Extension of the Firefly Algorithm and Preference Rules for Solving MINLP Problems

M. Fernanda P. Costa[1,a)], Rogério B. Francisco[2,b)], Ana Maria A.C. Rocha[3,c)] and Edite M.G.P. Fernandes[3,d)]

[1]*Centre of Mathematics, University of Minho, 4710-057 Braga, Portugal*
[2]*School of Technology and Management Felgueiras, 4610-156 Felgueiras, Portugal*
[3]*Algoritmi Research Centre, University of Minho, 4710-057 Braga, Portugal*

[a)]Corresponding author: mfc@math.uminho.pt
[b)]rbf@estgf.ipp.pt
[c)]arocha@dps.uminho.pt
[d)]emgpf@dps.uminho.pt

Abstract. An extension of the firefly algorithm (FA) for solving mixed-integer nonlinear programming (MINLP) problems is presented. Although penalty functions are nowadays frequently used to handle integrality conditions and inequality and equality constraints, this paper proposes the implementation within the FA of a simple rounded-based heuristic and four preference rules to find and converge to MINLP feasible solutions. Preliminary numerical experiments are carried out to validate the proposed methodology.

INTRODUCTION

In this paper, we address the problem of solving mixed-integer nonlinear programming (MINLP) problems by an extended version of the firefly algorithm (FA) that is capable of computing solutions with continuous and integer variables simultaneously. The MINLP problem has the form

$$\begin{array}{rl} \min_{(x,y)} & f(x,y) \\ \text{subject to} & g_j(x,y) \leq 0,\ \ j = 1,\ldots,p \\ & h_l(x,y) = 0,\ \ l = 1,\ldots,m \\ & x \in \Gamma_x,\ y \in \Gamma_y \cap \mathbb{Z}^{n_i} \end{array} \tag{1}$$

where the vector x represents the continuous variables with $\Gamma_x \subset \mathbb{R}^{n_c}$ being the set of simple bounds on x (or a polyhedral subset of $\mathbb{R}^{n_c}$) $\Gamma_x = \{x \in \mathbb{R}^{n_c} :\ l_x \leq x \leq u_x,\ l_x, u_x \in \mathbb{R}^{n_c}\}$. The integer variables are represented by the vector y, where Γ_y is the set of simple bounds on y, $\Gamma_y = \left\{y \in \mathbb{Z}^{n_i} :\ l_y \leq y \leq u_y,\ l_y, u_y \in \mathbb{Z}^{n_i}\right\}$, and the parameter $n = n_c + n_i$ represents the total number of variables. The function $f : \Gamma_x \times \Gamma_y \rightarrow \mathbb{R}$ is the objective function and $g : \Gamma_x \times \Gamma_y \rightarrow \mathbb{R}^p$ and $h : \Gamma_x \times \Gamma_y \rightarrow \mathbb{R}^m$ are the inequality and equality constraint functions respectively. Let $V(x,y)$ be the non-negative real function that is used to measure inequality and equality constraint violation:

$$V(x,y) = \sum_{j=1}^{p} \max\{0, g_j(x,y)\} + \sum_{l=1}^{m} |h_l(x,y)|.$$

In the following, a solution (x,y) that satisfies the inequality and equality constraints, i.e., that verifies $V(x,y) = 0$ is denoted by a constraint-feasible solution; if $V(x,y) > 0$, the point (x,y) is a constraint-infeasible solution.

The difficulties that arise when solving this type of problem are related to the integrality of the variables y and to the nonlinear functions in the objective and constraints. Penalty-based techniques have been used to handle integrality as well constraint violations [1, 2]. Although popular, their use turn out to be a challenging issue since they require finding suitable values for the penalty parameters.

In this paper, to overcome the penalty parameter issue, nonlinear inequality and equality constraints are handled by a set of four preference rules that favor constraint-feasible solutions against infeasible ones but without ignoring them. First, in the next section, we present the four preference rules that are used to select the best of two compared solutions; afterwards, we describe the heuristics used to handle integer variables in the FA.

HANDLING CONSTRAINTS BY PREFERENCE RULES

A quite common and popular constraint handling technique, presented in [3], relies on three rules that favor feasible points against infeasible ones. The constraint violation and the objective function values are used and compared separately in order to decide which point would be preferred between the two. The technique has been used with different metaheuristics, like the genetic algorithm, evolutionary algorithm, differential evolution, electromagnetism-like algorithm [3, 4, 5] Recently, a new rule has been added to the set with the goal of solving nonlinear global optimization problems [6]. The authors extended the FA to general constrained global optimization problems by exploring two strategies, one based on the global competitive ranking and the other on the below four described rules. The following rules, herein denoted by "preference rules", aim to compare two solutions so that the one that would be preferred is identified:

(P_1) Any constraint-feasible solution is preferred to any constraint-infeasible one.
(P_2) Between two constraint-feasible solutions, the one with lower objective function value f is preferred.
(P_3) Between two constraint-infeasible solutions, the one with lower constraint violation V is preferred.
(P_4) Between two constraint-infeasible solutions, the one having smaller number of violated constraints is preferred.

The rules (P_1) – (P_3) are the well-known feasibility and dominance rules proposed for binary tournaments [3]. The rule (P_4) means that a solution (x^l, y^l) with a smaller number of violated constraints would be preferred to any other (x^k, y^k) with a larger number of violated constraints, whatever the relation between $V(x^l, y^l)$ and $V(x^k, y^k)$.

FIREFLY ALGORITHM EXTENSION TO MIXED-INTEGER VARIABLES

A simple rounding-based heuristic is used to handle continuous and integer variables simultaneously in the FA. While the movement of the continuous variables in the search space proceeds as usual, the movement of the integer variables is controlled by search steps that take only integer values. This is achieved by using an operator that finds the nearest integer value to a real value. FA is a stochastic population-based algorithm for solving continuous bound constrained global optimization problems. It is inspired by the flashing behavior of fireflies at night. The FA was developed by [7] and the firefly movements are based on the following three main rules:

(i) all fireflies are unisex, meaning that any firefly can be attracted to any other of the brighter ones;
(ii) the attractiveness of a firefly is determined by its brightness which is associated with the encoded objective function;
(iii) the attractiveness is directly proportional to brightness but decreases with the distance.

In the context of solving MINLP problems, the position of a firefly l will be represented by the point $(x^l, y^l) \in \Gamma_x \times \Gamma_y$ and firefly k is brighter than firefly l if the point (x^k, y^k) is preferred to the point (x^l, y^l) according to the "preference rules" (P_1) – (P_4). We assume that the number of fireflies in the population is NP. At the beginning, the positions of the fireflies in the population are randomly generated. Then, they are evaluated, by computing f, V and the number of violated constraints. Afterwards, based on rules (P_1) – (P_4), the points are ranked in ascending order, i.e., (x^1, y^1) is the brightest firefly, (x^2, y^2) is the second brightest and so on. Subsequently, the fireflies are moved following rules (i) – (iii) and according to the below shown equations (3) and (4).

To randomly generate the positions of the NP fireflies, each with n_c continuous variables and n_i integer variables, in $\Gamma_x \times \Gamma_y$, we use for $k = 1, \ldots, NP$:

$$\begin{aligned} x_i^k &= (l_x)_i + \lambda_i^k((u_x)_i - (l_x)_i) && \text{for } i = 1, \ldots, n_c \\ y_j^k &= (l_y)_j + \tau_j^k && \text{for } j = 1, \ldots, n_i \end{aligned} \tag{2}$$

where the notation $(l_x)_i$ represents the component i of the vector l_x (similarly for $(u_x)_i, (l_y)_j, (u_y)_j$), λ_i^k is a number uniformly distributed in $[0, 1]$ and τ_j^k is a number randomly selected from the set $\{0, 1, \ldots, ((u_y)_j - (l_y)_j)\}$. The movement of fireflies in the original version of FA is related with the brightness emitted by each firefly and the degree of attractiveness that is generated between two fireflies. Now, extending these ideas to the mixed-integer variables, a firefly k ($k = 2, \ldots, NP - 1$) is moved towards the brighter fireflies $l = 1 \ldots, k - 1$ as follows:

$$\begin{aligned} x_i^k &= x_i^k + \beta(x_i^l - x_i^k) + \alpha(\lambda_i^k - 0.5)X_i && \text{for } i = 1, \ldots, n_c \\ y_j^k &= y_j^k + RND\left[\beta(y_i^l - y_i^k) + \alpha(\lambda_j^k - 0.5)Y_j\right] && \text{for } j = 1, \ldots, n_i \end{aligned} \tag{3}$$

where the operator $RND[z]$ gives the nearest integer vector to the real z, α is the randomization parameter defined by the user, usually a number in the range $[0, 1]$, and X_i and Y_j aim to scale the movement of the continuous and integer variables to the sets Γ_x and Γ_y respectively. We note that, based on rule (i) above, the brightest firefly (x^1, y^1) will not be moved. In this paper, a different movement is used to define a trial position for the worst firefly (x^{NP}, y^{NP}). A random movement is carried out componentwise around (x^1, y^1). After all movements have been carried out, the trial position of each firefly is checked against the corresponding bounds and a projection on to the boundary is carried out if the components fall outside. Then, the final trial position of each firefly is compared with its initial position according to rules (P_1) – (P_4) and the preferred one is maintained for the next iteration.

The parameter β in (3) gives the attractiveness between fireflies k and l, it varies with the brightness seen by adjacent fireflies and the distance between themselves [7], and is defined as follows

$$\beta = \beta_0 \exp\left(-\gamma \left\|(x, y)^l - (x, y)^k\right\|_2^2\right), \tag{4}$$

where the constant β_0 is the attraction parameter when the distance is zero. The parameter γ is used to increase the attractiveness between fireflies and it varies along the iterative process [8]. The randomization parameter α is made to decrease gradually as the iterative process proceeds and aims to increase diversity at the beginning of the process, see [2, 8] for details.

Finally, an intensification phase is included in the final step of the FA. A derivative-free method from the pattern local search class, known as Hooke-and-Jeeves (HJ) [9], is invoked starting from the brightest firefly. The HJ local search is implemented $5n$ iterations and has been extended to handle continuous and integer variables simultaneously by using the operator $RND[\cdot]$, similarly to (3).

NUMERICAL EXPERIMENTS

In order to analyze the performance of the extended FA based on the preference rules, we carried out preliminary numerical experiments with a set of small well-known MINLP problems available in the literature. We use a PC Intel Core 2 Duo Processor E7500 with 2.9GHz and 4Gb of memory RAM and the algorithm was coded in Matlab^{TM}Version 8.1 (R2013a). The results are shown in Table 1 that depicts the best obtained objective function value, f_{best}, and the corresponding violation, V_{best}, (out of the 30 independent runs), the average f of the 30 results, f_{avg}, and the average number of function evaluations, $n.f.eval$, required to reach a solution $(x^1(K), y^1(K))$ (at iteration K) that satisfies the stopping conditions

$$\frac{\left|f\left(x^1(K), y^1(K)\right) - f^*\right|}{|f^*|} \leq 1.0E - 4 \quad \text{and} \quad V\left(x^1(K), y^1(K)\right) \leq 1.0E - 3$$

where f^* is the best known optimal solution. If these conditions are not satisfied after $10n$ iterations, the algorithm stops and provides the best solution found thus far. Results in the table, collected from [10] (a filter-based genetic algorithm (GA)) and [11] (an extended version of the ant colony optimization (ACO)), correspond to a population of 20 points, 30 independent runs, and termination after 10 000 function evaluations or when a solution with error $1E - 04$ is reached. As it can be seen, the set of four preference rules when integrated into the extended FA yields a rather competitive methodology for solving MINLP problems. Since there is no penalty parameter to initialize and tune, the algorithm is easy to code and quickly implemented.

TABLE 1. Numerical results for $NP = 20$.

Prob.	n_c/n_i	f^*	extended FA based on (P_1)–(P_4)				Filter-based GA in [10]		Extended ACO in [11]	
			f_{best}	V_{best}	f_{avg}	$n.f.eval$	f_{avg}	$n.f.eval$	f_{avg}	$n.f.eval$
ex 12.2.1*	2/3	7.66718	7.6671	1.87E-04	8.0695	8622	7.7406	4720	7.6672	363
ex 12.2.2*	2/1	1.07654	1.0766	0.00E+00	1.0767	5178	1.0767	8074	1.1459	4250
ex 12.2.3*	3/4	4.57958	4.5796	0.00E+00	4.7758	12157	5.1322	8125	4.5796	731
ex 12.2.6*	1/1	-17.0000	-17.0000	5.44E-07	-16.9998	3243	-17	999	-17	307
st_e13†	1/1	2.0000	2.0000	0.00E+00	2.0000	3409	2.0000	4530	–	–
f_2 in [10]	2/1	2.124	2.1320	2.75E-07	2.7149	5253	2.1852	3799	–	–
f_{10} in [10]	1/1	-2.444	-2.4380	0.00E+00	-2.4380	3501	-2.4444	230	-2.4444	270
f_{11} in [10]	2/1	3.2361	3.2361	0.00E+00	3.2361	4405	3.4208	5616	23.475	1180

* Test problems from [12] (http://titan.princeton.edu/TestProblems/chapter12.html).
† Test problem in [13] (http://www.gamsworld.org/minlp/minlplib/points.htm); – means not available.

CONCLUSIONS

We have presented a simple heuristic to handle integer variables inside the firefly algorithm developed for continuous variables. The algorithm has also been enhanced with an intensification phase by using the Hooke-and-Jeeves pattern search-type method. The issue related to the inequality and equality constraints of the problem has been addressed by four preference rules that favor feasible solutions against infeasible ones. The preliminary numerical experiments show that the new methodology is rather competitive when compared with other stochastic-based techniques available in the literature for solving MINLP problems.

ACKNOWLEDGMENTS

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e Tecnologia, within the projects UID/CEC/00319/2013 and UID/MAT/00013/2013.

REFERENCES

[1] S. Lucidi and F. Rinaldi, J. Optim. Theory Appl. **145**(3), 479–488 (2010).
[2] M.F.P. Costa, A.M.A.C. Rocha, R.B. Francisco, E.M.G.P. Fernandes, Optimization **65**(5), 1085–1104 (2016).
[3] K. Deb, Comput. Methods Appl. Mech. Eng. **186**(2-4), 311–338 (2000).
[4] I.A.C.P.E. Santo, L. Costa, A.M.A.C. Rocha, M.A.K. Azad, E.M.G.P. Fernandes, "On challenging techniques for constrained global optimization" in *Handbook of Optimization*, edited by I. Zelinka et al., (Springer-Verlag Berlin Heidelberg 2013) pp. 641–671.
[5] A.M.A.C. Rocha and E.M.G.P. Fernandes, "Feasibility and dominance rules in the electromagnetism-like algorithm for constrained global optimizatio" in *ICCSA 2008*, Part II, Lect. Notes Comput. Sc. 5073, edited by O. Gervasi et al., (Springer-Verlag Berlin Heidelberg 2008) pp. 768–783.
[6] R.B. Francisco, M.F.P. Costa, A.M.A.C. Rocha, "Extensions of firefly algorithm for nonsmooth nonconvex constrained optimization problems" in *ICCSA 2016*, Part I, Lect. Notes Comput. Sc. 9786, edited by O. Gervasi et al., (Springer International Publishing Switzerland 2016) pp. 402–417.
[7] X.-S. Yang, "Firefly algorithms for multimodal optimization", in *SAGA 2009*, Lect. Notes Comput. Sc. 5792, edited by O. Watanabe, T. Zeugmann, pp. 169–178 (2009).
[8] M.F.P. Costa, A.M.A.C. Rocha, R.B. Francisco, E.M.G.P. Fernandes, Adv. Oper. Res. **2014**, Article ID 215182, 12 pages (2014).
[9] R. Hooke and T.A. Jeeves, Journal on Associated Computation **8**, 212–229 (1961).
[10] A. Hedar and A. Fahim, Numer. Algebra Control Optim. **1**(1), 99–116 (2011).
[11] M. Schlüter, J.A. Egea, J.R. Banga, Comput. Oper. Res. **36**(7), 2217–2229 (2009)
[12] C.A. Floudas et al., "Handbook of Test Problems in Local and Global Optimization", Nonconvex Optimization and its Applications, vol. 33, Kluwer Academic Publishers, 1999.
[13] G.R. Kocis and I.E. Grossmann, Ind. Eng. Chem. Res. **26**(9), 1869–1880 (1987).