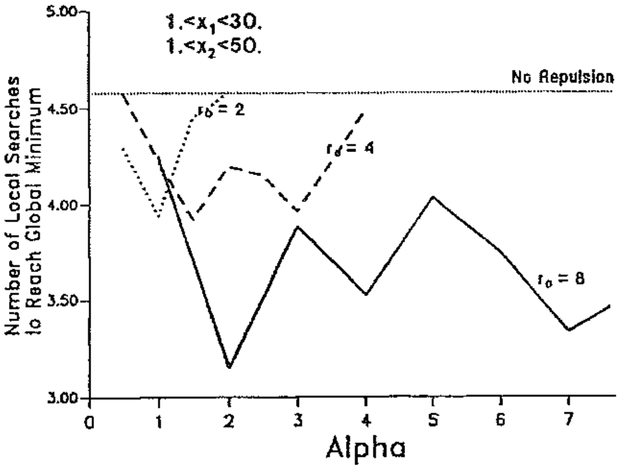

Fig. 3. Repulsion effect, case 1: grillage structure

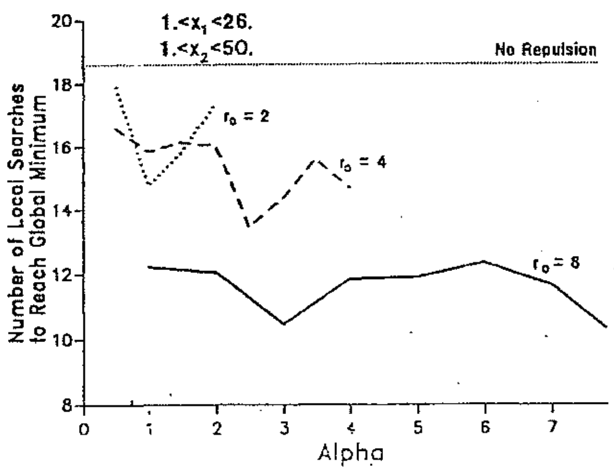

Fig. 4. Repulsion effect, case 2: grillage structure

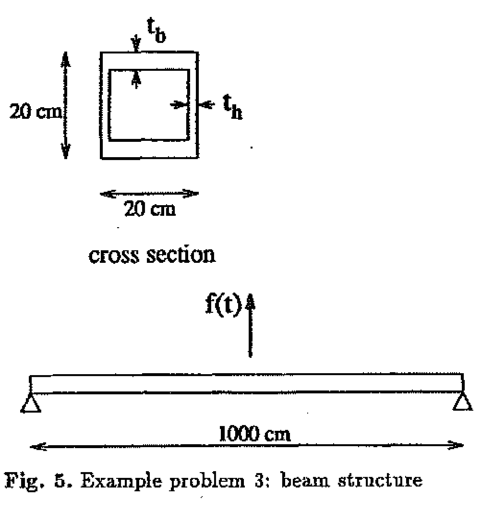

Fig. 5. Example problem 3: beam structure

($|w| \leq 1.0$ cm) and the maximum bending stress ($|\sigma| \leq 20$ ksi). The objective function is the total weight of the beam.

There are two independent load conditions for the problem, and there is no damping, therefore, due to resonance conditions, the design space is disjoint, as shown in Fig. 6. In this figure, shaded areas are infeasible design points. From this figure, it is also observed that each feasible component is not convex. In addition, Fig. 6 also shows the contours of the objective function for different weights, from where it is seen that the global optimum is obtained at $t_b = t_h = 0.5$ with a weight of 107 kg.

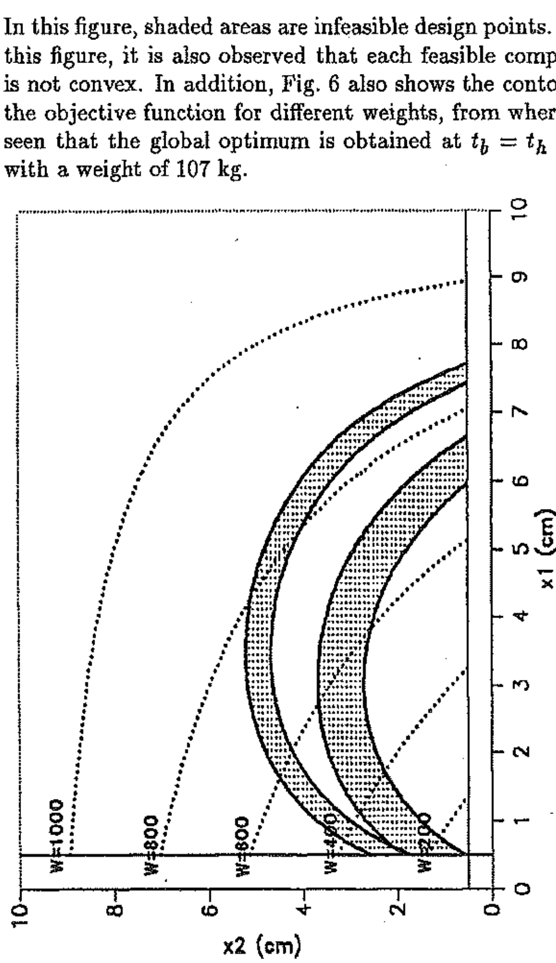

Fig. 6. Example problem 3: design space

The parameters for this problem were $a = 100$, $b = 100$, $c = 0.01$, and $\varepsilon = 99\%$. Two different strategies were used to solve the problem. The first strategy used only feasible initial designs to start local searches. The second strategy used both feasible and infeasible initial designs. In the second case, if the local optimizer did not produce a feasible design when starting from an infeasible point, then the complete sequence was deleted for repulsion purposes.

Figure 7 shows the number (average over 100 runs) of objective and constraint function evaluations as a function of α for both strategies. Since the problem was solved using finite differences, Fig. 7 includes the number of function evaluations necessary to estimate gradients. From Fig. 7 one observes that as α increases, i.e. as the repulsion increases, the number of function evaluations required for convergence decreases.

Figure 8 shows the average number (over 100 runs) of local searches for each of the two strategies. From the figure, one observes that the number of local searches is substantially larger when both infeasible and feasible points are used as starting points. This difference indicates that it is more efficient to start local searches from feasible points for the numerical implementation for local searches used in this paper. It is possible to assume that when other local optimizers are used (e.g. dual methods) the difference might not be as

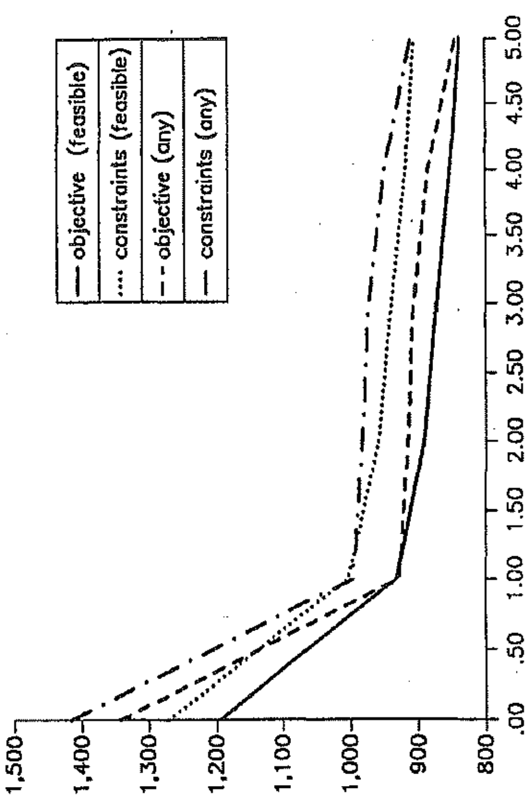

Fig. 7. Example problem 3: number of function evaluations

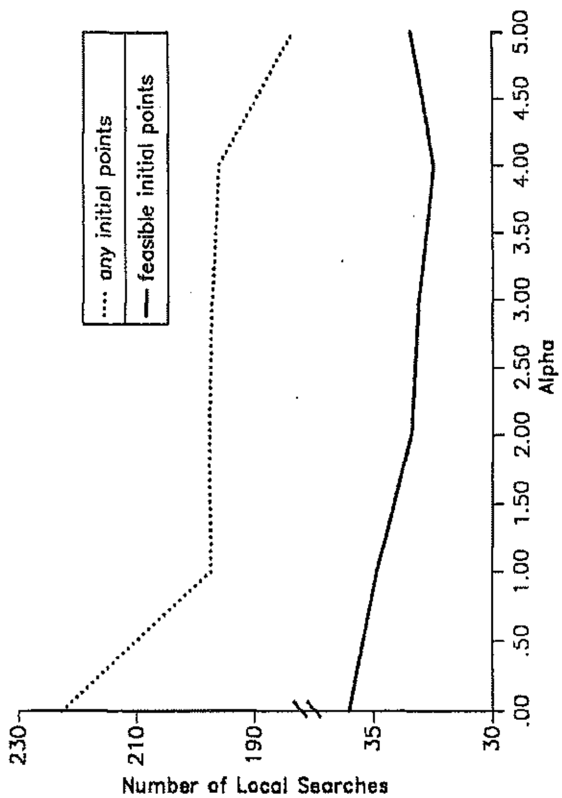

Fig. 8. Example problem 3: number of local searches

severe. Finally, Fig. 9 shows the number of local searches (average over 100 runs) to find the global optimum for the first time. As expected, when α increases, i.e. the repulsion increases, this number decreases since the probability of initiating searches in undetected regions of attraction increases.

4.4 Problem 4. Weight minimization of a speed reducer

This test problem is taken from the work of Floudas and Paradalos (1990) and Chew and Zheng (1988), and involves the design of a speed reducer for a small aircraft engine. In the design of speed reducer for small aircraft engines, a primary concern is the minimization of its weight, which it affects, for instance, the power-rating which is usually stated in terms of horsepower per engine weight as well as the costs of material and operations. The design problem can be formulated as a nonlinear minimization problem with constraints on design parameters including power transmission gas bending capacity, contact stress, the deflection and stress of shafts, and various constraints on the dimension of the weight reducer. The resulting optimization problem has the following form. Details on the meaning of the parameters and the derivation of the objective function can be found in the paper by Golinski (1970):

$$\min f(x) = 0.785 x_1 x_2^2 (3.333 x_3^2 + 14.933 x_3 - 43.093) - 1.508 x_1 (x_6^2 + x_7^2) + 7.477 (x_6^3 + x_7^3) + 0.785 (x_4 x_6^2 + x_5 x_7^2) \,. \quad (15a)$$

Subject to

$$x_1 x_2^2 x_3 \geq 27 \,, \quad x_1 x_2^2 x_3^2 \geq 397.5 \,,$$

$$x_2 x_6^4 x_3 x_4^{-3} \geq 1.93 \,, \quad x_2 x_7^4 x_3 x_5^{-3} \geq 1.93 \,,$$

$$\left[(745 x_4 x_2^{-1} x_3^{-1})^2 + 16.91 \times 10^6\right]^{0.5} x_6^{-3} \leq 110 \,,$$

$$\left[(745 x_5 x_2^{-1} x_3^{-1})^2 + 157.5 \times 10^6\right]^{0.5} x_7^{-3} \leq 85 \,, \quad (15b)$$

$$x_2 x_3 \leq 40 \,, \quad x_1 x_2^{-1} \geq 5 \,, \quad x_1 x_2^{-1} \leq 12 \,,$$

$$1.5 x_6 - x_4 \leq -1.9 \,, \quad 1.5 x_7 - x_5 \leq -1.9 \,,$$

and side constraints given by $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4$, $x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$ and $5 \leq x_7 \leq 5.5$.

The problem was solved with $r_0 = 3$ and different values of α. In all cases the global solution found by the repulsion algorithm is essentially the same, with only minor variations in objective function value. Table 2 shows the best solution found by the repulsion algorithm and the best solution given in the work of Floudas and Paradalos (1990). The design variable values are essentially identical with a difference of 0.4% in the final objective.

Table 2. Optimal solution for problem 4

	Repulsion Algorithm	Ref. 15
x_1	3.5	3.5
x_2	0.7	0.7
x_3	17	17
x_4	7.3	7.3
x_5	7.3	7.71
x_6	3.35	3.35
x_7	5.286	5.287
f(x)	2983.38	2994.47

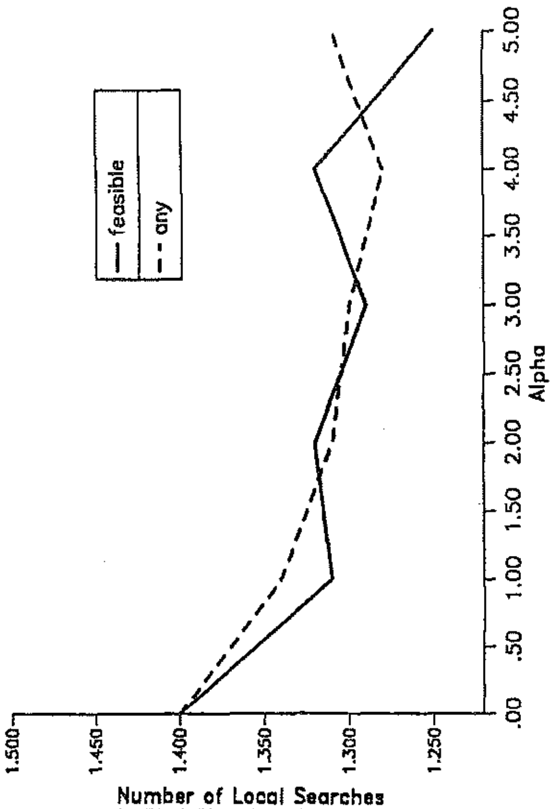

Fig. 9. Example problem 3: number of local searches to find global optimum

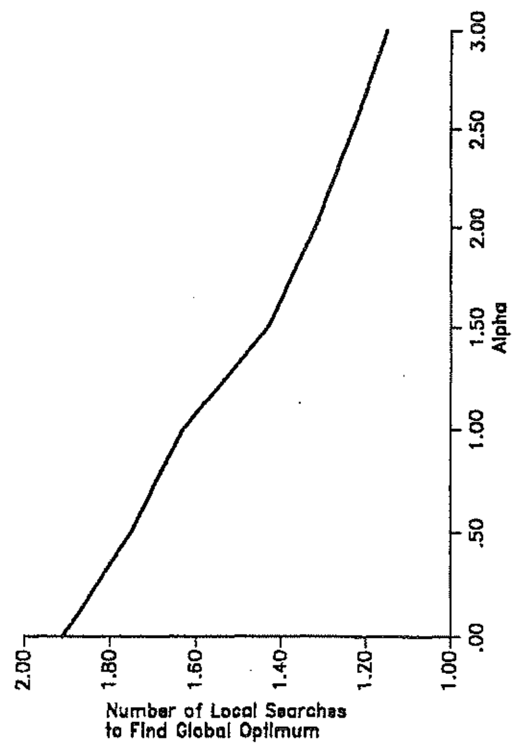

Fig. 10. Example problem 4: number of local searches to find global optimum

Figure 10 shows the number of local searches (average over 100 runs) to find the global optimum for the first time. The results show, as in the previous examples, that when α increases (repulsion increases) this number desreases indicating that with increasing repulsion, the probability of initiating searches in undetected regions of attraction increases.

5 Conclusions

This paper sets forth a multistart method to find the global optimum for nonconvex problems. The repulsion algorithm generates initial points for local searches to reduce the probability of initiating local searches in previously visited regions of attraction. Numerical results indicate that compared with classical multistart algorithms, the repulsion algorithm reduces both the number of local searches to find the global optimum for the first time and the number of function evaluations.

The introduction of the repulsion algorithm opens a number of interesting methodological problems. First, the stopping rule of Betro and Schoen (1987) assumes that the minima local searches find are stochastically independent and identically distributed. This assumption is reasonable when the starting points are generated independently from the same distribution. With the repulsion algorithm, however, the repulsion changes the distribution of the starting point from one local search to the next. This distribution depends on the points that previous local searches have generated, and therefore, the starting points are not independent. With this in mind, it might be possible to find a more suitable stopping rule to use with the repulsion algorithm than the rule of Betro and Schoen (1987).

It is possible to devise strategies other than the repulsion algorithm to increase the probability of starting a local search in an undetected region of attraction. One such strategy consists of generating candidate starting points independently and according to some distribution, e.g. uniform. After a candidate has been generated, with probability P one uses that point as a starting point, and with probability $1 - P$ one does not use it. The probability P might depend on the distance to points that previous local searches have generated. One would expect this algorithm to be somewhat less efficient than the repulsion algorithm in that a fraction of the candidate starting points are rejected.

Finally, from a standpoint of computational implementation, storing all visited points can take substantial storage if the number of design variables is large. In this case, visited points that are very close to each other can be replaced by their centroid, or alternatively by a single point which approximately generates the same repulsion. In this respect, the ideas of clustering (see e.g. Rinnooy-Kan and Timer 1986) commonly used for classical multistart algorithms can reduce the number of visited points for repulsion purposes.

References

Betro, B.; Schoen, F. 1987: Sequential stopping rules for the multistart algorithm in global optimization. *Math. Prog.* **38**, 271-286

Boender, C.G.E.; Rinnooy-Kan, A. 1987: Bayesian stopping rules for multistart global optimization methods. *Math. Prog.* **37**, 59-80

Cassis, J.H.; Schmit, L.A. 1976: Optimum structural design with dynamic constraints. *J. Struct. Div., ASCE* **102**, 2053-2071

Chew, S.H.; Zheng, Q. 1988: Integral global optimization. *Lecture Notes in Economics and Mathematical Systems* **298**. Berlin, Heidelberg, New York: Springer

Floudas, C.A.; Pardalos, P.M. 1990: A collection of test problems for constrained global optimization algorithms. *Lecture Notes in Computer Science* **455**. Berlin, Heidelberg, New York: Springer

Golinski, J. 1970: Optimal synthesis problems solved by means of nonlinear programming and random methods. *J. Mech.* **5**, 287-309

Hajela, P. 1990: Genetic search - an approach to the nonconvex optimization problem. *AIAA J.* **28**, 1205-1210

Johnson, E.H. 1976: Disjoint design spaces in the optimization of harmonically excited structures. *AIAA J.* **14**, 259-261

Kavlie, D.; Moe, J. 1971: Automated design of frame structures. *J. Struct. Div., ASCE* **97**, 33-62

Levy, A.V.; Gomez, S. 1985: The tunneling method applied to global optimization. In: Boggs, P.T.; Byro, R.H. and Schnabel, R.B. (eds.) *Numerical optimization*, SIAM Conference, Philadelphia, USA

Mills-Curran, W.C.; Schmit, L.A. 1985: Structural optimization with dynamic behavior constraints. *AIAA J.* **23**, 136-138

Moses, F.; Onoda, S. 1969: Minimum weight design of structures with application to elastic grillages. *Int. J. Num. Meth. Eng.* **1**, 311-331

Ratschek, H.; Rukne, J. 1988: *New computer methods for global optimization*. New York: John Wiley

Rinnooy-Kan, A.; Timmer, G.T. 1986: Stochastic global optimization methods. Part I: clustering methods; Part II: multilevel methods. *Math. Prog.* **39**, 27-78

Sepulveda, A.E.; Jin, I.M. 1992: Design of structure/control systems with transient response constraints exhibiting relative minima. *Proc. 4th AIAA/USAF/NASA/OAI Symp. on Multidisciplinary Analysis and Optimization, AIAA-92-4736-CP* (held in Cleveland OH), pp. 371-378

Sepulveda, A.E.; Schmit, L.A. 1993: Approximation-based global optimization strategy for structural synthesis. *AIAA J.* **31**, 180-188

Vanderplaats, G.N.; Hansen, S.R. 1989: *DOT user's manual, version 2.04*, Goleta, CA: VMA Engineering.

Received July 31, 1995
Revised manuscript received Nov. 13, 1995