Nonlinear Optimal Control Using Sequential Niching Differential Evolution and Parallel Workers

Yves Matanga [1,*], Yanxia Sun [1], and Zenghui Wang [2]

[1] Department of Electrical and Electronic Engineering, Science University of Johannesburg, Johannesburg, South Africa
[2] Department of Electrical Engineering, University of South Africa, Pretoria, South Africa
*Correspondence: yves.matanga@gmail.com (Y.M.)

***Abstract*—Optimal control is a high-quality and challenging control approach that requires very explorative metaheuristic optimisation techniques to find the most efficient control profile for the performance index function, especially in the case of highly nonlinear dynamic processes. Considering the success of differential evolution in nonlinear optimal control problems, the current research proposes the use of sequential niching differential evolution to boost further the solution accuracy of the solver owing to its globally convergent feature. Also, because sequential niching bans previously discovered solutions, it can propose several competing optimal control profiles relevant for control practitioners. Simulation experiments of the proposed algorithm have been first conducted on IEEE CEC2017/2019 datasets and n-dimensional classical test sets, yielding improved solution accuracy and robust performances on optimal control case studies.**

***Keywords*—sequential niching, differential evolution, nonlinear optimal control**

I. Introduction

Optimal Control (OC) is a control policy that aims to find the control trajectory that drives a dynamic system from one state to another with the least cost possible. This control philosophy has had a transformational impact on industrial processes allowing the use of optimisation theory to improve the quality of plant processes. To solve Optimal Control Problems (OCP), three solution techniques exist in the literature based on either the Pontryagin Maximum principle [1], dynamic programming [2] or direct nonlinear optimisation. While the first two methods can yield very accurate results, they are often not applicable when the complexity and dimensionality of the problem increase. Direct methods, however, discretise the optimal control problems (control and state) into nonlinear programming problems that can be solved by a plethora of solvers [3]. In the optimal control space, differential evolution has been one of the leading solvers used preferably among other meta-heuristic algorithms [4–7]. In order to further improve the performance of the solver, the current study proposes the use of sequential niching differential evolution. Because niching strategies permit the discovery of more than one candidate optimum, its integration with differential evolution will further boost the solution accuracy of the solver and also provide additional competing optimal control profiles relevant for control practitioners. The main contributions of this study are:

1) A sequential niching differential evolution (SNDE) coupled with parallel workers is proposed to discover multiple optima and fine-tune the search with the same amount of function evaluations.
2) An application of SNDE to Nonlinear Optimal Control (NOC) problems is proposed offering improved solution accuracy and several candidate control profiles.
3) A comparative analysis is performed against the state-of-the-art competing metaheuristic optimisation approaches applied to NOC problems.

The structure of this paper is as follows. Section II presents the class optimal control problems considered in the current study. Section III briefly discusses differential evolution and sequential niching. Section IV describes the proposed algorithm. Section V stipulates the simulation experiments used to benchmark the proposed algorithm as well as presents and discusses the results. Section VI provides a conclusion to the research work.

II. Class of Optimal Control Problems

Consider the class of optimal control problems of Bolza type with a fixed time window.

$$\underset{u}{minimize}\quad J(u) = h\left(x\left(t_f\right)\right) + \int_0^{t_f} g(x(t), u(t), t)dt \quad (1)$$

Subject to

$$\dot{x}(t) = f(x(t), u(t), t)$$
$$x(0) = x_0,\ x \in X,\ u \in U$$
$$f \text{ is } nonlinear$$

Manuscript received July 27, 2022; revised August 26, 2022; accepted October 31, 2022; published March 21, 2023.

where $J(u)$ is the performance measure to minimize, $h(.)$, the terminal cost of the system, $g(\cdot)$ the running cost of the system, $f(\cdot)$, the nonlinear dynamic system, t_f, the final time and $u(t)$, the control trajectory that needs estimation. The above OCP can be solved by direct discretisation methods that discretise the control and state system dynamics depending on the case, thus transforming the cost functional into a nonlinear programming problem. Three major discretisation techniques exist: single shooting that discretises the control only, multiple shooting that discretises the control in sub-intervals with knot constraints and the collocation methods that discretise both the control and system dynamics. More insight into these discretisation methods can be found in the work of Rao [8]. Direct single shooting is used in this work which can be described as follows:

$$\min_{u} J(u) = h(x[N]) + \sum_{k=0}^{N-1} \int_{t=kT}^{(k+1)T} g(x(t), u[k], t) \quad (2)$$

Subject to

$$\dot{x}(t) = f(x(t), u(t), t),\ k = 0,1,2, \dots N,$$

$$x(0) = x_0, x \in X, u \in U$$

The current OCP requires numerical integration and an ODE solver in order to solve the optimisation problem numerically. To simplify the problem, the OCP Eq. (2) can be converted to a Mayer-type formulation by the augmentation of a state variable x_{n+1} which represents the integral term:

$$\dot{x_{n+1}} = g(x(t), y(t), t) \quad (3)$$

$$x_{n+1}(0) = 0$$

which yields

$$\min_{u} J(u) = h(x(t_f), t_f) + x_{n+1}(t_f) \quad (4)$$

Subject to

$$\dot{x}(t) = f(x(t), u(t), t),\ k = 0,1,2, \dots N,$$

$$\dot{x_{n+1}} = g(x(t), y(t), t)$$

$$x(0) = x_0, x_{n+1}(0) = 0, x \in X, u \in U$$

III. DIFFERENTIAL EVOLUTION

Several optimal control problems have used differential evolution to find optimal control profiles [4–6]. The work in the literature advises that it has been the meta-heuristic optimisation technique of choice, especially for chemical engineering problems. Differential evolution (DE) is an evolutionary optimisation algorithm developed by Storn [9], Storn and Price [10], that proceeds similarly to the genetic algorithm (GA), however, with differences in its mutation and crossover approach.

Like GA, an initial population of N candidate solutions x_i is generated uniformly across the d-dimensional solution space. Each candidate solution vector is called a genome or chromosome. After initialisation, mutant or donor vectors v_i are created per population member (target vector x_i) typically using Eq. (5):

$$v_i^k = x_j^k + F(x_k^k - x_l^k) \quad (5)$$

whereby a new vector v_i is created by a combination of three independent and randomly selected vectors from the current population different from x_i and F is a scaling factor that varies from 0 to 1. Several variants of the mutation equation exist, and more details can be found in Das *et al.*'s research [11]. Crossover follows the mutation process, upon which trial or offspring vectors u_i are created by combination of the mutant vectors v_i and the target vectors x_i in the current population whereby either component of the target or mutant vector is used in forming u_i:

$$u_{i,j}{}^k = \begin{cases} v_{i,j}^k \text{ if } \mathrm{p} \leq \mathrm{C_r} \\ x_{i,j}, \text{ otherwise} \end{cases} \quad (6)$$

where $j \in [1,2, \dots, d]$, p is a randomly generated number from the crossover probability distribution and C_r is the cross probability ranging between 0 and 1.

Upon crossover, a selection phase follows that determines which N vectors can move to the next generation. In this phase, a comparison is made between the trial vector v_i and the target vector x_i per population member according to their fitness value. The fittest individuals move to the next generation:

$$\mathrm{x_i^{k+1}} = \begin{cases} \mathrm{u_i^k} \text{ if } \mathrm{f}(\mathrm{u_i^k}) \leq \mathrm{f}(\mathrm{x_i^k}) \\ x_i^k, \text{ otherwise} \end{cases} \quad (7)$$

This process continues iteratively until a stopping criterion is met. Algorithm 1 shows the pseudo-code of a typical DE procedure.

Algorithm 1: Typical DE procedure

Let $X_0 = [x^l, x^u], V = [-v_{max}, v_{max}]$
Set pop size N, max_iter K and $k = 0$, $BUB = -\infty$
Randomly generate initial population: $x_j^0 \in X_0$
while $k < K$ and **heuristic stop** not reached **do**
  for j=1: N **do**
    $v_j^k = x_p^k + F(x_q^k - x_r^k), p \neq q \neq r \neq j$
    $v_j^k = bound(v^k, x^l, x^u)$
    u_j^k = cross_over (v_j^k, x_j^k)
    $x_j^{k+1} = \arg\min\left(f(u_j^k), f(x_j^k)\right)$
    (x^*, BUB) = min $(f(x_j^{k+1}), BUB)$
  end
  $k = k + 1$
end
return $(x^*$, BUB)

A. Sequential Niching Metaheuristics

Sequential niching is a niching strategy that proceeds by a repeated search of the problem space during which previously discovered optima are banned in order to permit the discovery of new ones [12]. When a minimum is found in the search domain, the surrounding area, referred to as a niche, is "filled in" and rendered repulsive to other individuals, typically by modification of the objective function as new optima are discovered. Initially, a given meta-heuristic algorithm proceeds, as usual, using the raw objective function. Upon discovery of the first minimum, the objective function values of the individuals in the vicinity of the minimum are modified, and the swarm is restarted. The objective function is modified by the inclusion of a derating function using a recursive formula:

$$\prod_{n+1}(x) = \prod_n(x) G(x, s_n) \tag{8}$$

where $\prod_{n+1}(x)$ is the modified objective function to be used for searching the $n + 1^{th}$ minimum, $\prod_n(x)$ is the objective function used to for searching for the n^{th} minimum, $G(x, s_n)$ is the derating function, and s_n is the n^{th} found minimum. A typical derating function used in the literature is found in the work of Beasley *et al.* [12], Shabbir and Omenzetter [13].

$$G(x, s_n) = \begin{cases} exp\left(log\ m \frac{r-d(x,s_n)}{r}\right) \text{if } d(x, s_n) < r \\ 1,\ otherwise \end{cases} \tag{9}$$

where m is the derating value, r is the niche radius and $d(x, s_n)$ the distance between the current point x and the previously found minimum s_n . Unlike other niching approaches, Sequential Niching Techniques (SNT) present the good feature of approaching global optimality guarantees that offer deterministic global optimisation techniques [14]. It can be hypothesised that by sequentially banning previously discovered optima, all global optima in the problem space can be discovered [15].

IV. Proposed SNDE Algorithm

A. Constant Derating Function

In the current work, a much simpler derating function is used to achieve dynamic tunnelling by flagging the objective function of exploration individuals when within the basin of convergence of previously discovered optima:

$$f(x_i) = \begin{cases} f(x_i) \text{ if } \left\|x_i - x_g\right\| > r \\ +\infty \text{ if } \left\|x_i - x_g\right\| \leq r \end{cases} \tag{10}$$

This process Eq. (10) will ensure that when individuals fall into forbidden region, they will be pulled out of the region after a few iterations due to the process of mutation and evolutionary selection. The radius estimate proposed by Deb is used in this study [16].

$$r = \frac{\sqrt{d}}{2\sqrt[d]{p}} \tag{11}$$

where d is the dimension of the problem, and p, is the number of optima the problem is expected to have. In the same vein as Shabbir and Omenzetter's work [13], only 50% of the radius is used practically.

B. Search Stages

In the current implementation, the sequential niching approach proceeds in three phases: Discovery of promising basins of convergence, Parallel fine-tuning of the basins of convergence and Post optimisation of candidate optima using convex optimisation.

1) Discovering promising basins of convergence

Initially, a N_c number of discoverable basins of convergence is set to determine the maximum number of search restarts. A promising area is deemed obtained if the optimum function does not improve after a number of iterations:

$$\left|f_t^{Best} - f_{t-m}^{Best}\right| \leq \varepsilon \tag{12}$$

Upon discovery of a promising area, a basin of convergence is created around the best individual, and a niche sub-population is formed within the hyper-sphere (or hyper-box) containing all elite individuals within the basin and uniformly distributed additional individuals within the basin up to an N number of individuals overall. Differential evolution is thus repeated to discover new promising areas up to N_{sp}. m is set to 3 in the same vein as other works pertaining to the discovery of niche areas [17].

2) Refining the search quality

Each basin of convergence needs to be exploited to improve the quality of the optimum. Practically, it is worth noting that due to the collaborative nature of exploitation individuals, the exploitation quality of the search is also dependent on the number of exploitation individuals. In order to maintain the same exploitation performance as unimodal differential evolution, the use of parallel workers is proposed. This will ensure that the same number of function evaluations (i.e., maxFev) is used in all workers and each with N exploitation individuals.

3) Post optimisation

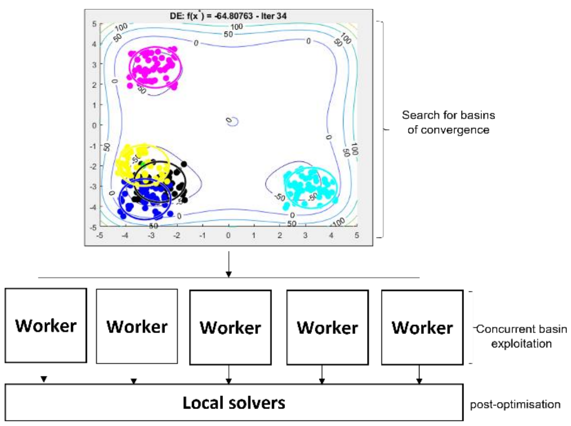

Figure 1. Sequential Niching Differential Evolution (SNDE).

Upon completion of the parallel workers, the best individuals in each sub-populations are used as initial vectors for convex optimisation in order to ensure that the candidate solutions converge to true minima. Considering that continuous NOCPs are typically differentiable, SQP solvers have been used in this study. Fig. 1 and Algorithm 2 present a summary of the proposed algorithm.

Algorithm 2: Sequential niching DE

Set pop size N, max_iter K, maxClusters N_{sp}
Create initial population $x_i \in X_0$
Set cluster_num $n_{sp} = 1$
while $iter < K$ and $n_{sp} < maxClusters$ **do**
  for x in main sub-population
    v = mutation(x)
    v = bound (v, x^l, x^u)
    u = cross_over(v,x)
    if x or u enter a niche_sub_pop
  penalise the fitness (f(x) = $+\infty$ or f(u)=$+\infty$))
    end
    select next-generation individual
  end
  if main sub-pop stalls and $n_{sp} < N_{sp}$
    create hyberbox around gBest
    copy all fittest individuals, n_f in hyberbox
    create niche_sub_pop with n_f individuals
    add $N - n_f$ individuals in the hyberbox
    restart main sub_pop with N new individuals
  end
  $iter = iter + 1$
end
for w=1:N_{sp}
  assign DE_solver($subpop_w$,$hyperbox_w$) to worker w
  set max_iter to $(K - iter)$
  $gBest_w$ = local_search(f,$gBest_w$)
  update (gBest,BUB) = min(BUB,f($gBest_w$))
end
return (x^*=gBest,BUB)

V. Simulation Experiments

A. Multimodal Test Functions for Benchmarking

In order to test the performance of the niching differential evolution, CEC2017/2019 datasets and n-dimensional multimodal classical sets from an extensive literature review have been used, all at the 10^{th} dimension (Table I). Performance profiles have been used to compare optimisation algorithms. The performance profile measures the likelihood of an algorithm performing better than other solvers at a given scaling factor (τ) tested over several test problems. It is a common index used to compare the robustness of optimisation algorithms and can be used with several metrics (i.e., CPU time, solution accuracy, function evaluation, etc.). A performance profile in terms of solution accuracy has been used in this study to assess the quality of the solution of each solver [18]. A solution accuracy measure $m_{(p,s)}$ defines the scaled distance to the optimal function value f^* a solver s obtains on a problem p:

$$m_{(p,s)} = \frac{\widehat{f_{(p,s)}} - f^*}{(f_w - f^*)} \quad (13)$$

where $\widehat{f_{(p,s)}}$ denotes the average estimate of the optimal function by solver s, f_w the worst function value found among the solvers on the problem and f^*, the true optimal function value if available or the best-found optimal function value for problem p among all solvers. The performance profile $\rho_s(\tau)$ of a solver s is thus defined as

$$\rho_s(\tau) = \frac{1}{n_p} size\{p \in P: r_{p,s}\} \quad (14)$$

$$r_{p,s} = \frac{m_{(p,s)}}{min\{m_{(p,s)} \text{ for all } s \in S\}} \quad (15)$$

which thus finds the total number of problems that solver s has a performance ratio $r_{(p,s)}$ with a factor τ. Fig. 2 presents the performance profiles of the genetic algorithm (GA-SQP), differential evolution (DE-SQP), particle swarm optimisation (PSO-SQP) and SQP-convex optimisation against the niching algorithm (SNDE-SQP). Table II gives a summary of each algorithm parameter configuration. All metaheuristic algorithms were tested with a maximum iteration count of 500, a population size of 30 and the average results after fifty optimisation runs were recorded.

TABLE I. Differentiable Multimodal Functions from CEC2017/2019 and from an Extensive Survey on Global Optimization Benchmark Functions (Dimension: $n = 10$) [19–21]

F_n	Functions	range
	CEC 2019	
F_4	Shifted and Rotated Rastrigin	$[-100,100]^D$
F_5	Shifted and Rotated Griewank	$[-100,100]^D$
F_6	Shifted and Rotated Weierstrass	$[-0.5,0.5]^D$
F_8	Shifted and Rotated Expanded Schaffer's F6	$[-100,100]^D$
F_9	Shifted and Rotated Griewank's plus Rosenbrock	$[-100,100]^D$
F_{10}	Shifted and Rotated Ackley	$[-100,100]^D$
	CEC 2017	
F_3	Shifted and Rotated Rosenbrock	$[-100,100]^D$
F_6^*	Shifted and Rotated Schaffer's F7	$[-100,100]^D$
F_9^*	Shifted and Rotated Levy	$[-100,100]^D$
HF_1	Hybrid Function 1	$[0,100]^D$
HF_4	Hybrid Function 4	$[0.1,100]^D$
HF_9	Hybrid Function 9	$[-0.5,0.5]^D$
	Classical n-dimensional	
SF_4	Modified Ackley function	$[-35,35]^D$
SF_7	Alpine 2 Function	$[0.1,100]^D$
SF_{38}	Cosine Mixture Function	$[-10,10]^D$
SF_{43}	Deb 1 Function	$[-100,100]^D$
SF_{44}	Deb 3 Function	$[0.1,100]^D$
SF_{87}	Pathological Function	$[-5,5]^D$
SF_{89}	Pinter Function	$[-100,100]^D$
SF_{110}	Salomon Function	$[-100,100]^D$
SF_{133}	Shubert Function	$[0,5]^D$
SF_{134}	Shubert 3 Function	$[0,5]^D$
SF_{135}	Shubert 4 Function	$[0,5]^D$
SF_{144}	Styblinski-Tang Function	$[-5,5]^D$
SF_{153}	Trigonometric 1 Function	$[-100,100]^D$
SF_{154}	Trigonometric 2 Function	$[-500,500]^D$
SF_{165}	W / Wavy Function	$[-100,100]^D$
SF_{167}	Whitley Function	$[-10,10]^D$
SF_{171}	Xin-She Yang Function	$[-20,20]^D$

TABLE II. PARAMETER CONFIGURATION OF TEST ALGORITHMS

GA	crossover: single point mutation: adaptive selection: stochastic uniform
DE	F = 0.9, c_r = 0.4
PSO	s_r = 2, c_r = 2, w_{max} = 0.9, w_{min} = 0.4
SNDE	SNDE & F = 0.9, c_r = 0.4, maxNiches = 5

B. Optimal Control Case-Studies

1) Isothermal CSTR with complex reactions [22]

$$\begin{aligned} \dot{x_1} &= q_1 - qx_1 - 17.6x_1x_2 - 23x_1x_6u_3 \\ \dot{x_2} &= u_1 - qx_2 - 17.6x_1x_2 - 146x_2x_3 \\ \dot{x_3} &= u_2qx_3 - 73x_2x_3 \\ \dot{x_4} &= -qx_4 + 35.2x_1x_2 - 51.3x_4x_5 \\ \dot{x_5} &= -qx_5 + 219x_2x_3 - 51.3x_4x_5 \\ \dot{x_6} &= -qx_6 + 102.6x_4x_5 - 23x_1x_6u_3 \\ \dot{x_7} &= -qx_7 + 46x_1x_6u_3 \\ \dot{x_8} &= 5.8(qx_1 - q_1) - 3.7u_1 - 4.1u_2 + \\ & q(23x_4 + 11x_5 + 28x_6 + 35x_7) - 5u_3^2 - 0.099 \end{aligned} \tag{16}$$

where $q = q_1 + u_1 + u_2$ and $q_1 = 6$.

The optimal control problem consists of maximising the following cost function:

$$J(u) = x_8(t_f) \tag{17}$$

where the final time is 0.2 h. The initial state is $x(0) = [0.1883\ 0.2507\ 0.0467\ 0.0899\ 0.1804\ 0.1394\ 0.10460]^T$ and the control variables are bounded by

$$0 \le u_1 \le 20, 0 \le u_2 \le 6, 0 \le u_3 \le 4$$

In the current study, the control variable has been discretised into 25 steps.

2) Fed-batch reactor [22]

$$\begin{aligned} \dot{x_1} &= g_1(x_2 - x_1) - \frac{u}{x_5}x_1 \\ \dot{x_2} &= g_2x_3 - \frac{u}{x_5}x_2 \\ \dot{x_3} &= g_3x_3 - \frac{u}{x_5}x_3 \\ \dot{x_4} &= -7.3g_3x_3 + \frac{u}{x_5}(20 - x_4) \\ \dot{x_5} &= u \end{aligned} \tag{18}$$

where

$$g_3 = \frac{21.87x_4}{(x_4+0.4)(x_4+62.5)} \tag{19}$$

$$g_2 = \frac{x_4e^{-5x_4}}{0.1+x_4} \tag{20}$$

$$g_1 = \frac{4.75g_3}{0.12+g_3} \tag{21}$$

The cost function to be maximised is the amount of secreted SUC2-s2 produced at the final time:

$$J(u) = x_1(t_f)x_5(t_f) \tag{22}$$

with $t_f = 15$h, and the control variable bounded as

$$0 \le u \le 10$$

The control variable is discretised into 25 steps as per the original publication.

3) Bifunctional catalyst blend OCP [5, 23]

$$\begin{aligned} \dot{x_1} &= -k_1x_1 \\ \dot{x_2} &= k_1x_1 - (k_2 + k_3)x_2 + k_4x_5 \\ \dot{x_3} &= k_2x_2 \\ \dot{x_4} &= -k_6x_4 + k_5x_5 \\ \dot{x_5} &= k_3x_2 + k_6x_4 - (k_4 + k_5 + k_8 + k_9)x_5 + k_7x_6 \\ & + k_{10}x_7 \\ \dot{x_6} &= k_8x_5 - k_7x_6 \\ \dot{x_7} &= k_9x_5 - k_{10}x_7 \end{aligned} \tag{23}$$

where the control variables are cubic functions:

$$k_i = c_{i1} + c_{i2}u + c_{i3}u^2 + c_{i4}u^3 \tag{24}$$

The values of the coefficients c_{ij} can be found in [23].

The problem consists of maximising the following cost function:

$$J(u) = x_7(t_f)10^3 \tag{25}$$

where $t_f = 2000$. The control variable discretisation step is set to 10.

4) Free-floating robot [24]

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= \frac{(u_1 + u_3)cos(x_5) - (u_2 + u_4)sin(x_5)}{M} \\ \dot{x_3} &= x_4 \\ \dot{x_4} &= \frac{(u_1+u_3)sin(x_5)+(u_2+u_4)cos(x_5)}{M} \\ \dot{x_5} &= x_6 \\ \dot{x_6} &= \frac{(u_1 + u_3)D - (u_2 + u_4)Le}{ln} \end{aligned} \tag{26}$$

The problem consists of minimising the following cost function:

$$J(u) = h\left(x(t_f)\right) + \int_0^5 \frac{1}{2}(u_1^2 + u_2^2 + u_3^2 + u_4^2)dt \tag{27}$$

$$h(x) = (x_1 - 4)^2 + (x_3 - 4)^2 + {x_2}^2$$

$$+x_4{}^2 + x_5{}^2 + x_6{}^2 \quad (28)$$

With $M = 10.0, D = 5.0, Le = 5.0$ and $ln = 12.0$, $|u_i| \le 5$ The control variable discretisation step is set to 10. Ten optimisation runs were performed for each solver for a maximum iteration count of 300, and the average solution accuracy for each solver is presented in Table III.

C. Results and Discussion

1) On the performance of sequential niching DE on benchmarking test functions

The performance profiles in Fig. 2 show that sequential niching DE is the best performing solver among the three traditional metaheuristics obtaining the best fitness value for 48% of all test problems while DE, GA, PSO and SQP obtain respectively 24%, 28%, 7% and 0%. GA obtained the best performance among the traditional metaheuristic algorithms, while DE was the most robust, with the highest likelihood of obtaining the lowest fitness values. It is worth noting that the choice of DE in the current study was preferential owing to research findings in optimal control problems, particularly for chemical engineering problems. These metaheuristics algorithms can be used interchangeably depending on the context [13, 25, 26].

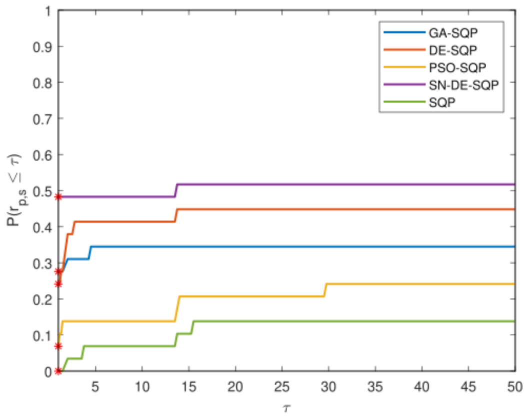

Figure 2. Performance profiles of competing algorithms: PSO, DE, GA, SQP, SNDE.

2) On the performance of sequential niching DE on optimal control case studies

The results in Table III show that sequential niching differential evolution is the most robust among alternatives obtaining the best average results across all NOC problems, which exemplifies the benefit of the proposed algorithm. Although the framework has been conceptualised for three decades now [12], the current study has proposed a practical implementation that does not require a complex derating function as well as propose a parallel framework of implementation that can shorten the search time and easily scale up when parallel computation resources are available. In order to enhance the optimisation results, sensitivity analysis can be performed to select the best DE parameters (i.e., F, c_r). From a computational point of view, the use of merging operators can be investigated to potentially merge niches that are likely to converge to the same basin and save search iterations for the discovery of other promising niches [27]. Also, efficient adaptive approaches to estimate the niche radius can be investigated.

TABLE III. SOLUTION ACCURACY OF NONLINEAR CONTROL PROBLEMS (NOCP) PER SOLVER

	NOCP1	NOCP2	NOCP3	NOCP4
SNDE	**20.0948**	**35.5493**	**10.0942**	**13.0390**
PSO	20.0842	5.1846	9.8010	**13.0390**
GA	20.0556	**32.5493**	9.9317	**13.0390**
DE	20.0870	32.4768	**10.0942**	**13.0390**
SQP	20.0855	0.0000	9.6473	**13.0390**

VI. CONCLUSION

The current study focused on improving the solution accuracy of differential evolution, which is often used for the optimal control of nonlinear processes. It has proceeded with the use of sequential niching to discover other promising regions, followed by a concurrent exploitation methodology using all available individuals owing to parallel workers. This approach not only improves the solution accuracy of the cost functional, but it also permits the obtention of alternate control profiles relevant to control practitioners. Future work will consist of estimating the radius of the basin of convergence adaptively and will make use of merging operators to join basins of convergence that could converge to the same optima and thus save computational resources or permit the search for other potential promising areas. Finally, sensitivity analysis of DE parameters will also be investigated.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Yves Matanga reviewed the literature, designed the research methodology, collected the results, and compiled the manuscript. Yanxia Sun supervised the research work and provided guidance. Zenghui Wang supervised the research work and provided guidance and funding. All authors had approved the final version.

FUNDING

This research was supported by South African National Research Foundation Grants (Nos. 132159, 114911, 137951 and 132797) and Tertiary Education Support Programme (TESP) of South African ESKOM.

REFERENCES

[1] D. E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publications, 2012, pp. 329–413.

[2] Y. Cai and K. L. Judd, "Advances in numerical dynamic programming and new applications," in *Handbook of Computational Economics*, vol. 3, 2014, pp. 479–516.

[3] E. Nolasco, V. S. Vassiliadis, W. Kähm, S. D. Adloor, R. Al-Ismaili, R. Conejeros, T. Espaas, N. Gangadharan, V. Mappas, F. Scott, *et al.*, "Optimal control in chemical engineering: Past, present and future," *Computers & Chemical Engineering*, vol. 155, 107528, 2021.

[4] R. Angira and A. Santosh, "Optimisation of dynamic systems: A trigonometric differential evolution approach," *Computers & Chemical Engineering*, vol. 31, pp. 1055–1063, 2007.
[5] I. L. Cruz, L. G. van Willigenburg, and G. van Straten, "Efficient differential evolution algorithms for multimodal optimal control problems," *Applied Soft Computing*, vol. 3, pp. 97–122, 2003.
[6] K. B. Cantún-Avila, D. González-Sánchez, S. Díaz-Infante, and F. Peñuñuri, "Optimising functionals using differential evolution," *Engineering Applications of Artificial Intelligence*, vol. 97, 104086, 2021.
[7] F. S. Lobato, V. Steffen Jr., and A. J. S. Neto, "Solution of singular optimal control problems using the improved differential evolution algorithm," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 1, 2011.
[8] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in Astronautical Sciences*, 2010.
[9] R. Storn, "On the usage of differential evolution for function optimisation," in *Proc. the North American Fuzzy Information Processing*, 1996.
[10] R. Storn and K. Price, "Minimising the real functions of the ICEC'96 contest by differential evolution," in *Proc. IEEE International Conference on Evolutionary Computation*, 1996.
[11] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
[12] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimisation," *Evolutionary Computation*, vol. 1, pp. 101–125, 1993.
[13] F. Shabbir and P. Omenzetter, "Particle swarm optimisation with sequential niche technique for dynamic finite element model updating," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, pp. 359–375, 2015.
[14] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, pp. 79–102, 2016.
[15] J. Barhen, V. Protopopescu, and D. Reister, "TRUST: A deterministic algorithm for global optimisation," *Science*, vol. 276, pp. 1094–1097, 1997.
[16] K. Deb, "Genetic algorithms in multimodal function optimisation," Doctoral dissertation, Clearinghouse for Genetic Algorithms, Dept. of Engineering Mechanics, University of Alabama, Tuscaloosa, USA, 1989.
[17] Q. Liu, S. Du, B. J. van Wyk, and Y. Sun, "Niching particle swarm optimisation based on Euclidean distance and hierarchical clustering for multimodal optimisation," *Nonlinear Dynamics*, vol. 99, pp. 2459–2477, 2020.
[18] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimisation test problems," *Journal of Global Optimisation*, vol. 31, pp. 635–672, 2005.
[19] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimisation," Technical Report, National University of Defense Technology, Changsha, China and Kyungpook National University, Daegu, South Korea, and Nanyang Technological University, Singapore, 2017.
[20] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimisation," Technical Report, Nanyang Technological University, 2018.
[21] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, pp. 150–194, 2013.
[22] R. Luus, "Parametrisation in nonlinear optimal control problems," *Optimisation*, vol. 55, pp. 65–89, 2006.
[23] W. R. Esposito and C. A. Floudas, "Deterministic global optimisation in nonlinear optimal control problems," *Journal of Global Optimisation*, vol. 17, pp. 97–126, 2000.
[24] B. C. Fabien, "A java application for the solution of optimal control problems," *Stevens Way, Box*, 352600, 1998.
[25] Y. Matanga, Y. Sun, and Z. Wang, "Globally convergent fractional order PID tuning for AVR systems using sequentially niching metaheuristics," in *Proc. 7th International Conference on Robotics and Automation Engineering*, Singapore, 2022.
[26] F. Shabbir and P. Omenzetter, "Model updating using genetic algorithms with sequential niche technique," *Engineering Structures*, vol. 120, pp. 166–182, 2016.
[27] R. Brits, A. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimiser," in *Proc. the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, Singapore, 2002.

Yves Matanga received the M.Sc. degree in electrical and electronic systems from ESIEE, France and the M.Tech. degree in electrical engineering from the Tshwane University of Technology, Pretoria, South Africa, in 2018. He holds a BTech degree in electrical engineering with specialisation in electronics obtained in 2014 at the Tshwane University of Technology. He is currently pursuing a PhD degree in electrical and electronic engineering at the University of Johannesburg, South Africa working on convergence improvement of deterministic and globally convergent optimisation algorithms with applications to control systems. His research interests include dynamic systems and control, optimisation, artificial intelligence, signal, and image processing.

Yanxia Sun received the joint D.Tech. degree in electrical engineering from the Tshwane University of Technology, South Africa, and the PhD degree in computer science from University Paris-EST, France, in 2012. She is currently working as a professor with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. She has more than 15 years of teaching and research experience. She has lectured five courses in universities. She has supervised or co-supervised six postgraduate projects to completion. She has published more than 110 articles, including 35 ISI master-indexed journal papers. She is the investigator or co-investigator for six research projects. She is a member of the South African Young Academy of Science (SAYAS). Her research interests include renewable energy, evolutionary optimisation, neural networks, nonlinear dynamics, and control systems.

Zenghui Wang received the B.Eng. degree in automation from the Naval Aviation Engineering Academy, China, in 2002, and the PhD degree in control theory and control engineering from Nankai University, China, in 2007. He is currently a professor at the Department of Electrical and Mining Engineering, University of South Africa (UNISA), South Africa. His research interests are industry 4.0, control theory and control engineering, engineering optimisation, image/video processing, artificial intelligence, and chaos.