



An evolutionary approach to solve a system of multiple interrelated agent problems

Tohid Erfani^{a,*}, Rasool Erfani^b^a University College London, Department of Civil, Environmental and Geomatic Engineering, London, UK^b Manchester Metropolitan University, Department of Mechanical Engineering, Manchester, UK

ARTICLE INFO

Article history:

Received 5 August 2014

Received in revised form 15 May 2015

Accepted 13 July 2015

Available online 13 August 2015

Keywords:

Agent based problems

Complementarity conditions

Evolutionary algorithm

Parallel search

ABSTRACT

Deterministic approaches to simultaneously solve different interrelated optimisation problems lead to a general class of nonlinear complementarity problem (NCP). Due to differentiability and convexity requirements of the problems, sophisticated algorithms are introduced in literature. This paper develops an evolutionary algorithm to solve the NCPs. The proposed approach is a parallel search in which multiple populations representing different agents evolve simultaneously whilst in contact with each other. In this context, each agent autonomously solves its optimisation programme while sharing its decisions with the neighbouring agents and, hence, it affects their actions. The framework is applied to an environmental and an aerospace application where the obtained results are compared with those found in literature. The convergence and scalability of the approach is tested and its search algorithm performance is analysed. Results encourage the application of such an evolutionary based algorithm for complementarity problems and future work should investigate its development as well as its performance improvements.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction and background

We study a class of problems in which solutions to n interrelated optimisation problems are simultaneously required. In this context, each agent solves one optimisation problem and seeks its own optimal strategies while interacting with the others. That is, each agent's problem is formed to find the best response to the decision of the interacting agents. More precisely, given a function $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ representing n agents, we find $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}_+^n$ by simultaneously solving the following n optimisation problems, P_i :

$$\begin{aligned} & \text{maximise}_{x_i} F_i(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in X_i. \end{aligned} \quad (P_i) \quad (1)$$

where each agent i controls vector $x_i \in \mathbb{R}^{v_i}$ ($\sum_{i=1}^n v_i = n$) to optimise the objective function F_i subject to the constraints set X_i containing $\mathbf{x} \in \mathbb{R}_+^n$. The interrelation is explained as the objective function and the constraints in P_i depend on other agents' decisions which

explains that the solution to one agent's problem affects that of the others.

This representation can be used to model each agent's personal interests. Given that aggregated optimisation framework seeks collective optimality, which does not necessarily comply with each agent's interest, it cannot be used to solve the aforementioned problem [1]. Deterministic approaches to simultaneously solve the above n problems lead to a system of nonlinear equations, which are formed by each agent's Karush–Kuhn–Tucker (KKT) formulation [1]. Under known conditions, the KKT satisfies necessary and sufficient condition for optimality of each problem i . However, employing the KKT depends on the properties of the function F , namely a smooth continuous function. If so, deterministic techniques generally exercise pivoting algorithm, interior point based approach, and generalised Newton method to find a feasible solution (e.g. look at [2,3]). Complementarity conditions introduced by using KKT provide insight into modelling techniques leading to a general class of nonlinear complementarity problems (NCP) used in economics [4], traffic modelling [5], robotics [6], fluid dynamics [7], and energy planning [8]. However, the non-differentiability and non-convexity of the function F make the deterministic approaches less efficient. Further, potential multimodality requires a good starting point to ensure convergence. To avoid such issues, application of global gradient-free approaches are encouraged to solve

* Corresponding author. Tel.: +44 0207 679 0673.

E-mail addresses: t.erfani@ucl.ac.uk (T. Erfani), r.erfani@mmu.ac.uk (R. Erfani).

the resulting NCPs. Example of these works are decomposition based method coupled with genetic algorithm [9], orthogonal genetic algorithm within successive reformulated optimisation problem for NCP constraint satisfaction [10], and evolutionary based approach using Nikaido Isoda function mapping [11]. General agent based approaches have also been presented in literature to solve complex interrelated optimisation problems. Ouelhadj and Petrovic [12] report the application of agent based modelling in complex scheduling in dynamic environments. Cowling et al. [13] and Lau et al. [14] simulate the supply chain system with multiple independent and autonomous agents for contractor selection. Sauvageau and Frayret [15] use agent approaches to represent the paper pulp recycling supply chain. Moreover, in transportation, scheduling and planning, agent based approaches are successfully employed for optimisation of train coupling systems [16,17], for routing decision making based on local information in dynamic environment [18], for solving the dynamic scheduling problem of a distributed project with self-interested participants [19], for dealing with energy systems planning and forecasting [20] and for land usage and environmental planning [21]. Combination of operational research and agent modelling has also been implemented for developing a decision support system in supply chain coordination context [22–24].

In this paper, we present an alternative approach to solve the above n problems without directly employing the KKT and NCP formulations. We employ an evolutionary algorithm which solves n interrelated optimisation problems in parallel. Section 2 explains the sharing strategies and Section 3 develops the algorithm and gives a demonstration on an illustrative example. Section 4 applies the proposed approach on two problems in environmental water pollution and fluid dynamics and discusses the results. The scalability of the method to higher dimension as well as its search algorithm performance are investigated in Section 5. Section 6 concludes the paper with extra discussion for future work.

2. Preliminaries of proposed formulation

Multiple agents communicate their decision to the others and hence their problems are interrelated. This is due to the fact that each agent deals with its own problem as well as reacts to the decisions of the others within an environment (each agent finds its best response to the decisions of the others). Given this, we propose to solve such interrelated agent problems using the idea of parallel genetic algorithm (GA) [25]. Parallel GA combines the hardware speed of parallel processors and the software speed of intelligent parallel searching [26]. The idea in parallel GA is to divide the problem's search population over multiple processors, typically for performance reasons which determines multiple sub-populations with information exchange between them [27,28]. This implementation promises a substantial performance achievement and therefore leads to extensive attempts to design and improve competitive distributed hardware for effective population communication and migration as well as fitness evaluations speeding up [29–31]. Most of the algorithms in literature run identical GA in parallel with one run per processor while they differ in the linkage of the populations and the information sharing (e.g. look at [32–34]). In this paper, instead of dividing the population over multiple processors, we borrow this idea and dedicate to each agent a population to solve its problem while interacting with the others.

More formally, define an agent problem i with P_i , $i=1, \dots, n$, and let X_i be a vector containing the decision variables of agent i used in problem P_i and let \mathbf{x}_{-i} be a vector containing the decision variables of all agents involved in problem P_i excluding that of the agent i . Further, define \mathbf{P} the problem formed by all P_i s. We

use parallel GA and the idea of co-evolution [35] to solve \mathbf{P} with an extension that each (sub-)problem P_i has its own objective function. Since there is interconnection between agents' problems, we solve each problem whilst it communicates with the other problems by sharing information. This concept is used in [36] to gain faster convergence to Pareto solution in multiobjective optimisation problem. Formally, the search algorithm is described by n different search trajectories performing in parallel through the following linkage H_i :

$$x_i^{t+1} = H_i(\bar{\mathbf{x}}_{-i}^t, x_i^t, P_i),$$

where H shows the interconnection between the agents. H acts as a synchronisation linkage for agent i to optimise problem P_i given the decisions of other interacting agents in its neighbourhood remain fixed shown by $\bar{\mathbf{x}}_{-i}^t$. H describes that X_i value is updated by a search on problem P_i at generation t linking decisions X_i and $\bar{\mathbf{x}}_{-i}$. In fact, in problem P_i , x_i is the reaction of agent i to the other agents' decisions given by $\bar{\mathbf{x}}_{-i}$. Due to problem P_i , each agent knows its own problem components and hence by communicating with other neighbouring agents through H , it has local activity for exploring the search space. Following Alba and Troya classification [37] and due to linkage H , our approach resembles to a fine grained topology of parallel GAs as each agent communicates with its neighbours while solving its own problem. Next, we give the details of the algorithm to solve the agents problems followed by an illustrative example.

3. Parallel search algorithm

Algorithm 1 shows the step by step procedure of the proposed method when each agent involves in maximisation problem. At the beginning, the number of agents n , population size m , maximum number of generation $MaxGen$ and convergence tolerance ϵ are specified.

Each agent i has a devoted search trajectory formed by a population of size m (Line 1). pop_i is a $m \times ne_i$ matrix and is populated randomly. Agent i and its neighbours make a set of ne_i agents (Line 2). Each individual p_k in population of agent i is composed of the decision variables of agent i and the decision variables of its neighbours being fixed, and therefore, $p_k = (x_i, \bar{\mathbf{x}}_{-i})$ has the size of ne_i . In other words, ne_i equals the number of neighbouring agents affecting the decision of agent i plus one. For agents to determine their best response to the decisions of their neighbours, all individuals p_k in population i undergoes a reproduction in each generation t of parallel searches (Line 8). pop_i is sorted in increasing order based on the objective value and pop_i^* , the best individual in pop_i is determined (Line 11 and 12). At the end of each generation t , the neighbouring agents ($j \in neighbours_i$) share their best individuals to form the updated population for next generation $t+1$. pop_i^* migrates to the population of the neighbours and remain fixed for the next generation (Line 13). This makes each agent at the end of each generation to be informed of the decisions of its neighbouring agents involved in its own problem. An example of population sharing scheme is illustrated in Fig. 1 with $ne_i=3$ and three agents dealing with problem P_i optimising for X_i . Due to n different search trajectories, the algorithm allows each agent to search for its best response to its neighbours' decisions by relying only on locally available information received from them through H . This procedure leads to the evolution of separate populations over successive generations.

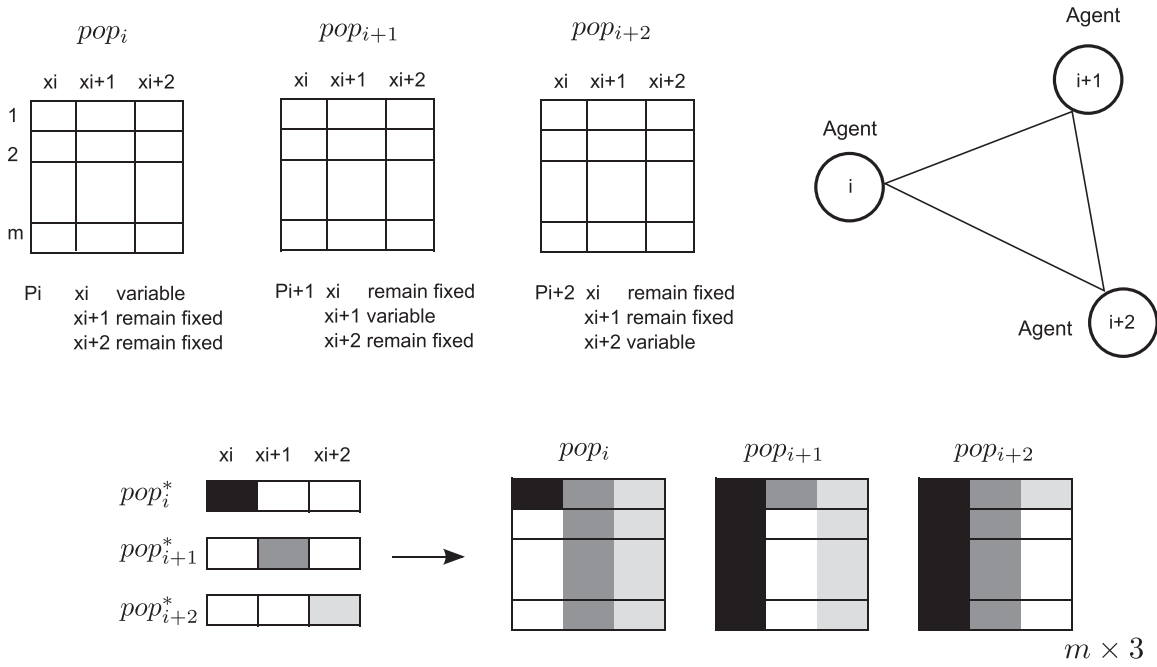


Fig. 1. Exchanging the best individual values within the the neighbouring populations at the end of each generation for $ne_i = 3$. x_i^* is fixed in pop_{i+1} and pop_{i+2} , x_{i+1}^* is fixed in pop_i and pop_{i+2} , and finally x_{i+2}^* is fixed in pop_i and pop_{i+1} .

Algorithm 1. Parallel search algorithm.

```

1  Set  $MaxGen$   $\epsilon$ ,  $n$  and  $m$ ;
2  Define the set  $neighbours_i$  and set  $ne_i = |neighbours_i| + 1$ ;
3  Randomly initialise  $n$  populations of size  $m$  ( $pop_i$ );
4  while Not  $MaxGen$  and  $\eta > \epsilon$  do
5      for  $i = 1$  to  $n$  do
6          for  $k = 1$  to  $m$  do
7              Randomly pick  $p_{s1} \neq p_{s2} \neq p_{s3} \neq p_k$  from  $pop_i$ ;
8               $p_b \leftarrow$  reproduction ( $p_{s1}, p_{s2}, p_{s3}$ );
9              if  $f_i(p_b) \geq f_i(p_k)$  then
10                  $p_k \leftarrow p_b$ 
11              $pop_i$  is sorted in increasing order based on  $f_i$  value;
12              $pop_i^* \leftarrow$  The best individual in  $pop_i$ ;
13          $\forall i, j = 1, \dots, n, i \neq j, pop_i \leftarrow pop_j^* \wedge j \in neighbours_i$ ;
14         calculate  $\eta$ ;
```

3.1. Termination check, constraint handling and reproduction stage

For stopping criterion, we follow the work of Sinha et al. [38] and the algorithm uses a variance-based convergence indicator defined by,

$$\eta = \sum \frac{\sigma_{V_{ic}}^2}{\sigma_{V_{io}}^2},$$

where $\sigma_{V_{ic}}^2$ and $\sigma_{V_{io}}^2$ denotes the variances of agents i 's variables in the current (c) and original (o) population, respectively. The value of η is restricted between 0 and 1 and upon convergence it is expected that this value converges to zero. Therefore, if η is less than ϵ then the algorithm terminates, otherwise, the algorithm continues until $MaxGen$ is reached. We also take a track of the mean of population of each agent's variables in each generation to further analyse the convergence.

Generation	pop ₁		pop ₂	
	x ₁	x ₂	x ₁	x ₂
1	22.938	50.259	22.938	50.259
	16.583	19.664	16.583	19.664
	41.448	34.853	41.448	34.853
2	16.583	50.259	22.938	0.000
	20.802	50.259	22.938	9.722
	22.938	50.259	22.938	10.854
3	12.312	0.000	16.583	6.908
	12.529	0.000	16.583	1.333
	14.872	0.000	16.583	9.709
⋮	⋮	⋮	⋮	⋮
23	7.000	10.000	7.000	10.000
	7.000	10.000	7.000	10.000
	7.000	10.000	7.000	10.000
24	7.000	10.000	7.000	10.000
	7.000	10.000	7.000	10.000
	7.000	10.000	7.000	10.000

Fig. 2. The algorithm procedure in evolution of two separate populations in converging to the best strategy for each agent in production competition. The arrows explain how, in each generation, the best individual of pop₁ remains fixed in pop₂.

The constraint domination technique is adopted for constraint handling [39]. The procedure ranks the feasible individuals based on their objective values while the infeasible individuals are ranked only based on their extent of constraint violation.

Each agent determines its best decision as the response to the decisions of its neighbours by applying a search on its population in line 7 and 8. In this paper, differential evolution (DE) [40] is adopted for reproduction as it provides promising results on many numerical test problems [41]. Nevertheless, other search algorithm can be exploited. In DE, corresponding to each p_k, three individuals p_{s1}, p_{s2} and p_{s3} are randomly chosen from population pop_i and new vector p_c is created by adding the weighted difference of p_{s2} and p_{s3} to the p_{s1} given by,

$$p_c = p_{s1} + MR(p_{s2} - p_{s3}),$$

where MR is the mutation rate. p_c is accepted as a new vector p_b if the following is satisfied

$$p_b = \begin{cases} p_c & \text{if } rand(0, 1) \leq CR \\ p_k & \text{o.w.} \end{cases} \quad (2)$$

CR is the probability of crossover and rand is pseudo random number between 0 and 1. If vector p_c violates the bound constraints, it gets the value half way between the p_k and the bound violated.

Before we test the algorithm’s application, we explain its implementation via a simple two agents problem as follows.

3.2. Illustrative example

To illustrate the algorithm procedure, consider two agents competing in producing a product. The benefit of each agent is given by

$$f(x_i) = (30 - x_1 - x_2)x_i - c_i x_i,$$

where c₁ = 6 and c₂ = 3 represents the unit production cost incurred to agent i. The benefit of one agent depends on the amount of production of the other. Therefore, there is an interconnection between these two and therefore, ne₁ = ne₂ = 2. We set n = 2, m = 10, ε = 10⁻⁵ and MaxGen = 100. As can be seen in Fig. 2, after initialising the

Table 1
Parameters for the water pollution problem (taken from [42]).

Firm _i	c _{1i}	c _{2i}	e _i	δ _{1i}	δ _{2i}
1	0.10	0.01	0.50	6.5	4.583
2	0.12	0.05	0.25	5.0	6.250
3	0.15	0.01	0.75	5.5	3.750

population for the first generation, the best individual in pop₁ represented as pop₁^{*} = 22.938 is fixed in the next pop₂ generation and pop₂^{*} = 50.259 is fixed in pop₁ (shown by grey and black colours, respectively). The population undergoes reproduction stage using DE and in second generation, pop₁^{*} = 16.583 is fixed in pop₂ and pop₂^{*} = 0 is fixed in pop₁ for third generation. The algorithm runs and the best of each population is fixed in the others in an iterative basis. This simulates a procedure in which each agent is reacting to the best action of its competitor. The competition through generation designs a learning environment which continues till both agents compromise on a solution. Fig. 2 only shows the first three individuals in each population for simplicity of presentation. The algorithm converges after 23 generations with η = 10⁻⁶.

In the next section, we investigate the algorithm performance for two multidisciplinary problems.

4. Experimental setup and results

We solve two problems related to environmental water pollution and fluid dynamics. The first problem involves three agents competing for water abstraction while the second one approximates the pattern of the velocity profile over an aerofoil. The first problem is taken from [42] where a deterministic algorithm is devised for obtaining the solution. The second problem, employs finite difference method for formulating the agent problems, and, to our best knowledge, has not been solved from multi-agent perspective before in literature.

In both problems, the population size for each agent is m = 50, the maximum number of generation is MaxGen = 100 and following Price et al. [40], MR and CR values used within DE equal 0.7 and 0.5, respectively. The convergence is assumed with tolerance of less than ε = 10⁻⁵.

4.1. Environmental water pollution

In this problem, three firms i = 1, 2, 3, are located along a river, abstracting water to produce paper pulp at a chosen level X_i. Due to paper production, the firms also produce one pollutant, e_iX_i, i = 1, 2, 3, with e_i being the emission coefficient of firm i (Table 1). The pollution is expelled into the river and reaches two monitoring stations l located along the river where local authority sets the maximum pollutant concentration levels K_l. The following constraints at each location l is then defined as:

$$q_l(\mathbf{x}) = \sum_{i=1}^3 \delta_{il} e_i x_i \leq K_l, \quad l = 1, 2,$$

where δ_{il} is the decay-and-transportation coefficient from firm i to location l (Table 1). Each firm is engaged in net profit maximisation equivalent to the following optimisation problems for firm_i, i = 1, 2, 3:

$$\begin{aligned} \max_{x_i} \quad & f_i = R_i - C_i \\ \text{subject to} \quad & q_l(\mathbf{x}) \leq K_l \quad l = 1, 2. \end{aligned} \quad (3)$$

where the revenue and cost functions for each firm are

$$R_i = (d_1 - d_2(x_1 + x_2 + x_3))x_i,$$

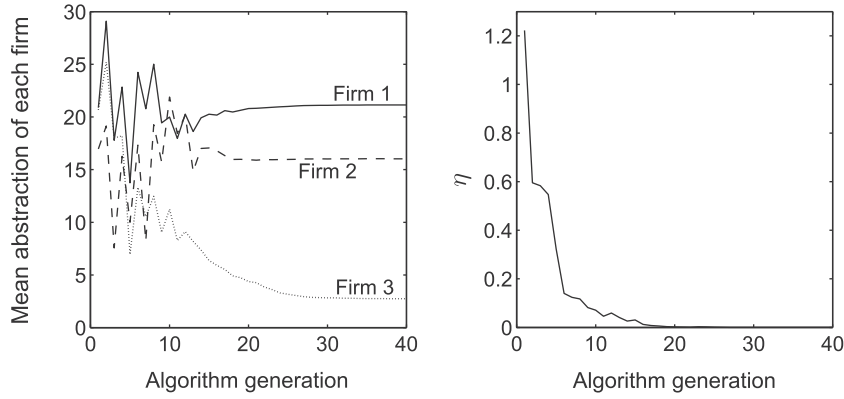


Fig. 3. The average amount of water abstraction for each firm in each algorithm generation.

Table 2

Results of each firm water abstraction: Proposed method and the deterministic approach in [42].

Method	Firm ₁	Firm ₂	Firm ₃
Deterministic approach by [42]	21.14	16.03	2.73
Parallel approach (mean)	21.14	16.03	2.73

$$C_i = (c_{1i} + c_{2i}x_i)x_i \quad i = 1, 2, 3,$$

respectively, with economic constants $d_1 = 3.0$ and $d_2 = 0.01$. The cost function coefficients c_1 and c_2 are given in Table 1 and $K_l = 100$, $l = 1, 2$ for this study.

The objective functions represents the gross benefit of each firm in participating in paper pulp production. The complementarity condition essentially state that at an equilibrium solution, marginal revenue must be equal to marginal cost for each firm abstracting a positive quantity of water, and that marginal revenue should not exceed marginal cost for firms which prefer not to be in business.

Due to R_i function, the decision of each firm depends on the other two and hence, $ne_i = 3$ for $i = 1, 2, 3$. The pollution constraints q_l is handled using domination technique; (a) the feasible solutions are always preferred, (b) between any two feasible solutions, the one with higher gross benefit is preferred, and (c) between any two infeasible solutions, the one which less violates the pollutant constraint is favoured.

The parallel search is constructed for three agents and the results of its application is shown in Table 2. The solution is consistent with the one obtained using deterministic approach reported by Krawczyk [42]. To show the convergence rate of the algorithm, we provide the mean of water abstraction for each agent in Fig. 3 as well as the η value against the algorithm generation. The figures illustrate the firms competition over water abstractions during the population evolution. At the initial generations, since one firm gain leads to others loss, there is an oscillating behaviour to reach equilibrium further down the generation. The convergence is quite fast and as can be seen in the figure, after only 10 generation the firms compromise on a solution while the algorithm tunes the solution afterwards. The approach can be employed for bigger problems where large number of firms compete for water abstraction with different institutional and quality constraints.

To further demonstrate the algorithm, we alter the production cost C_i to the non-differentiable and non-convex fixed-cost function,

$$C_i = \begin{cases} c_{1i} + c_{2i}x_i & x_i > 0 \\ 0 & x_i = 0. \end{cases} \quad (4)$$

and solve problem 2 with no constraint. The introduction of the above function reduces the applicability of the deterministic approaches with convexity and differentiability assumptions [4]. Employing the proposed method, the algorithm converges to $\mathbf{x} = (75.7, 71.7, 75.7)$ after 90 generations.

4.2. Profile of flow velocity on NACA aerofoil

In this problem we study the behaviour of the algorithm when it is applied to a fluid dynamic application. We solve the laminar flow around the leading edge of a symmetrical aerofoil, NACA 0015, with zero angle of attack, chord length $c = 32$ cm and $t_{\max}/2 = 2.4$ cm. The characteristics of this aerofoil are documented by Abbott and Doenhoff [43] and it exhibits well-behaved laminar flow around leading edge at low angles of attack. We consider the flow to be two-dimensional incompressible fluid defining a constant density of the fluid. Near low angle of attack, which results in zero lift, a steady flow past the aerofoil has the same pattern on upper and lower surfaces, except for a narrow boundary layer region close to the surface where the flow is retarded by friction. The resultant simplified Navier-Stokes equations for a steady, two-dimensional and incompressible flow can be found in [44]. In practice, by applying the potential flow theory, we can find the velocity profiles of the outer flow field of an aerofoil [44]. The subject of current study is to, from the multi-agent perspective, approximate the pattern of such velocity profiles passing from the leading edge of the aerofoil ignoring the boundary layer effect. Due to impermeability condition of the plate surface, velocity on the surface is zero and the gradient of velocity equals zero anywhere else in y -direction. These conditions give rise to the following complementarity problem:

$$\begin{aligned} \nabla x(x - s) &= 0, \\ x \geq s &\geq 0 \end{aligned} \quad (5)$$

where x is velocity profile and s is the leading edge of the NACA 0015 aerofoil. To obtain the pattern of the velocity profile x using the proposed parallel search, we approximate x in five different locations ($a-e$) on the surface s . Employing finite difference method on a mesh with step size H , gradient function approximation, ∇ , leads to the following set of equations for each location

$$\frac{x_{i-1} - x_i}{h}(x_i - s(y_i)) = 0, \quad \forall i = 1, \dots, n - 1. \quad (6)$$

where n is the number of nodes used to approximate X_i at coordinates y_i . If each node i is considered as an agent communicating its value to its neighbouring node due to $x_{i-1} - x_i$, the problem of

velocity profile approximation is a multi-agent system with the following problem for each agent i ,

$$\begin{aligned} \min_{x_i} \quad & f_i = \frac{x_{i-1} - x_i}{h} (x_i - s(y_i)) \\ \text{subject to} \quad & x_i \geq s_i \geq 0. \end{aligned} \quad (7)$$

If f_i approaches to its minimum value of zero, then the complementarity condition holds as either $(x_{i-1} - x_i)/h = 0$ or $(x_i - s(y_i)) = 0$ or both: velocity on the surface is zero and the gradient of velocity equals zero anywhere else in y -direction. To employ the parallel search, since n is the number of nodes used to approximate x , n agent problems are solved in five different locations (a – e) of the leading edge surface s shown in Fig. 4.

To solve the problem, we consider $n = 10$ nodes for approximation, and set $h = n$ to account for a regular mesh with $ne_i = 3$ to show that each node is communicating with other two neighbours. The convergence can be analysed by looking at Fig. 4. Fig. 4(a) shows the pattern of the velocity profiles on the leading edge on five different locations. Since the distance around either side of the aerofoil is the same, the average velocity above the aerofoil is the same as beneath it. So, we expect to see a similar flow profile both sides of the aerofoil. The variance-based metric in Fig. 4(b) and (c) implies a pretty fast convergence after only 25 iterations explaining that each black circle point in location b shown in Fig. 4(a) communicates its position with the neighbours before convergence.

5. Further consideration

5.1. Scalability to higher dimension

As opposed to population based methods, where the number of population grows as the dimension of the problem increases, the ability to scale up the performance is not a challenge for deterministic algorithms where mostly gradient search approaches are employed. Therefore, to test the scalability of the proposed population based algorithm in higher dimension problems, we use the velocity profile approximation problem and solve it for $n = 5, 10, 20, 30, 40$ and $n = 50$ nodes (agents). To account for variability of each run, we solve each problem 30 times and the results presented are averaged over 30 runs. We report the average number of generations to reach the convergence tolerance of less than $\epsilon = 10^{-5}$ as well as the average time taken for convergence. The scalability Fig. 5(a) suggests a mild exponential trend in computational burden as the number of nodes (agents) increases. While for $n = 5$, 0.5 seconds takes on average to solve the problem with average number of 17 generations, the average running time for $n = 50$ reaches up to 22 seconds with number of 104 generations. Fig. 5(b) presents the same situation as the one illustrated in Fig. 4(a). Fig. 5(b) indicates that at $n = 50$, the accuracy of the approximation is increased (compared with $n = 10$ in Fig. 4a) which leads to higher computational time for convergence (Fig. 5d).

5.2. Different search algorithms

We report the performance of the proposed approach on the above velocity profile approximation problem when the DE search algorithm is replaced by simple genetic algorithm (GA), genetic algorithm with elitism (GA-E) [45] and covariance matrix adaptation evolution strategy (CMA-ES) [46]. In GA and GA-E, we use roulette wheel selection, a mutation rate of 0.2 and population size of 10. In each generation of CMA-ES, new candidate solutions are sampled from the population size of 10 according to a multivariate normal distribution where a weighted combination of the $\mu = 2$ best out of $\lambda = 4$ new candidates is used to update the distribution

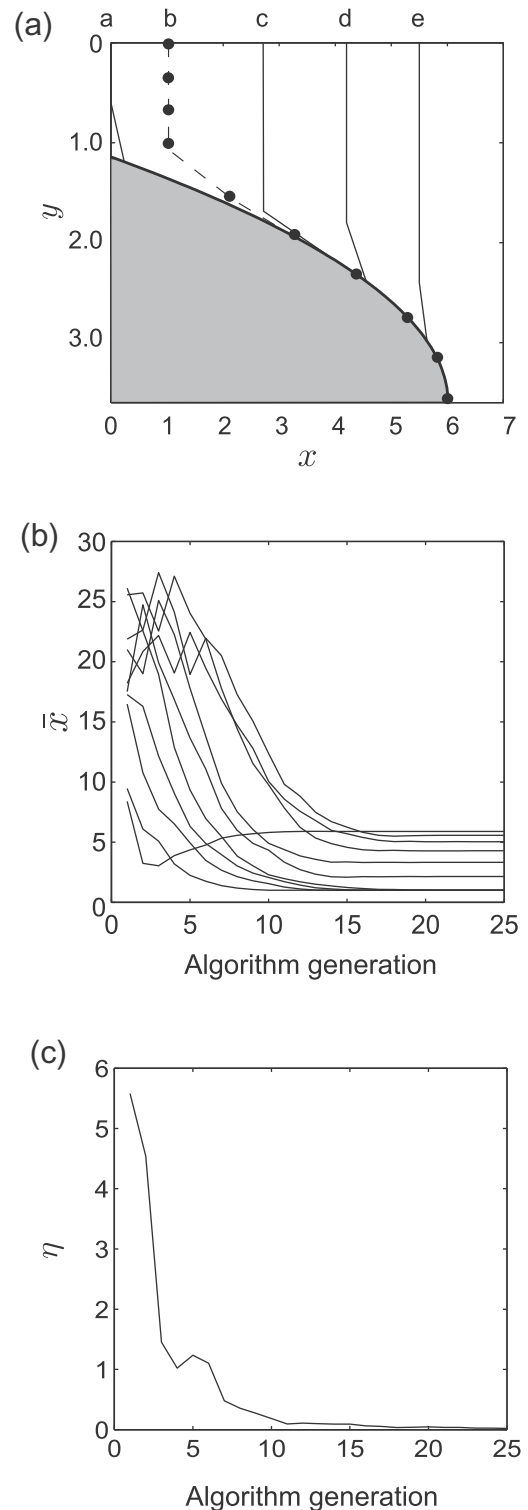


Fig. 4. The aerofoil leading edge, s , is the shaded area. (a) The pattern of velocity profile over the leading edge of the aerofoil for $n = 10$ at five different locations. (b) The pattern of velocity profile averaged in each algorithm generation in location b shown by dashed line ($x_0 = 1$). (c) The variance-based convergence metric η showing the evolution of population in each algorithm generation for problem defined in location b .

parameters. The reader is referred to the provided references for details of these algorithms. We note that Simple GA, GA-E, CMA-ES and DE are only the operators each agent use to solve their own problem as a result of communicating with neighbouring agents via linkage H . We let $n = 10$ and run the algorithm for 50 generations.

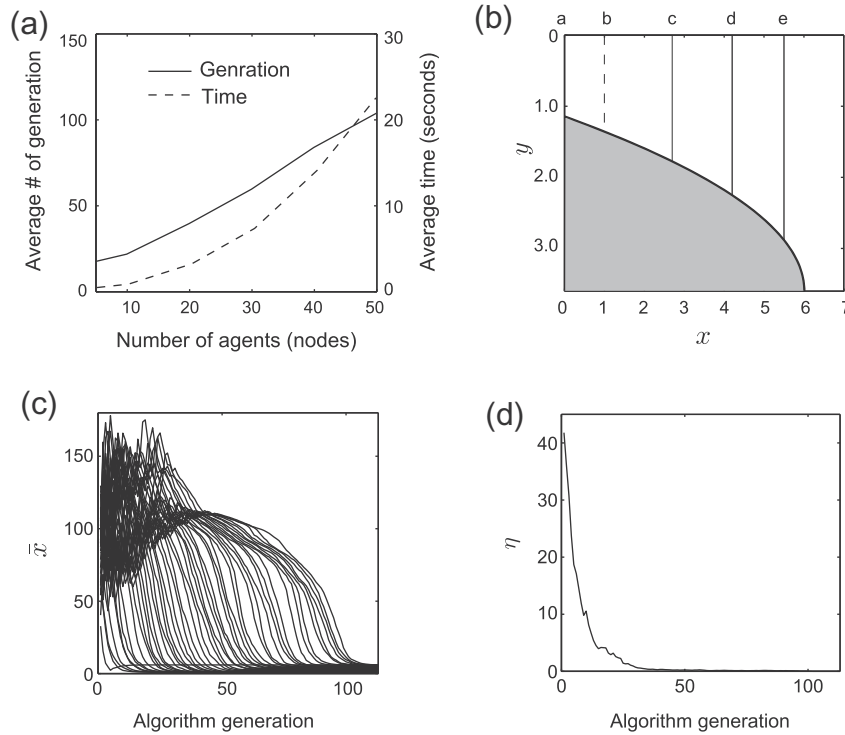


Fig. 5. The aerofoil leading edge, s , is the shaded area. (a) The average number of iteration and time in seconds for convergence to $\eta < 1e-5$ for $n = 5, 10, 20, 30, 40$ and $n = 50$. (b) The pattern of velocity profile over the leading edge of the aerofoil for $n=50$ at five different locations. (c) The pattern of velocity profile averaged in each algorithm generation in location b shown by dashed line ($x_0 = 1$) for $n=50$. (d) The variance-based convergence metric η showing the evolution of population in each algorithm generation for $n=50$ in location b .

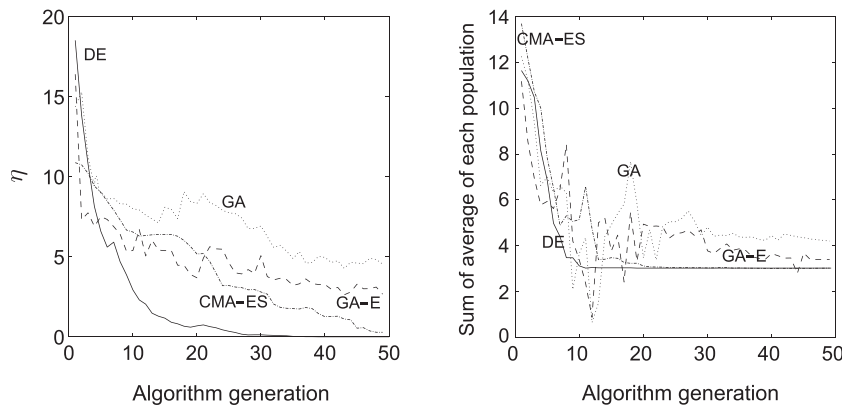


Fig. 6. The convergence of four search algorithms to solution of velocity profile approximation with $n = 10$. Left panel shows the η value and right one shows the sum of the average of population of each agent's variable in each generation.

We use η value as the performance metric to study the convergence characteristics. Fig. 6 shows the convergence behaviour of these four algorithms. As can be seen, all search operators get close to the solution. While DE and CMA-ES converged within 50 runs, it seems that the GA and GA-E requires more than 50 runs before they converge. DE and CMA-ES have smoother convergence behaviour than GA and GA-E do and the quick drop of the sum of the average of population of each agent's variables in each generation is an indication that the convergence was indeed fast.

6. Conclusion and future work

In this paper, we have developed an evolutionary approach to solve n optimisation problems simultaneously. This type of representation is used to model engineering and economic problems. The problem can be as simple as solving a system of equations or

as hard as a nonlinear combinatorial problem depending on the properties of the involved functions. We have built upon parallel genetic algorithm where separate populations for different agents are simultaneously evolved, each dealing with its own optimisation problem. Agents have partial information about the whole model and, in each generation, they synchronise their best fitted individual in the population with their neighbours. This keeps each agent aware of the other agents' decisions and in contact with the whole system. The procedure guides the search towards a compromised solution. The algorithm is illustrated on a simple two-agent problem and its performance is tested on two multi-disciplinary problems, namely on environmental water pollution and fluid dynamics, and the results demonstrate the applicability of the proposed technique to low and high dimension problems. We have investigated the performance of the approach when four different population based methods are used as the sub-problem

search algorithm. The study suggests that different algorithms can be exploited for each agent to solve its own problem.

In this study agents consider their interest in solving their local problem while interacting with each other. The applicability of the technique should be further elaborated when agents cooperates to solve a global problem. In addition, the parallel nature of the algorithm can motivate further research on designing a distributed architecture for implementation of the algorithm on multiple cores and threads synchronisations.

This framework can be extended to consider different problems in water market – where different users are trading water rights [47–49,53] in a resource constrained environment – and in plasma actuator study – where the velocity profiles induced by dielectric barrier discharge plasma actuator are necessary to be represented [50–52,54].

Acknowledgements

Six anonymous reviewers provided comments and suggestions that strongly improved the content and presentation of the paper. All errors or omissions are the authors alone.

References

- [1] M.C. Ferris, J.-S. Pang, Engineering and economic applications of complementarity problems, *Siam Rev.* 39 (4) (1997) 669–713.
- [2] P.T. Harker, B. Xiao, Newton's method for the nonlinear complementarity problem: a b-differentiable equation approach, *Math. Progr.* 48 (1–3) (1990) 339–357.
- [3] E.D. Andersen, Y. Ye, On a homogeneous algorithm for a monotone complementarity problem with nonlinear equality constraints, in: *Complementarity and Variational Problems: State of the Art*, SIAM, Philadelphia, PA, 1997, pp. 1–11.
- [4] C.D. Kolstad, L. Mathiesen, Computing cournot-nash equilibria, *Oper. Res.* 39 (5) (1991) 739–748.
- [5] T.L. Magnanti, *Models and Algorithms for Predicting Urban Traffic Equilibria*, Elsevier Science Publishers BV, 1984.
- [6] P. Panagiotopoulos, A. Al-Fahed, Robot hand grasping and related problems: optimal control and identification, *Int. J. Robot. Res.* 13 (2) (1994) 127–136.
- [7] M.M. Kostreva, Elasto-hydrodynamic lubrication: a non-linear complementarity problem, *Int. J. Numer. Methods Fluids* 4 (4) (1984) 377–397.
- [8] S.A. Gabriel, A.S. Kydes, P. Whitman, The national energy modeling system: a large-scale energy-economic equilibrium model, *Oper. Res.* 49 (1) (2001) 14–25.
- [9] S.R. Hejazi, A. Memariani, G. Jahanshahloo, M.M. Sepehri, Linear bilevel programming solution by genetic algorithm, *Comput. Oper. Res.* 29 (13) (2002) 1913–1925.
- [10] H. Li, Y.-C. Jiao, F.-S. Zhang, L. Zhang, An efficient method for linear bilevel programming problems based on the orthogonal genetic algorithm, *Int. J. Innov. Comput. Inf. Control* 5 (9) (2009) 2837–2846.
- [11] A. Koh, An evolutionary algorithm based on nash dominance for equilibrium problems with equilibrium constraints, *Appl. Soft Comput.* 12 (1) (2012) 161–173.
- [12] D. Ouelhadj, S. Petrovic, A survey of dynamic scheduling in manufacturing systems, *J. Sched.* 12 (4) (2009) 417–431.
- [13] P.I. Cowling, D. Ouelhadj, S. Petrovic, Dynamic scheduling of steel casting and milling using multi-agents, *Prod. Plan. Control* 15 (2) (2004) 178–188.
- [14] J.S. Lau, G.Q. Huang, K.-L. Mak, L. Liang, Agent-based modeling of supply chains for distributed scheduling, *IEEE Trans. Syst. Man Cybern. A: Syst. Hum.* 36 (5) (2006) 847–861.
- [15] G. Sauvageau, J.-M. Frayret, Waste paper procurement optimization: an agent-based simulation approach, *Eur. J. Oper. Res.* 242 (3) (2015) 987–998.
- [16] P. Davidsson, L. Henesey, L. Ramstedt, J. Törnquist, F. Wernstedt, Agent-based approaches to transport logistics, in: *Applications of Agent Technology in Traffic and Transportation*, Springer, 2005, pp. 1–15.
- [17] J. Böcker, J. Lind, B. Zirkler, Using a multi-agent approach to optimise the train coupling and sharing system, *Eur. J. Oper. Res.* 131 (2) (2001) 242–252.
- [18] D.G. Elliman, S.M. Youssef, A new intelligent agent-based strategy for constrained multiple destination routing problems, *Comput. J.* 47 (6) (2004) 708–727.
- [19] Y.-M. Chen, S.-C. Wang, Framework of agent-based intelligence system with two-stage decision-making process for distributed dynamic scheduling, *Appl. Soft Comput.* 7 (1) (2007) 229–245.
- [20] T. Ma, Y. Nakamori, Modeling technological change in energy systems-from optimization to agent-based modeling, *Energy* 34 (7) (2009) 873–879.
- [21] X. QuanLi, Y. Kun, W. GuiLin, Y. YuLian, Agent-based modeling and simulations of land-use and land-cover change according to ant colony optimization: a case study of the erhai lake basin, china, *Nat. Hazards* 75 (1) (2015) 95–118.
- [22] P. Forget, S. DAmours, J.-M. Frayret, Multi-behavior agent model for planning in supply chains: an application to the lumber industry, *Robot. Comput. Integr. Manuf.* 24 (5) (2008) 664–679.
- [23] K. Aksyonov, E. Bykov, O. Aksyonova, A. Nevolina, Solving the problem of efficient gas delivery with aid of decision support system: Bpsim. dss based on simulation modeling, in: *Proceedings of the 2014 Summer Simulation Multiconference*, Society for Computer Simulation International, 2014, p. 15.
- [24] E. Ngai, S. Peng, P. Alexander, K.K. Moon, Decision support and intelligent systems in the textile and apparel supply chain: an academic review of research articles, *Exp. Syst. Appl.* 41 (1) (2014) 81–91.
- [25] H. Mühlenbein, M. Schomisch, J. Born, The parallel genetic algorithm as function optimizer, *Parallel Comput.* 17 (6) (1991) 619–632.
- [26] H. Mühlenbein, Evolution in time and space-the parallel genetic algorithm, in: *Foundations of Genetic Algorithms*, Citeseer, 1991.
- [27] G. Luque, E. Alba, *Parallel Genetic Algorithms: Theory and Real World Applications*, vol. 367, Springer Science & Business Media, 2011.
- [28] E. Cantú-Paz, A survey of parallel genetic algorithms, *Calc. Paralleles, Reseaux Syst. Repartis* 10 (2) (1998) 141–171.
- [29] T.C. Belding, The Distributed Genetic Algorithm Revisited, 1995 [arXiv:adap-org/9504007](http://arxiv.org/abs/9504007)
- [30] H. Braun, On solving travelling salesman problems by genetic algorithms, in: *Parallel Problem Solving from Nature*, Springer, 1991, pp. 129–133.
- [31] E. Cantú-Paz, Designing Efficient Master-Slave Parallel Genetic Algorithms, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 1997, Report No. 97004.
- [32] J.-M. Li, X.-J. Wang, R.-S. He, Z.-X. Chi, An efficient fine-grained parallel genetic algorithm based on gpu-accelerated, in: *Network and parallel computing workshops, 2007. NPC workshops. IFIP international conference on, IEEE, 2007*, pp. 855–862.
- [33] C. Peterson, B. Söderberg, A new method for mapping optimization problems onto neural networks, *Int. J. Neural Syst.* 1 (01) (1989) 3–22.
- [34] L. Araujo, J.J. Merelo, Diversity through multiculturalism: assessing migrant choice policies in an island model, *IEEE Trans. Evolut. Comput.* 15 (4) (2011) 456–469.
- [35] M.A. Potter, K.A. De Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, *Evolut. Comput.* 8 (1) (2000) 1–29.
- [36] D. Lee, L.F. Gonzalez, J. Periaux, K. Srinivas, Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering, *IEEE Trans. Evolut. Comput.* 15 (2) (2011) 133–150.
- [37] E. Alba, J.M. Troya, A survey of parallel distributed genetic algorithms, *Complexity* 4 (4) (1999) 31–52.
- [38] A. Sinha, P. Malo, A. Frantsev, K. Deb, Finding optimal strategies in a multi-period multi-leader-follower stackelberg game using an evolutionary algorithm, *Comput. Oper. Res.* 41 (2014) 374–385.
- [39] K. Deb, An efficient constraint handling method for genetic algorithms, *Comput. Methods Appl. Mech. Eng.* 186 (2) (2000) 311–338.
- [40] K.V. Price, J.A. Lampinen, R.M. Storn, *Differential Evolution: A Practical Approach to Global Optimization*, Springer-Verlag, Berlin, Heidelberg, 2005.
- [41] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, *Artif. Intell. Rev.* 33 (1–2) (2010) 61–106.
- [42] J.B. Krawczyk, Coupled constraint nash equilibria in environmental games, *Resour. Energy Econ.* 27 (2) (2005) 157–181.
- [43] I. Abbott, A. Von Dönhof, *Theory of Wing Sections, Including a Summary of Airfoil Data*, Dover Books on Aeronautical Engineering Series, Dover Publications, 1959.
- [44] J.H. Ferziger, M. Perić, *Computational Methods for Fluid Dynamics*, vol. 3, Springer, Berlin, 1996.
- [45] R.L. Haupt, S.E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, 2004.
- [46] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evolut. Comput.* 9 (2) (2001) 159–195.
- [47] T. Erfani, I. Huskova, J.J. Harou, Tracking trade transactions in water resource systems: a node-arc optimization formulation, *Water Resour. Res.* 49 (5) (2013) 3038–3043, <http://dx.doi.org/10.1002/wrcr.20211>.
- [48] T. Erfani, O. Binions, J.J. Harou, Simulating water markets with transaction costs, *Water Resour. Res.* 50 (6) (2014) 4726–4745, doi:10.1002/2013WR014493.
- [49] T. Erfani, O. Binions, J.J. Harou, Protecting environmental flows through enhanced water licensing and water markets, *Environ. Earth Syst. Sci.* 19 (2) (2015) 675–689, <http://dx.doi.org/10.5194/hess-19-675-2015>.
- [50] R. Erfani, T. Erfani, S.V. Utyuzhnikov, K. Kontis, Optimisation of multiple encapsulated electrode plasma actuator, *Aerosp. Sci. Technol.* 26 (1) (2013) 120–127, <http://dx.doi.org/10.1016/j.ast.2012.02.020>.
- [51] R. Erfani, H. Zare-Behtash, K. Kontis, Influence of shock wave propagation on dielectric barrier discharge plasma actuator performance, *J. Phys. D: Appl. Phys.* 45 (2) (2012) 225201.
- [52] R. Erfani, H. Zare-Behtash, K. Kontis, Plasma actuator: influence of dielectric surface temperature, *Exp. Therm. Fluid Sci.* 42 (2012) 258–264.
- [53] T. Erfani, R. Erfani, Fair resource allocation using multi-population evolutionary algorithm, in: *Applications of Evolutionary Computation*, Springer International Publishing, 2015, pp. 214–224, http://dx.doi.org/10.1007/978-3-319-16549-3_18.
- [54] R. Erfani, H. Zare-Behtash, C. Hale, K. Kontis, Development of DBD plasma actuators: the double encapsulated electrode, *Acta Astronaut.* 109 (2015) 132–143.