

## Dear Author

Here are the proofs of your article.

- You can submit your corrections **online** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- Please return your proof together with the permission to publish confirmation.
- For **fax** submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the journal title, article number, and your name when sending your response via e-mail, fax or regular mail.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style. Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.

### Please note

Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI.

**Further changes are, therefore, not possible.**

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL:

<http://dx.doi.org/10.1007/s00158-009-0433-x>

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information, go to:

<http://www.springerlink.com>.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us, if you would like to have these documents returned.

The **printed version** will follow in a forthcoming issue.

**To: Jaymeer B Bunag**  
**E-mail: j.bunag@spi-bpo.com**  
**Fax: +1-202-3155796**

**Re: SPi**  
**SPi Building, Pascor Drive**  
**Sto Nino, Paranaque City 1700**  
**Philippines**

**Structural and Multidisciplinary Optimization DOI 10.1007/s00158-009-0433-x**  
Improvements to single-objective constrained predator-prey evolutionary optimization algorithm  
**Chowdhury · Dulikravich**

## Permission to publish

I have checked the proofs of my article and

- I have **no corrections**. The article is ready to be published without changes.
- I have **a few corrections**. I am enclosing the following pages:
- I have made **many corrections**. Enclosed is the **complete article**.

**Date / signature:** \_\_\_\_\_

**Metadata of the article that will be visualized in Online**


---

**Please note: Image will appear in color online but will be printed in black and white.**

---

1	Article Title	<b>Improvements to single-objective constrained predator–prey evolutionary optimization algorithm</b>	
2	Article Sub- Title		
3	Article Copyright - Year	<b>Springer-Verlag 2009 (This will be the copyright line in the final PDF)</b>	
4	Journal Name	Structural and Multidisciplinary Optimization	
5		Family Name	<b>Dulikravich</b>
6		Particle	
7		Given Name	<b>George S.</b>
8		Suffix	
9	Corresponding Author	Organization	Florida International University
10		Division	MAIDROC Laboratory, Department of Mechanical and Materials Eng.
11		Address	10555 West Flagler St., EC 3474, Miami 33174, FL, USA
12		e-mail	dulikrav@fiu.edu
13			Family Name
14		Particle	
15		Given Name	<b>Souma</b>
16		Suffix	
17	Author	Organization	Florida International University
18		Division	MAIDROC Laboratory, Department of Mechanical and Materials Eng.
19		Address	10555 West Flagler St., EC 3474, Miami 33174, FL, USA
20		e-mail	
21		Received	26 September 2008
22	Schedule	Revised	17 August 2009
23		Accepted	22 August 2009
24	Abstract	In predator–prey algorithm, a relatively small number of predators (“lions”) and a much larger number of prey (“antelopes”) are randomly placed on a two dimensional lattice with connected ends representing an unfolded surface of a torus. The predators are partially or completely biased towards one or more objectives, based on which each predator kills the weakest prey in its neighborhood. A stronger prey created through evolution replaces this prey. In case of constrained problems, the sum of constraint violations serves as an additional objective. Modifications of the basic predator–prey algorithm have been implemented in this paper regarding the selection procedure, apparent movement of the predators, and mutation strategy. Further modifications have been made making the algorithm capable of handling multiple equality and inequality constraints. The final modified algorithm was tested on standard linear/nonlinear and constrained/unconstrained single-objective optimization problems.	
25	Keywords separated by ' - '	Predator–prey algorithm - Constrained optimization - Evolutionary algorithms - Crossover - Mutation	
26	Foot note information	This paper has not been published nor submitted for publication anywhere else besides Structural and Multidisciplinary Optimization.	

# Improvements to single-objective constrained predator–prey evolutionary optimization algorithm

Souma Chowdhury · George S. Dulikravich

Received: 26 September 2008 / Revised: 17 August 2009 / Accepted: 22 August 2009  
© Springer-Verlag 2009

1 **Abstract** In predator–prey algorithm, a relatively small  
2 number of predators (“lions”) and a much larger number  
3 of prey (“antelopes”) are randomly placed on a two dimen-  
4 sional lattice with connected ends representing an unfolded  
5 surface of a torus. The predators are partially or completely  
6 biased towards one or more objectives, based on which  
7 each predator kills the weakest prey in its neighborhood. A  
8 stronger prey created through evolution replaces this prey.  
9 In case of constrained problems, the sum of constraint  
10 violations serves as an additional objective. Modifications  
11 of the basic predator–prey algorithm have been imple-  
12 mented in this paper regarding the selection procedure,  
13 apparent movement of the predators, and mutation strategy.  
14 Further modifications have been made making the algo-  
15 rithm capable of handling multiple equality and inequal-  
16 ity constraints. The final modified algorithm was tested  
17 on standard linear/nonlinear and constrained/unconstrained  
18 single-objective optimization problems.

19 **Keywords** Predator–prey algorithm · Constrained  
20 optimization · Evolutionary algorithms · Crossover ·  
21 Mutation

## 1 Introduction 22

The last few decades have seen the development of opti- 23  
mization algorithms inspired by the principles of natural 24  
evolution. These algorithms, often termed Evolutionary 25  
Optimization Algorithms (EOAs), use a set of candidate 26  
solutions (population space) and follow an iterative proce- 27  
dure to produce a final set of the best compromise 28  
solutions, the graphical representation of which is termed 29  
as the Pareto front (Deb 2002). In case of single objec- 30  
tive problems the Pareto front reduces to a single optimal 31  
solution known as the global minimum or global maxi- 32  
mum. Genetic algorithm, differential evolution, particle 33  
swarm, and predator–prey algorithms are some of the most 34  
prominent EOAs. 35

Hybrid optimization techniques with automatic switch- 36  
ing capability among a number of EOAs and classi- 37  
cal gradient-based optimization algorithms have also been 38  
developed (Dulikravich et al. 1999; Colaço et al. 2005; 39  
Moral and Dulikravich 2008) and successfully implemented 40  
in multi-disciplinary problems (Martin and Dulikravich 41  
2002). 42

In 1998, Hans Paul Schwefel proposed a new EOA to 43  
search for Pareto-optimal solutions (Laumanns et al. 1998) 44  
from a randomly generated initial population of candidate 45  
solutions. This algorithm imitates the natural phenomena 46  
that a predator kills the weakest prey in its neighborhood, 47  
and the next generations of prey that evolve are relatively 48  
stronger and more immune to such predator attacks. 49

In nature, individual predators have different means of 50  
tracking their prey, as a result of which their choice of 51  
prey might differ. This algorithm mimics such preferential 52  
hunting tactics in associating each predator or a group of 53  
predators with different objectives. In course of their ran- 54  
dom movements in the prey neighborhood, each predator 55

This paper has not been published nor submitted for publication  
anywhere else besides Structural and Multidisciplinary Optimization.

S. Chowdhury · G. S. Dulikravich (✉)  
MAIDROC Laboratory, Department of Mechanical  
and Materials Eng., Florida International University,  
10555 West Flagler St., EC 3474, Miami, FL 33174, USA  
e-mail: dulikrav@fiu.edu  
URL: <http://www.eng.fiu.edu/mme/>, <http://maidroc.fiu.edu>,  
<http://www.ISIPSE.net>

56 tracks down the weakest local prey, that is, the one which  
57 is the most vulnerable to their particular hunting tactics.  
58 This refers to the prey which has the worst objective value  
59 with respect to that predator. The prey, thus killed, is then  
60 replaced by a stronger successor or a child.

61 Such phenomenon repeating itself over generations leads  
62 to evolution of the prey population into stronger species that  
63 are more immune to the distinct hunting tactics of the differ-  
64 ent predators. This, in mathematical terms, reflects improve-  
65 ments of the prey population as a whole with respect to  
66 all function objectives. In case of a multi-objective opti-  
67 mization problem, natural selection based on such a method  
68 ensures convergence of solutions towards the Pareto front  
69 without any direct implementation of a dominance based  
70 criterion. This is the major contribution of the predator-prey  
71 class of evolutionary algorithms.

72 In this light, application of the predator-prey selection  
73 technique to unconstrained single-objective optimization  
74 problems does not furnish any unique benefit. Conse-  
75 quently, in case of single-objective optimization problems,  
76 this algorithm acts like a typical genetic algorithm.

77 However, there is one major difference between the  
78 predator-prey algorithm and any other evolutionary algo-  
79 rithm. Absence of any global mixing of the population  
80 members results in localized improvements of the prey  
81 population in the predator-prey algorithm. This is further  
82 facilitated by the evolutionary techniques (crossover and  
83 mutation) employed in predator-prey algorithm that estab-  
84 lish a localized and adaptive search, thereby enhancing the  
85 robustness of this optimization algorithm. This proves to be  
86 slower, but a more reliable mechanism of progress towards  
87 the global optimum in case of complex single-objective  
88 functions, like ones with multiple local optima.

89 In the predator-prey algorithm, prey, which represent  
90 members of the population/solution space are randomly  
91 placed (unique integer co-ordinates are randomly generated  
92 for each prey) on a two dimensional lattice with connected  
93 ends, that is, an unfolded surface of a torus. Predators,  
94 which are comparatively fewer in number than prey, are  
95 placed at the cell centers of the same 2D lattice. Each  
96 predator is completely biased towards one of the objec-  
97 tives, which form the quantitative basis of determining  
98 the weakest local prey. After the weakest local prey (the  
99 local solution candidate with the lowest value of the fit-  
100 ness function) is identified, it is eliminated (this "prey" is  
101 "killed") and a new prey is created through mutation of  
102 one of the immediate surviving neighboring prey. While  
103 the prey remain stationary, the predators move to a random  
104 neighboring location after every generation.

105 However, this original predator-prey optimization algo-  
106 rithm appears to have difficulty in producing well dis-  
107 tributed non-dominated solutions along the Pareto front.  
108 Since then, several modifications of the above algorithm

109 have appeared in literature. Deb (2002) suggested an  
110 improved version of the algorithm which involved the asso-  
111 ciation of each predator with a weighted sum of objectives  
112 instead of one particular objective. Certain new features,  
113 namely, the 'elite preservation operator', the 'recombina-  
114 tion operator' and the 'diversity preservation operator' were  
115 also included. A further modified version of the algorithm  
116 was proposed by Li (2003), where a dynamic spatial struc-  
117 ture of the predator-prey population was used. It involved  
118 the movement of both predators and prey and changing  
119 population size of prey. Some other versions of the algo-  
120 rithm have been presented by Grimme and Schmitt (2006)  
121 and Silva et al. (2002). The former uses a modified recom-  
122 bination and mutation model. The latter, predominantly a  
123 particle swarm optimization algorithm, introduces the con-  
124 cept of predator-prey interactions in the swarm to control  
125 the balance between exploration and exploitation, hence  
126 improving both diversity and rate of convergence.

127 Most of the above versions are strictly directed towards  
128 unconstrained multiobjective optimization problems. The  
129 majority of practical applications of optimization involve  
130 constraints. This demands optimization algorithms capa-  
131 ble of producing solutions that are both optimum as well  
132 as feasible with respect to the problem constraints. There  
133 exist very few instances of published applications of any  
134 form of the predator-prey algorithm to such real world  
135 problems. Nevertheless, since the basic concept of the  
136 predator-prey algorithm is significantly different from other  
137 standard EOAs, there is sufficient basis to believe that the  
138 potentials of this algorithm have not been fully realized.

139 The fundamental idea of the work presented here is  
140 to combine the basic predator-prey algorithm with some  
141 advanced features such as the constraint dominance cri-  
142 terion, hypercube sizing and the epidemic operator, to  
143 develop a reliable method of solving complex constrained/  
144 unconstrained single-objective optimization problems.

145 There are two benefits of using a multi-objective  
146 approach. The algorithm can be used without changing  
147 the basic dynamics of the predator-prey interaction and  
148 weighted objective association of predators.

149 In case of solving constrained single-objective problems,  
150 the total constraint violation acts as the third objective. The  
151 constraint dominance criterion gives preference to selection  
152 based on lower constraint violation. On the other hand, the  
153 property that the first two objectives are equal to the actual  
154 problem objective function leads to two-thirds biasing of  
155 predators towards this objective. Both these factors acting  
156 together provide a balance between selections of prey (solu-  
157 tions) based on actual objective value as well as its distance  
158 from the feasible domain (constraint violation).

159 This method is somewhat similar to the filter method  
160 of constrained optimization, with the dominance crite-  
161 rion biased towards selection based on total constraint

162 violation. The weighted function association of predators on  
 163 the other hand creates a counter effect as explained above.  
 164 However, the version of the algorithm that was found to  
 165 perform most satisfactorily when dealing with constrained  
 166 problems involved selection based on ‘constraint domi-  
 167 nance’ criterion instead of the ‘weighted sum of objectives’  
 168 method which makes SOMPP very similar to NSGA II by  
 169 Deb et al. (2000) with respect to the selection procedure.

170 This study presents the development of a constrained  
 171 single-objective version of the modified predator–prey  
 172 algorithm which involves new features that are expected  
 173 to promote dependability in terms of convergence of  
 174 solutions as well as reduction of the number of func-  
 175 tion evaluations necessary. This single-objective, modi-  
 176 fied, predator–prey algorithm (SOMPP) has been derived  
 177 from the basic predator–prey algorithm. Any unconstrained  
 178 single-objective optimization problem was treated as a two-  
 179 objective optimization problem, where the second objective  
 180 is just a clone of the first one. In case of the constrained  
 181 problems, all the equality and inequality constraints were  
 182 collaged together to form a third objective and the prob-  
 183 lem was solved as a three-objective optimization problem  
 184 (Chowdhury et al. 2009, 2010) where the first two are  
 185 equivalent and different from the third objective (constraint  
 186 objective).

## 187 2 Single objective modified predator–prey algorithm

188 Any general constrained single objective test problem is  
 189 reformulated as follows.

$$190 \begin{aligned} \text{Minimize } f_1 &= f(X) \\ \text{Minimize } f_2 &= f_1 \end{aligned} \quad (1)$$

191 subject to

$$192 \begin{aligned} g_{ic} &\leq 0, \quad ic = 1, 2, 3, \dots, p \\ h_{ic} &= 0, \quad ic = p + 1, p + 2, \dots, p + q \\ p, q &\in Nc \end{aligned} \quad (2)$$

193 Here,  $X$  is the vector of design variables, that is,

$$194 X = (x_1, x_2, \dots, x_v, \dots, x_{N_v}) \quad , \quad x_v \in R$$

195 The inequality and equality constraints are added up to form  
 196 the third objective

$$197 \begin{aligned} \text{Minimize } f_3 &= \sum_{ic=1}^p \max(g_{ic}, 0) \\ &+ \sum_{ic=p+1}^{p+q} \max((h_{ic} - \varepsilon), 0) \end{aligned} \quad (3)$$

199 where  $\varepsilon$  is the tolerance for equality objectives.

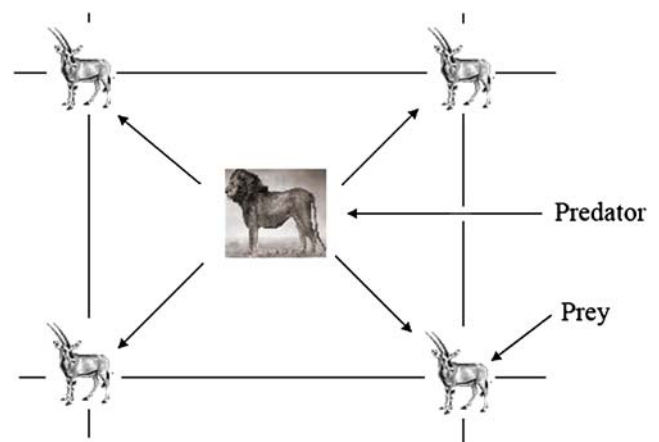
200 In case of unconstrained problems, SOMPP indeed acts  
 201 as a generic genetic algorithm which selects solutions based  
 202 on their objective value. However, SOMPP also applies the  
 203 hypercube technique as a qualification criterion for accept-  
 204 ing new/child solutions. This incorporates diversity into the  
 205 population in the same way as the concept of crowding  
 206 distance does in case of NSGA II by Deb et al. (2000)  
 207 and the recently developed Constrained Particle Swarm  
 208 Optimization by Venter and Haftka (2009).

209 The initialization and subsequent steps executed by the  
 210 algorithm in each generation in solving a single-objective  
 211 optimization problem are sequentially presented below. It  
 212 should be noted that in case of a maximization problem the  
 213 function is multiplied by ‘ $-1$ ’, to convert it into a general  
 214 minimization problem.

215 First, a population of  $N$  candidate solutions (prey) is  
 216 created using Sobol’s (1976) quasi-random sequence gen-  
 217 erator to generate their vectors of design variables. Using  
 218 these values of design variables, objective functions for each  
 219 candidate solution are evaluated. Sobol’s algorithm offers  
 220 significantly more uniform distribution of random numbers  
 221 than a typical random number generator routine.

222 Then, the prey are placed at the nodes of a two dimen-  
 223 sional grid with connected ends hence having a toroidal  
 224 nature. The grid is allowed to adjust its size dynamically  
 225 according to the population size maintaining the dimensions  
 226  $I \times J$ , where we found after numerical experimentation that  
 227 the most suitable value for  $J$  is  $J = 5$ . Consequently,  $I$   
 228 is chosen such that  $I$  is the lowest possible integer for which  
 229  $N < I \times J$ . Random members of the prey population are  
 230 cloned (four or less) if required in order to ensure that all  
 231 grid points (having integer co-ordinates) are occupied by  
 232 prey.

233 Similarly,  $M$  predators are placed on the same 2D  
 234 grid such that they occupy random cell centers (Fig. 1).



**Fig. 1** An active four prey locality/neighborhood on the grid drawn on an unfolded toroidal surface

235 The value of  $M$  is determined by the following empirical  
236 formula.

$$237 \quad M = \max \left( \left( \left\lceil \frac{N}{20} \right\rceil \times N_f \right), 4 \right) \quad (4)$$

238 where,  $\lceil r \rceil$  is the lowest integer greater than  $r$ ,  $r \in R^+$ , and  
239  $N_f$  is the number of objectives. Each predator is associated  
240 with a weighted value of the objectives as follows.

$$241 \quad f^j = \sum_{i=1}^2 w_i^j f_i \quad (5)$$

$$242 \quad \sum_{i=1}^2 w_i = 1$$

242 Here,  $f^j$  is the effective objective function value that the  
243  $j$ th predator is associated with and  $w_i^j$  is respective weight of  
244 the  $i$ th objective with respect to the  $j$ th predator. The weights  
245 are distributed uniformly in case of two-objective problems  
246 (from (0,1) to (1,0)) and using Sobol's (1976) algorithm in  
247 case of problems with more than two objectives (constrained  
248 problems).

249 Thus, the basic SOMPP algorithm was designed to han-  
250 dle problems with more than one objective. The dynamics of  
251 the algorithm is also conducive to multi-objective optimiza-  
252 tion. Hence, an unconstrained single objective optimization  
253 problem is treated as a two-objective problem with equal  
254 objective values, that is,

$$255 \quad f_1(X) = f_2(X)$$

256 This allows one to use the same predator-prey dynamics  
257 as in a multi-objective problem. However, mathematically  
258 the algorithm will be solving the single objective problem,  
259 because each predator will be completely biased towards a  
260 single objective (5) since  $f^j = f_1 = f_2$  for each predator  
261 in the grid.

262 In case of a constrained single-objective problem, the  
263 total constraint violation is treated as the third objective.

264 This is why the number of objectives is defined by the  
265 general notation  $N_f$ , where  $N_f = 2$  for an unconstrained  
266 single-objective problem and  $N_f = 3$  for a constrained  
267 single-objective problem. However, the single-objective  
268 MPP (SOMPP) algorithm is significantly different from  
269 the multi-objective MPP in some other features, such as  
270 the mutation operator, the hypercube operator, rank based  
271 predator relocation, nine prey neighborhoods and the epi-  
272 demic operator. By the virtue of the above features, SOMPP  
273 acts as a distinct version of the Modified Predator Prey algo-  
274 rithm suitable for handling single-objective optimization  
275 problems.

276 The presence of three objectives, among which the first  
277 two are equivalent and different from the third objective

(constraint objective), creates a platform where one can reap  
278 substantial benefits from the preferential hunting nature of  
279 predators based on their colligation with different objec-  
280 tives. The property that the first two objectives are equal to  
281 the actual problem objective function entails a higher prob-  
282 ability of prey killings (almost two thirds) based on their  
283 weakness with respect to the actual objective. On the other  
284 hand, the constraint dominance measure that acts as one  
285 of the qualification criterion when accepting new (replace-  
286 ment for the killed) prey, promotes selection based on lower  
287 constraint violation.  
288

289 Notice also that the constraint handling technique used  
290 in SOMPP is free of any user-defined constants/coefficients,  
291 unlike most penalty function methods (shown in the review  
292 by Coello Coello 2002), where the user has to tune different  
293 problem specific algorithm coefficients. Consequently, the  
294 scope of applying SOMPP universally to any single objec-  
295 tive problem with equality and/or inequality constraints  
296 seems promising.

297 Since predators are randomly located at the centers of  
298 quadrilateral cells drawn on an unfolded toroidal surface,  
299 each neighborhood that contains a predator can be termed  
300 as an 'active locality' as shown in Fig. 1. In each of these  
301 localities/cells, the value of  $f$  as defined by (5) correspond-  
302 ing to the local predator, is calculated for each prey (local  
303 fitness of prey). The weakest prey, that is, the prey hav-  
304 ing the maximum value of  $f$  is selected to be killed and  
305 replaced by a new prey produced by the crossover of the  
306 two strongest neighboring prey and a subsequent mutation  
307 of the crossover child.

308 The blend crossover (BLX- $\alpha$ ) (Deb 2002) was used in  
309 this case.

$$310 \quad y_v^{(1,t+1)} = (1 - \gamma_v) x_v^{(1,t)} + \gamma_v x_v^{(2,t)} \quad (6)$$

$$311 \quad \gamma_v = (1 + 2\alpha) u_v - \alpha$$

312 Here,  $x_v^{(1,t)}$  and  $x_v^{(2,t)}$  are the design variables that define  
313 parent solutions,  $y_v^{(1,t+1)}$  is the design variable that defines  
314 the child solution and  $u_v$  is a random number between 0 and  
315 1. A value of 0.5 was used for  $\alpha$  as suggested by Deb (2002).

316 Non-uniform mutation (Deb 2002), as defined below,  
was used in this algorithm.

$$317 \quad \beta = 10^{-\left(1 + K^t / t_{\max}\right)} \quad (7)$$

$$318 \quad y_v^{(1,t+1)} = x_v^{(1,t+1)} + \tau \left( x_v^{(U)} - x_v^{(L)} \right) \left( 1 - r_v^{\left(1 - t / t_{\max}\right)^b} \right) \times \beta$$

319 Here,  $10^{-K}$  is the terminal order of magnitude of the extent  
320 of mutation,  $y_v^{(1,t+1)}$  is the child produced by mutation of  
321

322 the  $v$ th variable,  $x_v^{(U)}$  and  $x_v^{(L)}$  are upper and lower lim- 370  
 323 its of the  $v$ th variable,  $r_v$  is a random number between 371  
 324 0 and 1,  $\tau$  takes a Boolean value  $-1$  or  $1$ , each with a 372  
 325 probability of  $0.5$ ,  $t$  and  $t_{\max}$  are the number of function 373  
 326 evaluations performed until then and maximum allowed 374  
 327 number of function evaluations, respectively, while  $b$  is the 375  
 328 user defined parameter ( $b = 1.5$  determined empirically) 376  
 329 and  $\beta$  is the scaling parameter. 377

330 The evolutionary operators such as crossover and muta- 378  
 331 tion operator applied in SOMPP are based on the niching 379  
 332 strategies used in genetic algorithms. The BLX- $\alpha$  crossover 380  
 333 is utilized since it facilitates genetic recombination that 381  
 334 is adaptive to the existing diversity in the parent popula- 382  
 335 tion; a desirable characteristic for Pareto front convergence. 383  
 336 Also, non uniform mutation (Michalewicz 1992) is utilized 384  
 337 since it provides a uniformly distributed search in the ear- 385  
 338 lier generations and a relatively focused search in the later 386  
 339 ones. 387

340 The child prey produced by crossover and mutation qual- 388  
 341 ifies to be accepted only if it fulfills the following three 389  
 342 criteria: 390

- 343 1. The child is stronger than the worst local prey based on 391  
 344  $f$  calculated by (2), 392
- 345 2. The child is non-dominated (Deb 2002) with respect to 393  
 346 the other three local prey, and 394
- 347 3. The child is not within the objective space hypercube 395  
 348 (Deb 2002) of the remaining three neighboring prey. 396

349 Apparently, the treatment of constrained single-objective 397  
 350 problems as bi-objective problems with total constraint vio- 398  
 351 lation as the second objective is similar to the constrained 399  
 352 handling method adopted in other filter algorithms. Nev- 400  
 353 ertheless, the selection criterion is different from the con- 401  
 354 ventional weak dominance criterion used in multi-objective 402  
 355 problems. Instead, the constrained dominance criterion as 403  
 356 introduced by Deb et al. (2000) is used in SOMPP. The con- 404  
 357 straint dominance criterion for a minimization problem is 405  
 358 defined as follows. 406

359 Solution  $i$  is said to dominate solution  $j$  if:

- 360 1. Both solutions are infeasible, and solution  $i$  has lower 407  
 361 value of constraint violation than solution  $j$  (i.e.,  $f_3^i <$  408  
 362  $f_3^j$ ) 409
- 363 2. Solution  $i$  is feasible and solution  $j$  is infeasible. 410
- 364 3. Both solutions are feasible (or problem is uncon- 411  
 365 strained) and solution  $i$  has a lower objective value than 412  
 366 solution  $j$  (that is,  $f_1^i < f_1^j$ ). 413

367 SOMPP, thus, does not follow the actual Pareto approach 414  
 368 in searching for optimal solutions. Under such circum- 415  
 369 stances the predator–prey algorithm demonstrates a desir- 416

able balance between selection of solutions based on actual 370  
 objective value and its distance from the feasible domain. 371  
 This allows one to incorporate the useful genetic traits of 372  
 strong infeasible solutions, while driving the prey popula- 373  
 tion towards the feasible domain. 374

In case of the third criterion, each old local prey is con- 375  
 sidered to be at the centre of its hypercube, the size of which 376  
 is dynamically updated with generations and is determined 377  
 by the following novel equation. 378

$$\omega = 10^{-\left(2+L\frac{t}{t_{\max}}\right)}$$

$$\eta_i = \omega \times \min\left(f_i^{\text{new prey}}, f_i^{\text{old prey}}\right) \quad (8)$$

Here,  $10^{-L}$  is the terminal order of magnitude of relative 379  
 window size,  $\omega$  is the window size of the hypercube and 380  
 $\eta_i$  is the half side length of the hypercube corresponding to 381  
 the  $i$ th objective. The first two criteria promote convergence 382  
 towards the global minimum. The third criterion helps in 383  
 maintaining diversity in the solution space in order to avoid 384  
 converging to a local minimum. Ten trials were allowed to 385  
 produce a qualified child that satisfies these three criteria, 386  
 failing which the worst prey was retained. 387

Hence, to conclude, selection in SOMPP is chiefly based 388  
 on constraint dominance. This gives feasibility a preference 389  
 over optimality, but promotes both simultaneously, which is 390  
 partially similar to the filter algorithms. At the same time, 391  
 the mutation operator and the hypercube operator incorpo- 392  
 rate the traits of niching. Niching has been applied in the 393  
 field of evolutionary algorithms using various techniques 394  
 such as dynamic mutation, preselection (Cavicchio 1970), 395  
 crowding distance concept (Dejong 1975), sharing func- 396  
 tion model (Goldberg and Richardson 1987), etc. However, 397  
 SOMPP demonstrates a search radius that is adaptive to 398  
 the extent of convergence of the population (through adap- 399  
 tive mutation) and a diversity preserving technique (the 400  
 hypercube operator) adaptive to the current diversity of the 401  
 population; the simultaneous existence of both is rare in lit- 402  
 erature. This reinforces SOMPP with the ability to adapt to 403  
 the complexities of the problem (especially multimodality) 404  
 at hand. 405

Thus, we can conclude that the preferential hunting tac- 406  
 tics of predators in the predator–prey algorithm do not 407  
 contribute any unique gain in case of unconstrained single- 408  
 objective problems. However, when dealing with con- 409  
 strained single-objective problems, such characteristic is 410  
 highly favorable to ensure simultaneous achievement of 411  
 feasibility and objective optimization. 412

Upon completion of the above predator–prey interactions 413  
 in each active locality, the predators were relocated ran- 414  
 domly. A probability based relocation criterion was intro- 415  
 duced here, which ensures that each cell is visited, therefore 416  
 favoring an even distribution of the number of visitations by 417



418 a predator to each cell. The predator relocation criterion is  
419 defined as follows:

$$\text{if } \text{cellcount}(i, j) > \text{cellcount}_{\text{avg}} + 1, \text{ locate} = \text{no} \\ \text{else}, \text{ locate} = \text{yes}$$

420 (9)

421 Here,  $\text{cellcount}(i, j)$  is the number of times predators have  
422 visited the cell  $(i, j)$  in previous generations,  $\text{cellcount}_{\text{avg}}$   
423 is the average of all  $\text{cellcount}(i, j)$  and  $(i, j)$  is the ran-  
424 domly generated location on the 2D lattice. This new feature  
425 ensures that every member of the prey population irrespec-  
426 tive of its location in the 2D lattice gets fair opportunity of  
427 improvement.

428 At the end of each generation the objective value of  
429 the strongest prey (based on dominance criterion) is found  
430 and the algorithm checks for termination. The termination  
431 criteria are as follows:

- 432 1. Maximum allowed number of function evaluations  
433 ( $f_{\text{callmax}}$ ) has been exhausted, or
- 434 2. The best objective value searched by the algorithm has  
435 not changed during the last 100 generations.

436 The dynamic reduction of the window size of the hyper-  
437 cube and the mean extent of mutation along the course of  
438 generations introduces the desirable attribute of 'adaptive  
439 shrinkage of the search radii' as solutions converge towards  
440 the global optimum.

441 The above steps summarize the basic version of SOMPP  
442 which can be termed as SOMPP Version-1. During the  
443 course of further development of SOMPP, other alter-  
444 ations/additional features were also implemented causing  
445 minor to significant improvements in its performance. These  
446 versions of SOMPP are described in detail as follows.

#### 447 2.1 SOMPP version-2: rank based predator relocation

448 Localities with relatively stronger prey were designed to  
449 have a higher affinity of attracting predators. The probabil-  
450 ity ' $\text{cellprob}_{i,j}$ ' of locating a predator in a particular locality  
451 (co-ordinates  $i, j$  generated by a random number generator)  
452 is determined as follows.

$$\text{cellrank}_{i,j} = \min \begin{pmatrix} \text{rank}_{i,j} & \text{rank}_{i+1,j} \\ \text{rank}_{i+1,j+1} & \text{rank}_{i,j+1} \end{pmatrix} \quad (10)$$

$$\text{cellprob}_{i,j} = \frac{N - \text{cellrank}_{i,j}}{N}$$

454 Here,  $\text{cellrank}_{i,j}$  is the rank of the cell/locality  $(i, j)$  and  
455  $\text{rank}_{i,j}$  is the rank of the prey located at the grid point  $(i, j)$ ,  
456 ranking being determined on the basis of dominance.  $N$

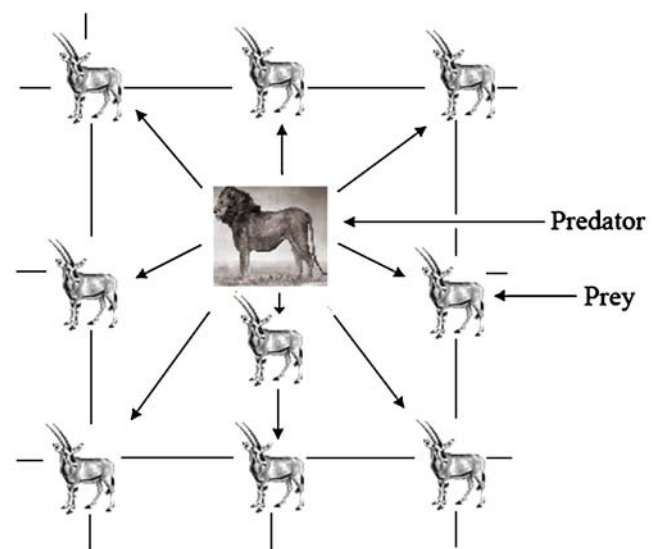
is the total number of prey, hence equal to the maximum 457  
rank in the population. This feature introduces substantial 458  
amount of elitism into the algorithm thereby speeding up 459  
convergence. However, in some cases this might limit the 460  
domain of search and hence should be applied carefully. 461

#### 462 2.2 SOMPP version-3: nine prey neighbourhood

463 Instead of the predator being located at the center of a four- 463  
vertex quadrilateral cell, the predator is now located on the 464  
same grid nodes as prey and allowed to have access to all 465  
eight preys around it as well as the prey at that very grid 466  
location (Fig. 2). This increases the neighbourhood scope of 467  
the predator from four to nine. Since prey are not relocated 468  
in SOMPP, this modification facilitates faster communica- 469  
tion of genetics among prey irrespective of their location on 470  
the unfolded toroidal surface grid, which in turn accelerates 471  
the rate of improvement of the prey population as a whole. 472

#### 473 2.3 SOMPP version-4: global elitist crossover

474 Here, the worst prey in each active neighbourhood is 474  
replaced by the crossover of the strongest two prey in the 475  
entire prey population, instead of the strongest two local 476  
prey. Strength of the prey in this case is determined on the 477  
basis of the objective value. This significantly decreased 478  
the number of function evaluations necessary, but often led 479  
to stalling of solutions at the local minima. This might be 480  
avoided by selecting the parents for crossover out of the top 481  
' $p$ ' percentage of the prey population based on dominance, 482  
instead of the two global prey with minimum objective val- 483  
ues. Nevertheless, even then the fundamental characteristics 484



485 **Fig. 2** An active nine prey locality/neighbourhood on the grid drawn  
486 on an unfolded toroidal surface

485	of predator–prey approach, that is localized evolution of	2.5 SOMPP version-6: version-5 with dominance based	521
486	solutions, will be lost.	selection in active neighbourhoods	522
487	2.4 SOMPP version-5: version-2 and version-3 combined	Here, the relative strength of the prey in an active locality is	523
488	with an epidemical operator	determined on the basis of the dominance criterion instead	524
489	In this version of SOMPP, the concepts of nine-prey active	of the weighted $f$ value given by (5). In case of uncon-	525
490	neighborhoods and rank based relocation of predators are	strained problems, this has no additional influence because	526
491	implemented simultaneously to promote faster convergence	the dominance is merely based on the actual objective value.	527
492	and better communication among the prey. However, the	However, in case of constrained problems, this modification	528
493	rank for each cell is calculated as the average of the ranks	helps significantly in directing solutions into the feasible	529
494	of all the local prey in that cell. In addition to that, to	region first, before the process of minimization takes over.	530
495	counteract the possibility of convergence to a local min-	This is because the dominance criterion (Deb 2002) was	531
496	imum, a concept of an epidemic genetic operator was	designed so that feasibility has a preference over minimiza-	532
497	introduced as implemented by Cuco et al. (2008) in the	tion. This in turn substantially reduces the domain of search	533
498	Epidemic Genetic Algorithm. If the objective value of the	at the later stages making the algorithm more robust and	534
499	strongest prey does not change over a certain number of	efficient.	535
500	consecutive iterations, a part of the prey population is dis-	The final version of SOMPP (version 6) also incorpo-	536
501	carded and replaced with new population generated using	rates rank based relocation of predators. This is a specific	537
502	Sobol's (1976) quasi-random sequence generator. This is	attribute of this single-objective version of predator–prey.	538
503	implemented as follows.	Single-objective optimization demands more focused search	539
504	If $N_{chnng} > 10$ ,	for optimal solutions compared to multi-objective prob-	540
505	1. Rank prey population by dominance.	lems. The rank based relocation ensures that the algorithm	541
506	2. Discard weakest $0.0 < f_w < 1.0$ fraction of the prey	does not waste too much time searching sections of the	542
507	population.	domain which are less likely to contain the optimal solu-	543
508	3. Set variable limits suitable to the order of magnitude of	tion. However, this can prove to be disadvantageous in cases	544
509	the remaining prey and generate $N \times f_w$ new prey to	of highly non-convex or discontinuous functions (like delta	545
510	replace the discarded ones.	functions).	546
511	Here, $N_{chnng}$ is the consecutive number of generations with-	It should be noticed that in SOMPP version 5, the	547
512	out any change in the objective value of the strongest prey	weighted sum of objectives determines the strength of prey,	548
513	by a relative tolerance of $10e-03$ . Numerical experiments	each predator being associated with a different distribu-	549
514	showed that a high value of $f_w$ ( $f_w = 0.9$ ) should be used	tion of weights. Whereas in SOMPP version 6 selection is	550
515	for all test cases since whenever the above conditions for	guided by the constraint dominance criterion. This proves to	551
516	the application of the epidemical operator was satisfied, the	be more favorable for faster convergence of solutions.	552
517	existing diversity in the population was significantly below	<b>3 Numerical experiments</b>	553
518	that required to produce new solutions. As a result of which,	All six versions of SOMPP were implemented using a	554
519	retaining a few representative solutions (strongest of the lot)	C++ programming language. The objective functions were	555
520	from the existing population, should be sufficient.	evaluated by the corresponding external executable files.	556

**Table 1** Details of three unconstrained single-objective test cases

Problem	$N_v$	Variable limits	Objective function 1	Analytical solution
Griewank	2	$x_i \in [-600, 600]$	$f(X) = \sum_{i=1}^m \frac{x_i^2}{4000} - \prod_{i=1}^m \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$f(X) = 0, x_i = 0$
Rosenbrock	2	$x_i \in [-2.048, 2.048]$	$f(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	$f(X) = 0, x_i = 1$
Miele-Cantrell	4	$x_i \in [-10, 10]$	$f(X) = (e^{(x_2 - x_1)})^4 + 100(x_2 - x_1)^6 + (\tan^{-1}(x_3 - x_4))^4 + x_1^2$	$f(X) = 0, x_1 = 0, x_2 = x_3 = x_4 = 1$

 $N_v$  number of variables

557 The final version of the algorithm, that is, SOMPP  
 558 Version-6 was initially tested on three popular uncon-  
 559 strained single-objective test functions namely Griewank  
 560 function, Rosenbrock function and Miele-Cantrell function  
 561 as evaluated by Colaco et al. (2008). Details about these  
 562 functions are given in Table 1.

563 The user-defined parameters used in the SOMPP  
 564 Version-6 algorithm in case of the above three test problems  
 565 are summarized in Table 2.

566 The prey population size used here is the 'small set' pop-  
 567 ulation size defined by Colaco et al. (2008) as equal to  
 568  $10 N_v$ . The crossover probability should be maintained at  
 569 unity (i.e., 100%) since localized recombination is abso-  
 570 lutely necessary for evolution of a population which lacks  
 571 global mixing of solutions. The mutation probability used is  
 572 also high (around 0.25, i.e., 25%) which is usual for applica-  
 573 tion of evolutionary algorithms to unconstrained problems.  
 574 However, specific real world problems might demand a  
 575 higher or lower mutation probability, which may not be  
 576 possible to predict a priori without some knowledge of the  
 577 function topology.

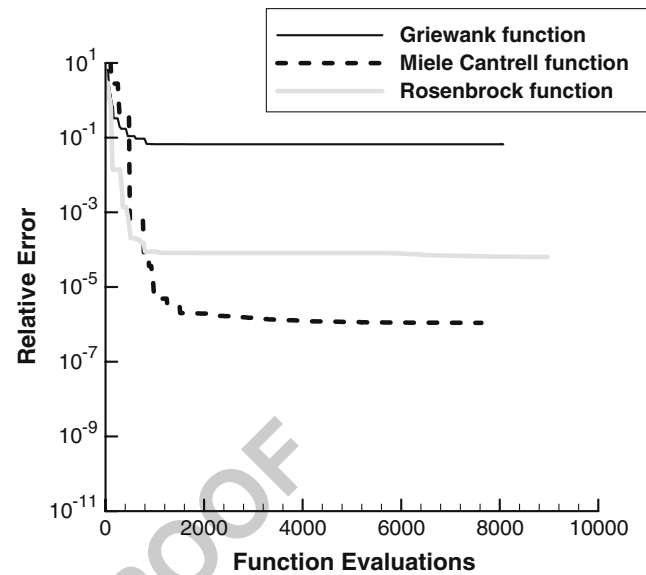
578 The values of  $K$  and  $L$  reflect the degree of conver-  
 579 gence that the user expects to achieve. However, care should  
 580 be taken to allow a sufficient number of function evalua-  
 581 tions to converge. Otherwise, the local search radii would  
 582 reduce too much and too soon rendering the algorithm  
 583 incapable of producing substantially better solutions in sub-  
 584 sequent generations. In this case, high values of  $K$  and  $L$   
 585 are used because the above test problems are unconstrained and  
 586 relatively easy to solve.

587 The test functions were run until the relative error in  
 588 the computed minima reduced to  $10e-09$  or the maximum  
 589 allowed number of function evaluations was exhausted. The  
 590 relative error was calculated as follows.

$$591 \text{ relative error} = \begin{cases} \frac{|Min_{comp} - Min_{anal}|}{Min_{anal}}, & \text{if } Min_{anal} \neq 0 \\ |Min_{comp} - Min_{anal}|, & \text{if } Min_{anal} = 0 \end{cases} \quad (11)$$

**Table 2** SOMPP Version-6 user-defined parameters for three single-objective test cases

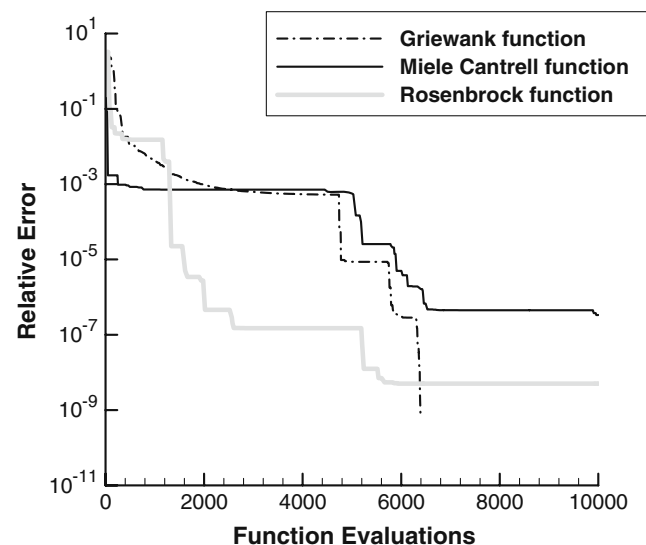
Parameter	Value
Population size (# prey)	$10 N_v$
Crossover probability	1.0
Mutation probability	0.25
Maximum allowed function evaluations	10,000
$K$ (mutation)	6
$L$ (hypercube)	10
$f_w$ (epidemic operator)	0.9



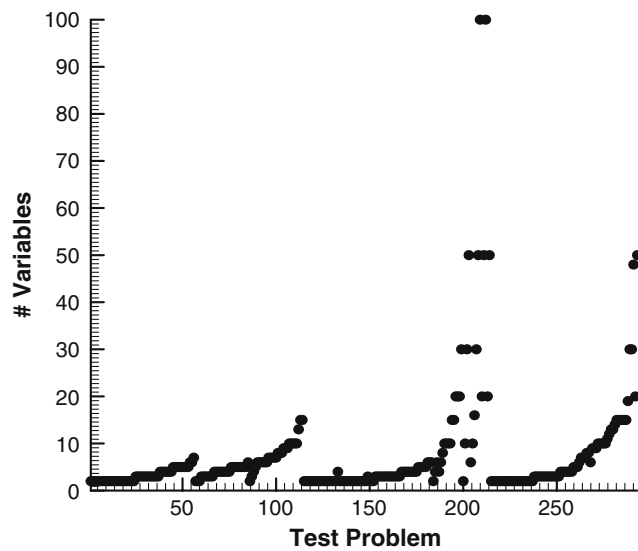
**Fig. 3** Convergence history of SOMPP version-1 applied to Griewank, Miele-Cantrell and Rosenbrock functions

592 The convergence histories of the three test problems are  
 593 shown in Figs. 3 and 4. It is noticeable that numerous mod-  
 594 ifications introduced in SOMPP Version-6 made it superior  
 595 to SOMPP Version-1.

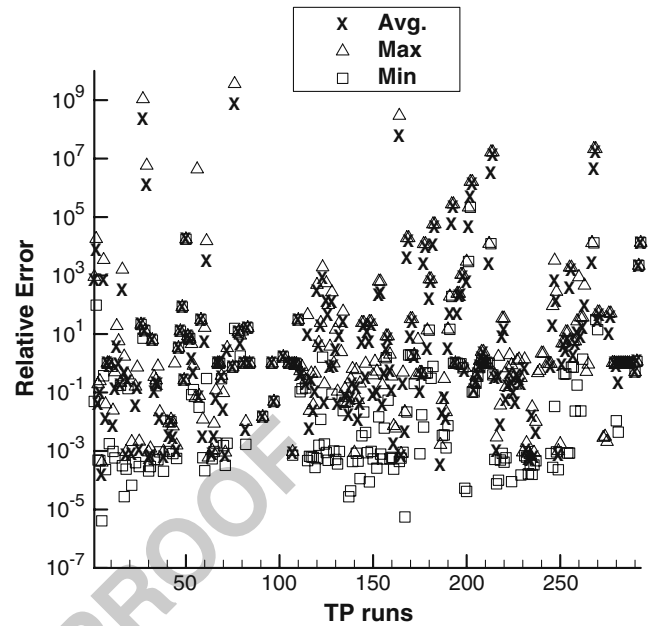
596 The results for these three test cases for SOMPP Version-  
 597 1 (Fig. 3) and SOMPP Version-6 (Fig. 4) are shown a priori  
 598 for ease of latter comparison against results of testing all  
 599 six version of SOMPP on a much larger set of test func-  
 600 tions. Though multiple runs were performed, the outcome  
 601 of only one of the representative runs is shown here due to  
 602 obvious constraints in demonstrating convergence histories  
 603 of multiple runs together on the same graph.



**Fig. 4** Convergence history of SOMPP version-6 applied to Griewank, Miele-Cantrell and Rosenbrock functions



**Fig. 5** Number of variables for each of the 293 test cases defined by Hock and Schittkowski (1981)



**Fig. 6** Relative error of computed minima for the 293 test problems (SOMPP version-1)

604 Figure 4 shows that the SOMPP Version-6 reduced the  
 605 relative error by ten orders of magnitude in less than 10,000  
 606 function evaluations for the Griewank function. However,  
 607 in case of the Rosenbrock function and the Miele–Cantrell  
 608 function the algorithm ran for 10,000 function evaluations  
 609 to reduce the relative error by ten orders and by only three  
 610 and a half orders of magnitude, respectively.

611 Further fine calibration of the extent of mutation and  
 612 the relative hypercube size together with allowing more  
 613 function evaluations is likely to achieve better accuracy in  
 614 finding the global minimum.

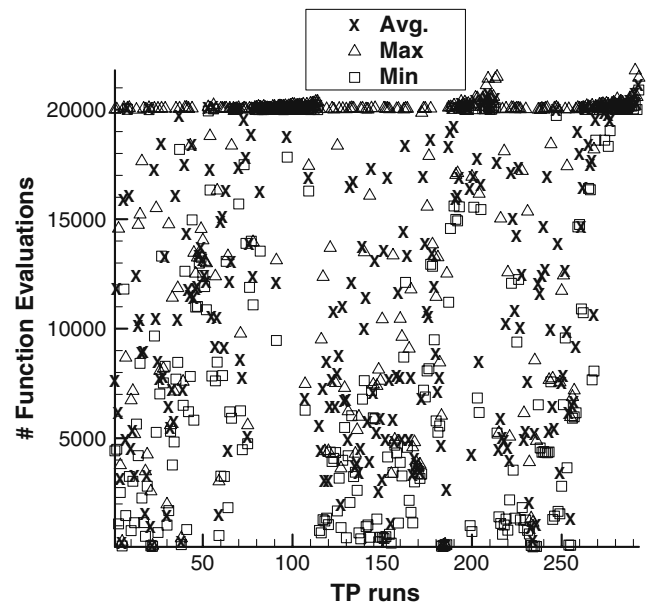
615 In order to test the SOMPP thoroughly, the algorithm  
 616 in its original version (SOMPP Version-1) was tested on  
 617 the 293 constrained and unconstrained single objective test  
 618 cases with known analytic solutions that were derived from  
 619 the collection of 395 linear/nonlinear test cases (actually  
 620 295 test problems) formulated by Hock and Schittkowski  
 621 (1981) and Schittkowski (1987). The number of variables  
 622 involved in these 293 cases ranges from two to 100 as shown  
 623 in Fig. 5. The number of inequality and equality constraints  
 624 range from 0 to 38 and 0 to 6, respectively.

The user-defined parameters used in the SOMPP 625  
 Version-1 algorithm in case of the above 293 test problems 626  
 are summarized in Table 3. 627

A lower mutation probability is used in this case, because 628  
 most of the above test cases are constrained and care should be 629  
 taken to avoid already feasible solutions (near the boundaries of 630  
 the feasible domain) from leaving the feasible space. 631  
 Similarly, lower values of K and L were used to impose 632  
 stricter restrictions on the rate of decrease of search radii, 633

**Table 3** SOMPP Version-1 user-defined parameters for the 293 test cases

Parameter	Value
Population size (# prey)	10 $N_v$
Crossover probability	1.0
Mutation probability	0.1
Maximum allowed function evaluations	20,000
K (mutation)	2
L (hypercube)	4



**Fig. 7** Number of function evaluations for each of the 293 test problems (SOMPP version-1)

**Table 4** Details of the 13 test problems from the set of 293

#	TP	$N_v$	$q$	$P$ value
1	1	2	0	0
2	37	3	0	2
3	44	4	0	6
4	55	6	6	0
5	75	4	3	2
6	110	10	0	0
7	112	10	3	0
8	118	15	0	29
9	246	3	0	0
10	251	3	0	1
11	301	50	0	0
12	393	48	2	1
13	395	50	1	0

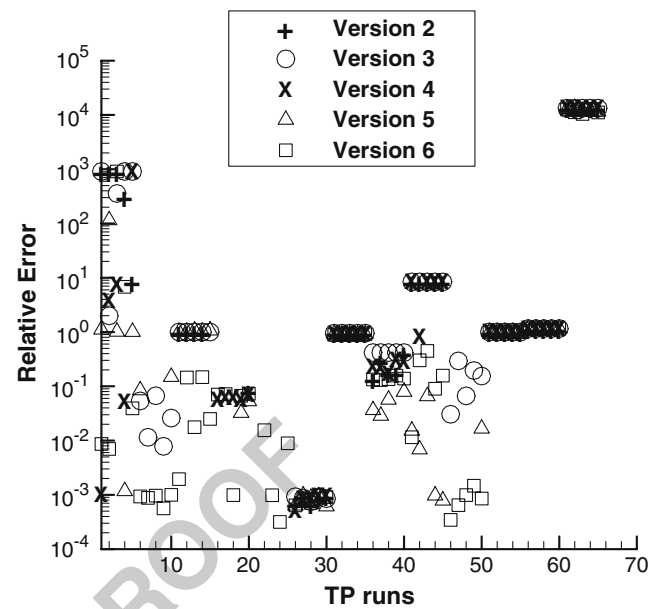
634 since a lower degree of convergence is expected for such  
 635 a set of complex constrained/unconstrained test problems.  
 636 A relative tolerance of  $\varepsilon = 0.001$  was used for equality  
 637 constraints.

638 To compensate for performance fluctuations induced by  
 639 random generators used in creating the initial population  
 640 and other genetic operators, the algorithm was run five times  
 641 for each of the 293 test problems resulting in a total of 1,465  
 642 test runs. An explicit termination criterion was also imple-  
 643 mented when relative error became less than 0.001. The  
 644 final relative error for the computed minimum and the num-  
 645 ber of function evaluations exhausted in doing so for each  
 646 of these test problems can be seen in Figs. 6 and 7. In both  
 647 of these figures the corresponding maximum, minimum and  
 648 average (of five runs) are given for each test problem.

649 It is evident from Fig. 6 that some of the test cases exhibit  
 650 partial convergence with a relative error of the order of  
 651 around 1.0. This can be attributed to the presence of either  
 652 multiple equality or inequality constraints (linear /nonlin-  
 653 ear) or both in most of these test problems (Hock and  
 654 Schittkowski 1981; Schittkowski 1987). Some of the test  
 655 cases do not converge at all leading to a relative error of  
 656 orders above unity. This is primarily due to the lack of any

**Table 5** SOMPP user-defined parameters for the 13 test cases

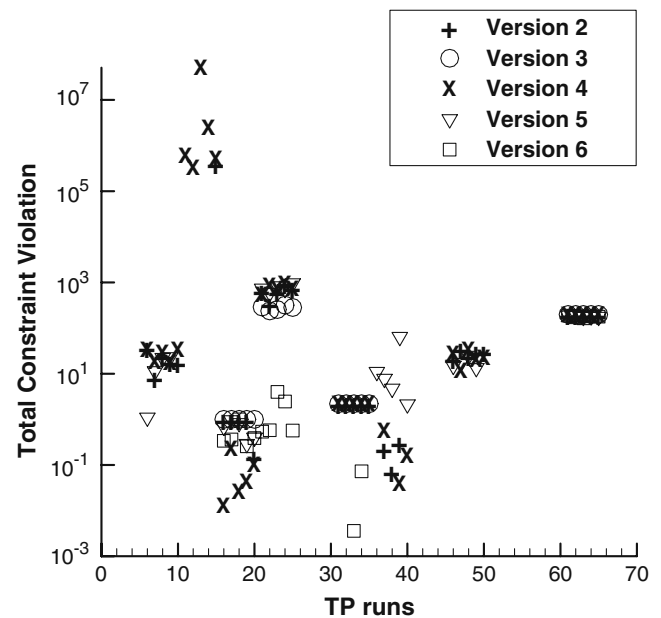
Parameter	Value
Population size (# prey)	10 $N_v$
Crossover probability	1.0
Mutation probability	0.25
Maximum allowed function evaluations	20,000
$K$ (mutation)	3
$L$ (hypercube)	6
$f_w$ (epidemic operator)	0.9



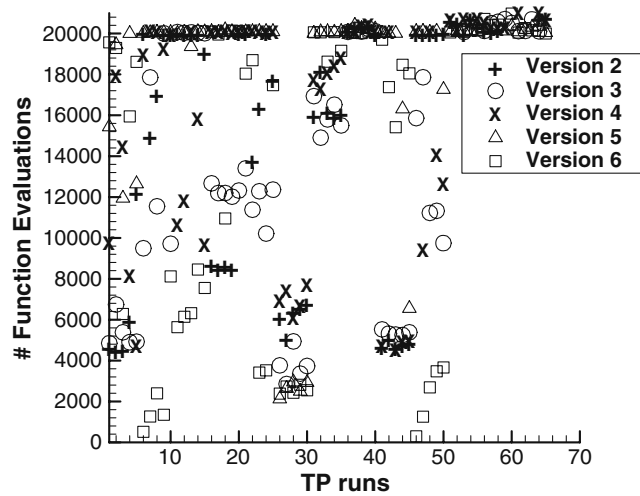
**Fig. 8** Relative error of computed minima for the 13 test problems

657 specified variable limits for some of the variables in the origi-  
 658 nal publications. In such cases, a comprehensive range of  
 659  $-10e10$  to  $+10e10$  was assigned for each design variable.

660 The number of function evaluations varied significantly  
 661 from problem to problem as seen from Fig. 7. Test problems  
 662 (TP) from TP-80 onwards till TP-118 (test runs 400–590)  
 663 have relatively high number of constraints leading to a  
 664 higher number of function evaluations. Whereas test prob-  
 665 lems ranging from TP-190 to TP-210 as well as from  
 666 TP-260 to TP-293 have a relatively high number of design



**Fig. 9** Total constraint violation for each of the 13 test problems that are constrained



**Fig. 10** Number of function evaluations for the 13 test problems

667 variables leading also to a higher number of the objective  
 668 function evaluations. Though 20000 function evaluations  
 669 were allowed, some test cases show total executed num-  
 670 ber of evaluations to be a little more than that. This is  
 671 because in SOMPP the number of function evaluations in  
 672 each generation is not limited to the population size (not  
 673 predictable either) unlike in other evolutionary algorithms.  
 674 Consequently, the total number of function evaluations  
 675 might just exceed that allowed in course of the last executed  
 676 generation. This is also evident from Table 6 presented later  
 677 in the paper.

678 Running all 293 test problems in series is computationally  
 679 extremely time consuming. Consequently, a set of 13

test problems were chosen from among these 293 cases. 680  
 These 13 test cases involve number of variables ranging 681  
 from two to 50 (with or without specified limits), num- 682  
 ber of equality constraints ranging from 0 to 6 and number 683  
 of inequality constraints ranging from 0 to 38, thereby 684  
 exhibiting varying degree and nature of complexity. Details 685  
 pertinent to these test problems are given in Table 4. It 686  
 should be noted that, compared to Table 3, a higher mutation 687  
 probability was used to prevent intermediate convergence 688  
 to local minima and subsequent stagnancy in the region of 689  
 the local minima. Higher values of  $K$  and  $L$  were used to 690  
 achieve better accuracy. 691

Here,  $p$  = number of inequality constraints,  $q$  = number 692  
 of equality constraints. 693

All the latter five versions of SOMPP (versions 2 to 6) 694  
 were tested on these 13 test problems. Each of these test 695  
 problems was run five times on a small population size 696  
 ( $10 N_v$ ) as before. The user-defined parameters used in the 697  
 SOMPP algorithm in case of these 13 test problems are 698  
 summarized in Table 5. 699

The relative error of the computed minima, the con- 700  
 straint violation of the computed minima, and the number 701  
 of function evaluations exhausted for each of the five ver- 702  
 sions of SOMPP running on each of the 13 test problems 703  
 thus resulting in 65 runs can be seen in Figs. 8, 9 and 10. 704

It can be observed from Fig. 8 that SOMPP Version- 705  
 6 performs better than the other versions of SOMPP in 706  
 approaching the global minima. It also has the maximum 707  
 potential in driving solutions into the feasible domain as 708  
 seen from Fig. 9. In case of some of the constrained prob- 709  
 lems the data points are not visible in Fig. 9. This is because 710  
 the constraint violation is zero, that is the final computed 711  
 minima in these cases are feasible solutions, and hence 712

**Table 6** Output for the 13 test problems with SOMPP Version-6

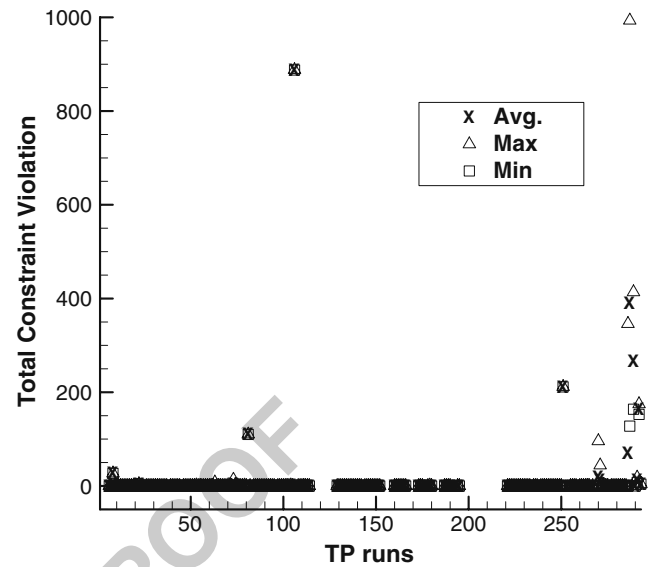
TP	Computed minima	Actual minima	Relative error	Constraint violation	Number of function evaluations	Computing time (s)
1	0.00701	0.00	0.00701		19,291	989
37	-3,454.06	-3,456	0.00056	0	1,347	69
44	-14.9708	-15.00	0.00195	0	5,635	290
55	6.33959	6.3333	0.00098	0.996963	10,952	568
75	5,176.05	5,174.41	0.00031	2.45536	3,522	182
110	-45.7493	-45.7785	0.00064		2,385	123
112	-0.05151	-0.47761	0.89215	0	20,059	1,045
118	751.617	664.82	0.130556	0	20,031	1,045
246	0.011518	0.00	0.011518		19,696	1,021
251	-3,454.81	-3,456	0.000345	0	294	15
301	0	-50	1.000000		20,052	1,062
393	1.8623	0.86338	1.15699	0	20,712	1,192
395	19,990.6	1.91667	10,428.9	163.789	20,150	1,071

713 cannot be represented in a logarithmic plot of Fig. 9. The  
 714 relevant output parameters relating to the most accurate  
 715 solution (of the five runs for each problem) for SOMPP  
 716 Version-6 running on the 13 cases are summarized in  
 717 Table 6.

718 The significantly low accuracy and inability to find feasi-  
 719 ble solutions in case of TP-395 can be attributed to the fact  
 720 that there were no specified variable limits for any of the  
 721 50 design variables involved in this problem provided in the  
 722 original publications (Schittkowski 1987).

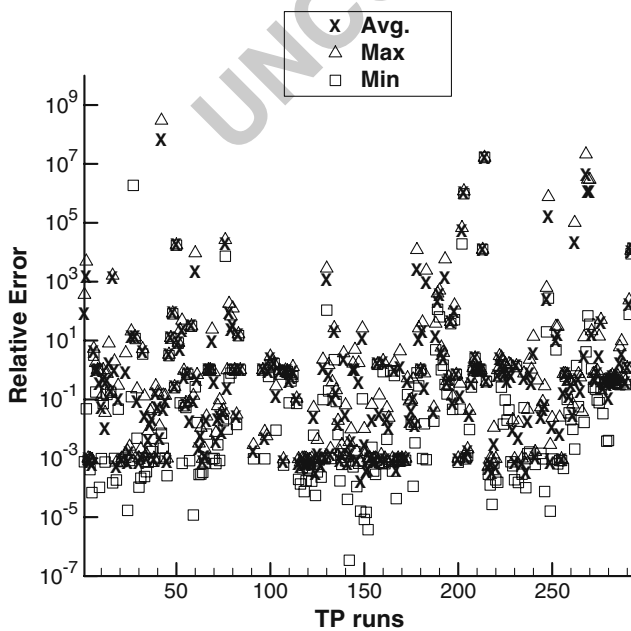
723 SOMPP Version-6 being the most efficient and robust of  
 724 all the different forms of the SOMPP, was then tested on the  
 725 entire set of 293 single objective test problems (Hock and  
 726 Schittkowski 1981; Schittkowski 1987) run five times each.  
 727 The various user-defined parameters used were the same as  
 728 given in Table 5. The relative error of the computed mini-  
 729 ma, the constraint violation of the computed minima and  
 730 the number of function evaluations exhausted for all the 293  
 731 test runs are displayed in Figs. 11, 12 and 13 respectively. In  
 732 all the three figures, the corresponding maximum, minimum  
 733 and average (of five runs) are given for each test problem.

734 It is seen from Fig. 11 that SOMPP Version-6 performs  
 735 well in achieving relative error of the order of less than 1.0,  
 736 except for in cases which have a high number of design  
 737 variables with unspecified variable limits. However, the  
 738 most prominent improvement of this version of SOMPP is  
 739 its ability to find the feasible space in case of constrained  
 740 problems (as shown in Fig.12) irrespective of the number  
 741 and complexity of the inequality and equality constraints  
 742 (whether linear or nonlinear). It should be noted that in

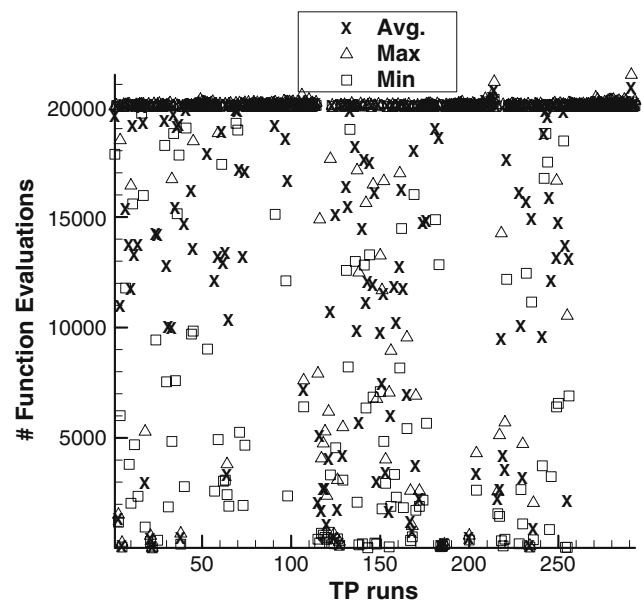


**Fig. 12** Total constraint violation for each of the 293 test problems that are constrained (SOMPP version-6)

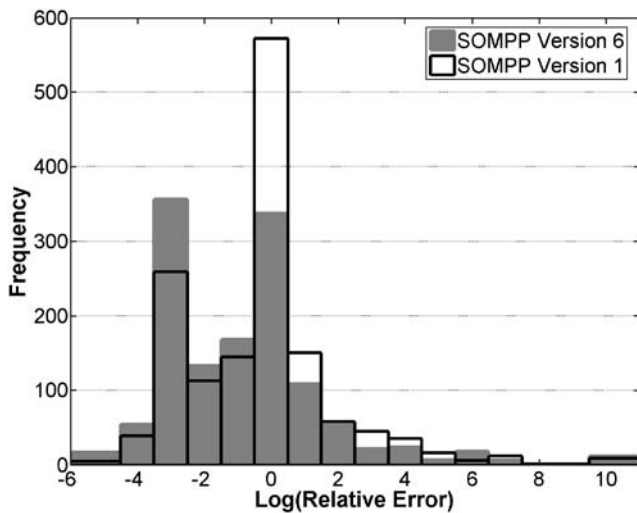
many of these constrained problems the initial population is 743  
 completely in the infeasible space. The inability to converge 744  
 to the feasible space in case of the last few test problems 745  
 can be attributed to the involvement of relatively high number 746  
 of design variables (from 20 to 50) as seen from Fig. 5. 747  
 The number of function evaluations exhausted by SOMPP 748  
 Version-6 is relatively high as shown in Fig. 13, which 749



**Fig. 11** Relative error of computed minima for the 293 test problems (SOMPP version-6)



**Fig. 13** Number of function evaluations made for the 293 test problems (SOMPP version-6)



**Fig. 14** Comparison of the frequency of occurrence of different orders of magnitude of relative error in the computed minima between SOMPP version-1 and SOMPP version-6. Note: frequency is the number of test runs that converged to that particular order of magnitude of relative error

750 is expected as a substantial amount of functions evaluations are consumed in successfully searching for the feasible space in case of constrained problems.

751  
752  
753 The improved performance of SOMPP Version-6 becomes more evident from the histogram presented in Fig. 14.

754  
755  
756 It is seen from Fig. 14 that in case of SOMPP Version-6, more test cases have converged to relative errors of orders of magnitude less than 1.0 (higher histogram bars for log (relative error)  $\leq 0$ ).

760 **4 Conclusion**

761 All versions of the predator-prey algorithm that exist in literature are mostly suited for unconstrained multiobjective optimization problems. Consequently, the predator-prey algorithm in its modified form (SOMPP) is the first of its kind that specifically deals with constrained single-objective optimization problems. It performs well on the popular unconstrained test functions, namely Griewank, Rosenbrock and Miele-Cantrell functions. The 293 single-objective test problems given by Hock and Schittkowski (1981) and Schittkowski (1987) form the most expansive set of single objective test functions (both constrained and unconstrained and linear and nonlinear) available in the literature. SOMPP performs satisfactorily on a large number of these test problems, in driving solutions into the feasible domain and consequently converging to the global mini-

776 mum, using a relatively frugal population size defined by the 'small set', i.e. ten times the number of design variables Colaco et al. (2008). However, the accuracy of SOMPP is noticeably affected by the absence of specified limits of design variables especially in problems with a large number of design variables.

777  
778  
779  
780  
781  
782 SOMPP proves expensive in terms of function evaluations when dealing with multiple equality/inequality constraints. This can be attributed to the fact that a substantial amount of function calls are consumed in search of the feasible domain. This expense increases significantly with increase in the dimensionality of the problem, which is however a generic problem with any kind of evolutionary algorithm. Another drawback of SOMPP is that the algorithm demands fine tuning of three user-defined parameters namely the mutation probability, the relative hypercube window size  $L$ , and the relative extent of mutation  $K$ . Depending upon the problem, a value of 0.05 to 0.25 is suggested for the probability of mutation, whereas values of  $K$  and  $L$  are subject to the convergence expected with  $L - K \geq 2$  always. Nevertheless, coupling SOMPP with an efficient response surface model that interpolates both linear and highly non linear functions in multidimensional spaces (Colaco et al. 2008) is expected to improve the robustness and accuracy of the SOMPP algorithm considerably.

801 **Acknowledgments** The authors are grateful for the financial support provided for this work by the US Air Force Office of Scientific Research under grant FA9550-06-1-0170 monitored by Dr. Todd E. Combs, Dr. Fariba Fahroo and Dr. Donald Hearn and by the US Army Research Office/Materials Division under the contract number W911NF-06-1-0328 monitored by Dr. William M. Mullins. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US Air Force Office of Scientific Research, the US Army Research Office or the US Government. The US Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

814 **References**

815 Cavicchio DJ (1970) Adaptive search using simulated evolution. PhD thesis. Ann Arbor, MI: University of Michigan  
816  
817 Chowdhury S, Dulikravich GS, Moral RJ (2009) Modified predator-prey (MPP) algorithm for constrained multi-objective optimization. In: Burczynski T (ed) EUROGEN 2009, Cracow, Poland  
818  
819 Chowdhury S, Dulikravich GS, Moral RJ (2010) Modified predator-prey algorithm for constrained and unconstrained multi-objective optimization problems. Int J Math Model Numer Optim (in press)  
820  
821 Coello Coello C (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 191(11-12):1245-1287  
822  
823 Colaco MJ, Dulikravich GS, Orlando HRB, Martin TJ (2005) Hybrid optimization with automatic switching among optimization  
824  
825  
826  
827  
828



- 829 algorithms. In: Annicchiarico W, Périaux J, Cerrolaza M, Winter  
830 G (eds) A chapter in evolutionary algorithms and intelligent tools  
831 in engineering optimization. WIT, CIMNE, Barcelona, pp 92–  
832 118
- 833 Colaco MJ, Dulikravich GS, Sahoo D (2008) A response surface  
834 method-based hybrid optimizer. *Inverse Probl Sci Eng* 16(6):717–  
835 741
- 836 Cuco APC, Neto AJS, Velho HFC, de Sousa FLD (2008) Solu-  
837 tion of an inverse adsorption problem with an epidemic genetic  
838 algorithm and the generalized extremal optimization algorithm.  
839 *Inverse Probl Sci Eng* 16(8):901–914
- 840 Deb K (2002) Multi-objective optimization using evolutionary algo-  
841 rithms. Wiley, Chichester
- 842 Deb K, Agarwal S, Pratap A, Meyarivan T (2000) A fast elitist non-  
843 dominated sorting genetic algorithm for multi-objective optimiza-  
844 tion: NSGA-II. In: Proceedings of the parallel problem solving  
845 from nature VI conference, pp 849–858
- 846 Dejong KA (1975) An analysis of the behavior of a class of genetic  
847 adaptive systems. PhD thesis. Ann Arbor, MI: University of  
848 Michigan. Dissertation Abstracts International 36(10), 5140B  
849 (University Microfilms No. 76–9381)
- 850 Dulikravich GS, Martin TJ, Dennis BH, Foster NF (1999) Multi-  
851 disciplinary hybrid constrained GA optimization. In: Miettinen  
852 K, Makela MM, Neittaanmaki P, Periaux J (eds) A chapter in  
853 EUROGEN'99- evolutionary algorithms in engineering and com-  
854 puter science: recent advances and industrial applications. Wiley,  
855 Jyvaskyla, pp 233–259
- 856 Goldberg DE, Richardson J (1987) Genetic algorithms with shar-  
857 ing for multimodal function optimization. In: Proceedings of  
858 the first international conference on genetic algorithms and their  
859 applications, pp 41–49
- Grimme C, Schmitt K (2006) Inside a predator–prey model for mul- 860  
tiobjective optimization—a second study. In: Proc. genetic and 861  
evolutionary computation conf. GECCO 2006. Seattle WA. ACM, 862  
New York, pp 707–714 863
- Hock W, Schittkowski K (1981) Test examples for nonlinear pro- 864  
gramming codes. Lecture notes in economics and mathematical 865  
systems, vol 187. Springer, Berlin 866
- Laumanns M, Rudolph G, Schwefel HP (1998) A spatial predator–prey 867  
approach to multi-objective optimization: a preliminary study. 868  
*Lect Notes Comput Sci* 1498:241–249 869
- Li X (2003) A real-coded predator–prey genetic algorithm for multi- 870  
objective optimization. *Lect Notes Comput Sci* 2632:69 871
- Martin TJ, Dulikravich GS (2002) Analysis and multidisciplinary opti- 872  
mization of internal coolant networks in turbine blades. *AIAA J* 873  
*Propuls Power* 18(4):896–906 874
- Michalewicz J (1992) Genetic algorithms + data structures = evolu- 875  
tionary programs. Berlin, Springer-Verlag 876
- Moral MJ, Dulikravich GS (2008) Multi-objective hybrid evolution- 877  
ary optimization utilizing automatic algorithm switching. *AIAA J* 878  
46(3):673–700 879
- Schittkowski K (1987) More test examples for nonlinear programming. 880  
Lecture notes in economics and mathematical systems, vol 282. 881  
Springer, New York 882
- Silva A, Neves E, Costa E (2002) An empirical comparison of particle 883  
swarm and predator–prey optimization. *Lect Notes Comput Sci* 884  
2464:103–110 885
- Sobol IM (1976) Uniformly distributed sequences with an additional 886  
uniform property. *USSR Comput Math Math Phys* 16:236–242 887
- Venter G, Haftka RT (2009) Constrained particle swarm optimization 888  
using a bi-objective formulation, structural and multidisciplinary 889  
optimization. *Struct Multidisc Optim* (in press) 890

---

## AUTHOR QUERY FORM

Dear Author,

During the preparation of your manuscript for typesetting, some questions have arisen. These are listed below. Please check your typeset proof carefully and mark any corrections in the margin of the proof or compile them as a separate list.

### Queries and/or remarks

Location in Article	Query / remark
Reference List	Please provide updated publication status for Venter and Haftka (2009) and Chowdhury et al. (2010).

Thank you for your assistance