

# Combining Global Tabu Search with Local Search for Solving Systems of Equalities and Inequalities

Gisela C.V. Ramadas<sup>1</sup> and Edite M.G.P. Fernandes<sup>2</sup>

<sup>1</sup> *Department of Mathematics, Polytechnic I. Porto, 4200-072 Porto, Portugal*

<sup>2</sup> *Algoritmi R&D Center, University of Minho, 4710-057 Braga, Portugal*

emails: [gcv@isep.ipp.pt](mailto:gcv@isep.ipp.pt), [emgpf@dps.uminho.pt](mailto:emgpf@dps.uminho.pt)

**Abstract.** This paper aims at providing a combined strategy for solving systems of equalities and inequalities. The combined strategy uses two types of steps: a global search step and a local search step. The global step relies on a tabu search heuristic and the local step uses a deterministic search known as Hooke and Jeeves. The choice of step, at each iteration, is based on the level of reduction of the  $l_2$ -norm of the error function observed in the equivalent system of equations, compared with the previous iteration.

**Keywords:** System of equalities and inequalities; Global optimization; Tabu search; Hooke and Jeeves search

**PACS:** 02.60.Pn

## INTRODUCTION

Global optimization aims at finding the extreme value of a nonconvex function in a certain feasible region. A classical formulation is

$$\min_{x \in \Omega \subset \mathfrak{R}^n} \Psi(x) \quad (1)$$

where  $\Psi(x)$  is the objective function and  $\Omega \equiv [l, u] = \{x \in \mathfrak{R}^n : l_i \leq x_i \leq u_i, i = 1, \dots, n\}$  is the feasible region. This problem is known as bound constrained global optimization problem. An example of a global optimization problem emerges when the global smallest  $l_2$ -norm error function, related to solving the system of equations

$$F(x) = 0,$$

where  $F : \Omega \subset \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ , is required. In this case,  $\Psi(x) \equiv \|F(x)\|_2$  is the objective function. Here, the global minimum, and not just a local minimum of problem (1) is to be found. Classical local search methods have some disadvantages, when compared to global search methods since the final solution is heavily dependent on the starting point of the iterative process and they can be trapped in local minima. In this paper, we are particularly interested in solving the following system of equalities and inequalities:

$$\begin{cases} F_I(x) \leq 0 \\ F_E(x) = 0 \end{cases} \quad (2)$$

where  $I = \{1, \dots, m\}$  and  $E = \{m+1, \dots, n\}$ . Throughout the paper we assume that the vector and its components  $F(x) \equiv (F_I(x), F_E(x))^T = (f_1(x), \dots, f_n(x))^T$ ,  $f_i : \Omega \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ , for any  $i \in \{1, \dots, n\}$ , are continuously differentiable. This type of system appears frequently in box constrained variational inequalities and linear or nonlinear complementarity problems. Classical methods for solving problem (2) use Newton-type methods [2]. Mayne, Polak and Heunis proposed in [7] a method that combines the use of a modified Newton step and a conventional first-order step. Obtaining a solution of (2) by applying a Levenberg-Marquardt algorithm to solving smoothed problems successively, is proposed in [8]. The authors construct a smoothing function to approximate the objective function of the problem. First-order derivatives are then required. Other smoothing-type algorithms have been developed for a direct solving of systems of inequalities. A proposal relies on using a special two-parameter continuously differentiable function  $\theta(\mu, s_i)$  to approximate  $f_i(x) \leq 0$ , for  $i = 1, \dots, m$ , where  $\mu > 0$ , and solving a sequence of systems of the form

$$\Theta(\mu, x, s) \equiv (f_1(x) - s_1, \dots, f_m(x) - s_m, f_{m+1}(x), \dots, f_n(x), \theta(\mu, s_1), \dots, \theta(\mu, s_m))^T = 0$$

for a monotonically decreasing sequence of  $\mu \downarrow 0$ , see [6]. A Newton-type method is then used. A similar smoothing technique is presented in Zhang and Huang's paper [9]. Global convergence is therein guaranteed by a nonmonotone line search technique.

One of the simplest way to solve (2) reformulates the inequalities into equalities, making use of the definition  $x_+ \equiv (\max\{0, x_1\}, \dots, \max\{0, x_n\})^T$ . Thus, system (2) is equivalent to the following system of equations:

$$\begin{cases} F_I(x)_+ = 0 \\ F_E(x) = 0 \end{cases} \Leftrightarrow \begin{cases} \max\{0, f_j(x)\} = 0, j = 1, \dots, m \\ F_E(x) = 0. \end{cases} \quad (3)$$

Since some functions in the equivalent system (3) are nonsmooth, in particular, the corresponding objective function  $\Psi(x)$  of problem (1) is only once differentiable even if all  $f_i(x)$ ,  $i \in \{1, \dots, n\}$ , are twice continuously differentiable, Newton's method cannot be directly applied to solve it. In this paper, we aim at providing a combined derivative-free metaheuristic strategy for solving systems of equalities and inequalities by solving the equivalent system of equations (3), using the function  $\Psi(x)$  as a measure of the progress of the algorithm towards the solution. According to the formulation (1), this means that the fitness of each solution  $x$  is assessed by evaluating the function  $\Psi$  at  $x$ . The combined strategy relies on derivative-free search techniques and uses two types of steps: a global search step and a local search step. The global step relies on a tabu search heuristic [1,4] and the local step performs exploratory and pattern moves as outlined in the Hooke and Jeeves (HJ) method [5]. The choice of step, at each iteration, is based on the sufficient reduction observed in the  $l_2$ -norm error function  $\Psi(x)$ , when compared with the previous iteration.

## COMBINED GLOBAL TABU SEARCH AND LOCAL SEARCH

The Tabu Search (TS), introduced by Cvijović and Klinowski [1] for continuous global optimization, is an iterative process which operates in the following way. The algorithm starts with a randomly generated initial solution,  $x$ , and by applying pre-defined moves in its neighborhood it generates a set  $Y$  of trial solutions. The objective function to be minimized,  $\Psi(x)$ , is evaluated at all solutions in  $Y$ , and the best of all,  $y^*$ , becomes the current solution,  $x \leftarrow y^*$  (even if it is worse than  $x$ ). When uphill moves are accepted, the algorithm avoids to get trapped in a local minimum. The previous procedure is repeated until a given stopping condition is reached. To avoid cycling, since a point already visited may be generated again, a set of points already visited are stored in a list, called Tabu List (TL). The trial solutions in  $Y$  that belong to the TL are eliminated. Usually, the slow convergence of TS is overcome by incorporating a classical local search strategy into the main algorithm. In general, this type of hybridization occurs in the final stage of the iterative process when the solution is in the vicinity of the global solution. An example of such method is presented in [4]. The therein proposed method, called Directed Tabu Search (DTS), uses strategies, like the Adaptive Pattern Search, to direct a tabu search and generate the set  $Y$  of trial solutions. When one of the best obtained trial solutions is sufficiently close to a global minimum, the DTS algorithm uses an intensification search procedure to refine the best solution.

The herein proposed hybridization defines an algorithm that is able to combine two types of steps. The idea is borrowed from the algorithm presented in [3] where a classical Quasi-Newton nonmonotone strategy is used to solve nonlinear systems of equations. The choice of which step is implemented at each iteration depends on the sufficient reduction verified in  $\Psi$  when compared with the previous iteration. ALGORITHM 1 summarizes the main steps of the proposed Combined Global Tabu Search and Local Search (CGTSLS) algorithm. The global search is carried out by a modified version of DTS method, where the HJ local search method is used as the intensification procedure. The local search uses the direct search method HJ [5].

ALGORITHM 1 (CGTSLS algorithm)

Given:  $x^0$ , set  $k=0$ ,  $\eta_0=1$ ,  $flag=1$ ,  $0 < \gamma < 1$ .

Step 1: If  $\Psi(x^k) \leq \eta^*$  or  $k \geq k_{max}$  then stop

Step 2: If  $flag=1$  then

use local HJ search to compute an  $\eta_k$ -approximation  $x^{k+1}$

else

use global DTS search to compute an  $\eta_k$ -approximation  $x^{k+1}$

endif

Step 3: If  $\Psi(x^{k+1}) \leq \gamma \Psi(x^k)$  then set  $flag=1$  else set  $flag=0$  endif

Step 4: Set  $k=k+1$ , set  $\eta_k = \max\{\eta^*, 0.1 \eta_{k-1}\}$  and go to Step 1.

In both global and local searches of the algorithm, it is sufficient to compute an  $\eta_k$ -approximation  $x^{k+1}$  to the optimal solution, i.e., each search terminates when, at iteration  $k$ , the  $\Psi$  value at the best solution found so far is less than  $\eta_k$ . The tolerance defines a decreasing sequence of  $\eta_k$  values (see *Step 4*). Condition  $\Psi(x^{k+1}) \leq \gamma \Psi(x^k)$  in the algorithm defines the sufficient reduction that we aim to observe in  $\Psi$ . The closer  $\gamma$  is to one, the smaller is the required reduction. For the preliminary numerical experiments we define  $\gamma = 0.5$ ,  $\eta^* = 10^{-6}$  (accuracy of the solution) and  $k_{max} = 15n$ . The iterative process (local or global step) in *Step 2* of the algorithm is called inner cycle in opposition to the process indexed by  $k$ , called outer cycle. We use the total number of function evaluations, balanced with the total number of inner and outer iterations to analyze the efficiency of the algorithm.

## NUMERICAL RESULTS AND DISCUSSION

To assess the performance of the proposed CGTSLs algorithm, we selected five problems from the literature [9] and coded them in MATLAB®. We compare our results with those obtained by DTS method, HJ method and the results in [6,9]. When DTS and HJ methods are used separately, a solution within a  $10^{-6}$  accuracy is also required. Both methods are allowed to run for a maximum of  $15n$  iterations as well. The results of the numerical experiments were obtained in a personal computer with an AMD Turion 2.20 GHz processor and 3 GB of memory. Due to the stochastic nature of algorithms based on tabu search, each problem was run 30 times and the best of the 30 solutions is registered. The tested problems are described below, with the initial approximation proposed in [9].

*Example 1:* Consider (2) where  $m=2$ ,  $n=2$  and  $f_1(x) \equiv x_1^2 + x_2^2 - 1 \leq 0$ ,  $f_2(x) \equiv -x_1^2 - x_2^2 + (0.999)^2 \leq 0$ ;  $x^0 = (0, 5)^T$ . Then,  $\Psi(x) = [(\max\{0, x_1^2 + x_2^2 - 1\})^2 + (\max\{0, -x_1^2 - x_2^2 + (0.999)^2\})^2]^{1/2}$ . FIGURE 1 (a) shows  $\Psi(x)$ .

*Example 2:* Consider (2) where  $m=2$ ,  $n=2$  and  $f_1(x) \equiv \sin(x_1) \leq 0$ ,  $f_2(x) \equiv -\cos(x_2) \leq 0$ ;  $x^0 = (0, 0)^T$ . FIGURE 1 (b) illustrates  $\Psi(x)$ .

*Example 3:* Consider (2) where  $m=3$ ,  $n=5$  and  $f_1(x) \equiv x_1 + x_3 - 1.6 \leq 0$ ,  $f_2(x) \equiv 1.333x_2 + x_4 - 3 \leq 0$ ,  $f_3(x) \equiv -x_3 - x_4 + x_5 \leq 0$ ,  $f_4(x) \equiv x_1^2 + x_3^2 - 1.25 = 0$ ,  $f_5(x) \equiv x_2^{1.5} + 1.5x_4 - 3 = 0$ ;  $x^0 = (0.5, 2, 1, 0, 0)^T$ .

*Example 4:* Consider (2) where  $m=1$ ,  $n=3$  and  $f_1(x) \equiv x_1 + x_2 \exp(0.8 x_3) + \exp(1.6) \leq 0$ ,  $f_2(x) \equiv x_1^2 + x_2^2 + x_3^2 - 5.2675 = 0$ ,  $f_3(x) \equiv x_1 + x_2 + x_3 - 0.2605 = 0$ ;  $x^0 = (-1, -1, 1)^T$ .

*Example 5:* Consider (2) where  $m=0$ ,  $n=2$  and  $f_1(x) \equiv x_1 - 0.7\sin(x_1) - 0.2\cos(x_2) = 0$ ,  $f_2(x) \equiv x_2 - 0.7\cos(x_1) + 0.2\sin(x_2) = 0$ ;  $x^0 = (0, 1)^T$ .

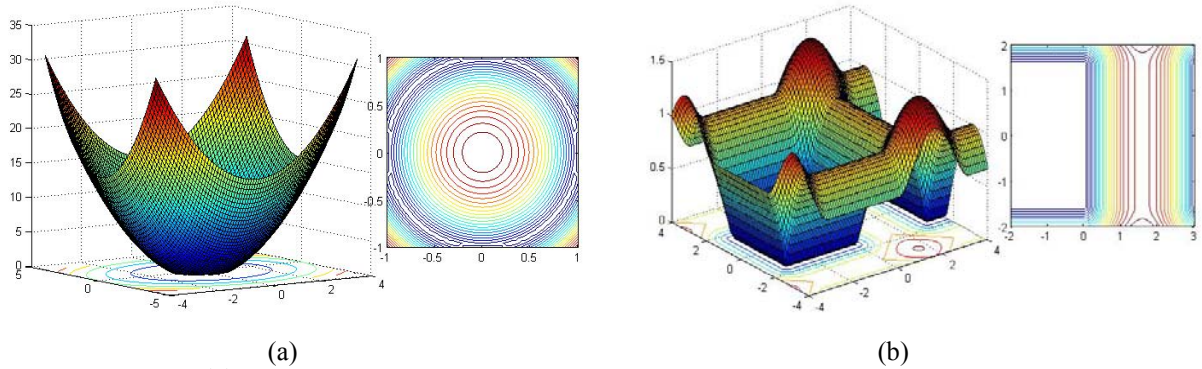


FIGURE 1. Plot of  $\Psi(x)$ : (a) *Example 1*, with contour plot for  $[-1, 1]^2$ ; (b) *Example 2*, with contour plot for  $[-2, 3] \times [-2, 2]$ .

TABLE 1 summarizes the results obtained during the numerical experiments: number of (outer) iterations, “ $k$ ”, total number of inner iterations, “ $k_{inner}$ ”, total number of function evaluations for a function  $F: \mathcal{R}^n \rightarrow \mathcal{R}^n$ , “ $N_{f_{eval}}$ ”, the obtained solution,  $x^*$ , and  $\Psi(x^*)$ . Although DTS, HJ and CGTSLs require larger amounts of function evaluations when compared with the methods presented in [6,9], those are based on a function vector  $F$  of  $n$  components, while the function vector  $H$  in [6,9], used to count the number of function evaluations, has  $1+n+m$  components. The difference between the number of iterations and function evaluations is expected when methods that use derivatives and derivative-free methods are compared all together. Furthermore, Newton-type methods are computationally more expensive since a matrix is evaluated and a linear systems is solved at each iteration. We observe that HJ did not reach the required accuracy in *Example 4* and *Example 5* and DTS in *Example 5*, because they exceeded the pre-specified number of iterations. However, CGTSLs reached a solution with the required

accuracy in all examples. An important issue to be addressed in the future is the reduction in the number of function evaluations. Following ideas in [3,9], a nonmonotone reduction in  $\Psi$  will be implemented in *Step 3* of the ALGORITHM 1.

TABLE 1. Numerical results with an accuracy of  $10^{-6}$ .

Example		$k$	$k_{inner}$	$N_{feval}$	$\Psi(x^*)$	$x^*$
<b>1</b>	in [9] (monotone)					failed
	in [6] (†)	8	-	-	-	(-0.6188, 0.7853)
	results from DTS	30	21 (§)	341	0	(-0.6438, -0.7647)
	results from HJ	26		119	0	(0.6172, 0.7867)
	results from CGTSLS	4	27	271	0	(-0.6227, -0.7812)
<b>2</b>	in [9] (monotone)	4	-	4 (‡)	-	(-0.0152, 0.7206)
	in [6] (†)	7	-	-	-	(-0.0004, 4.7125)
	results from DTS	0	18 (§)	78	0	(0, 0)
	results from HJ	18		74	0	(0, 0)
	results from CGTSLS	1	1	6	0	(0, 0)
<b>3</b>	in [9] (monotone)	5	-	5 (‡)	-	(0.5563, 1.3259, 0.9698, 0.9822, 1.1545)
	results from DTS	75	36 (§)	1712	3.5e-7	(0.5, 2.1678, 1.0, -0.1279, 0)
	results from HJ	36		370	3.5e-7	(0.5, 2.1678, 1.0, -0.1279, 0)
	results from CGTSLS	7	117	1860	3.5e-7	(0.5, 2.1678, 1.0, 0.1279, 0)
<b>4</b>	in [9] (monotone)	6	-	6 (‡)	-	(-0.8299, -0.8663, 1.9566)
	results from DTS	45	37 (§)	815	3.8e-7	(-0.8205, -0.8754, 1.9565)
	results from HJ	45		263	1.2e-6	(-0.7161, -0.9742, 1.9508)
	results from CGTSLS	7	106	889	5.3e-7	(-0.8106, -0.8851, 1.9536)
<b>5</b>	in [9] (monotone)	50	-	207 (‡)	-	(0.5265, 0.5079)
	results from DTS	30	30 (§)	380	2.9e-5	(0.5265, 0.5079)
	results from HJ	30		88	4.0e-4	(0.5258, 0.5078)
	results from CGTSLS	7	79	356	1.9e-7	(0.5265, 0.5079)

(†) – accuracy of  $10^{-3}$ , (‡) - for a function  $H : \mathcal{R}^{1+n+m} \rightarrow \mathcal{R}^{1+n+m}$ , (§) – in the intensification procedure; “-” - not available

## ACKNOWLEDGMENTS

The first author is supported by the CIDEM - Centre for Research & Development in Mechanical Engineering, from Portugal.

## REFERENCES

1. D. Cvijović and J. Klinowski, Taboo search: an approach to the multiple minima problem, *Science* **267**, 664-666 (1996).
2. J.W. Daniel, Newton’s method for nonlinear inequalities, *Numer. Math.*, **21**, 381-387 (1973).
3. A. Friedlander, M.A. Gomes-Ruggiero, D.N. Kozakevich, J.M. Martinez and S.A. Santos, Solving nonlinear systems of equations by means of Quasi-Newton methods with a nonmonotone strategy, *Optim. Meth. Soft.* **8**, 25-51 (1997).
4. A. Hedar and M. Fukushima, Tabu search direct by direct search methods for nonlinear global optimization, *European J. Oper. Research* **170**, 329-349 (2008).
5. R. Hooke and T.A. Jeeves, Direct search solution of numerical and statistical problems, *Journal on Associated Computation*, **8**, 212-229, (1961).
6. Z.-H. Huang, Y. Zhang and W. Wei, A smoothing-type algorithm for solving system of inequalities, *J. Comput. Appl. Math.* **220**, 355-363 (2008).
7. D.Q. Mayne, E. Polak and A.J. Heunis, Solving nonlinear inequalities in a finite number of iterations, *J. Optim. Theory Appl.* **33**, 207-221 (1981).
8. H. X. Yin, Z. H. Huang and L. Qi, The convergence of a Levenberg-Marquardt method for the  $l_2$ -norm solution of nonlinear inequalities, *Numer. Func. Anal. Optim.* **29**, 687-716 (2008).
9. Y. Zhang and Z.-H. Huang, A nonmonotone smoothing-type algorithm for solving the system of equalities and inequalities, *J. Comput. Appl. Math.* **233**, 2312-2321 (2009).