An evolutionary algorithmic approach to determine the Nash equilibrium in a duopoly with nonlinearities and constraints

Malcolm Brady
DCU Business School
Dublin City University
Dublin, Ireland
Tel.: +353 1 7005188
Email: malcolm.brady@dcu.ie

Vamsee Krishna Mamanduru
Department of Industrial and Systems Engineering
IIT Kharagpur
Kharagpur, India
Email: vamsee.iitkgp2011@gmail.com

Manoj Kumar Tiwari
Department of Industrial and Systems Engineering
IIT Kharagpur
Kharagpur, India
Tel.: +91 3222 283746
Email: mkt09@hotmail.com

An evolutionary algorithmic approach to determine the Nash equilibrium in a duopoly with nonlinearities and constraints

Abstract

This paper presents an algorithmic approach to obtain the Nash Equilibrium in a duopoly. Analytical solutions to duopolistic competition draw on principles of game theory and require simplifying assumptions such as symmetrical payoff functions, linear demand and linear cost. Such assumptions can reduce the practical use of duopolistic models. In contrast, we use an evolutionary algorithmic approach (EAA) to determine the Nash equilibrium values. This approach has the advantage that it can deal with and find optimum values for duopolistic competition modelled using non-linear functions. In the paper we gradually build up the competitive situation by considering non-linear demand functions, non-linear cost functions, production and environmental constraints, and production in discrete bands. We employ particle swarm optimization with composite particles (PSOCP), a variant of particle swarm optimization, as the evolutionary algorithm. Through the paper we explicitly demonstrate how EAA can solve games with constrained payoff functions that cannot be dealt with by traditional analytical methods. We solve several benchmark problems from the literature and compare the results obtained from EAA with those obtained analytically, demonstrating the resilience and rigor of our EAA solution approach.

Key words: Nash Equilibrium, Evolutionary Algorithm, Swarm Intelligence, Constrained Games, Non-Linear Payoff Functions.

Abbreviations:

NE:	Nash Equilibrium
EAA:	Evolutionary Algorithmic Approach
BRF:	Best Responsive Function
PR:	Promising region
PSO:	Particle Swarm Optimization
PSOCP:	Particle Swarm Optimization with Composite Particles
GBP:	Global Best Position
LBP:	Local Best Position

1. Introduction

The concept of Nash equilibrium (Nash, 1951) is central to the analysis of strategic interactions between different firms involved in non-cooperative games. Nash equilibrium (NE) is the stable state of a system involving competitive firms who try to optimize their personal pay off functions by employing a wide range of strategies. In other words, given the actions of other firms, no firm can improve its position by adopting a different strategy. In general, individual profit functions are considered as pay off functions. However it can also be minimizing loss function, maximizing sales etc. For the purpose of better illustration of our approach we restrict ourselves to a duopoly case where two firms will be involved in the game and pay off functions are their individual profit functions. Each firm responds in the best possible way, given the strategies of their competitors. This yields a best response function (BRF) for each firm that is employed to determine its decision, taking into account the decisions made by rival firms. Analytically, the point of intersection of all the BRFs, each one corresponding to a firm competing in the market, is known as the Nash equilibrium (Nash, 1950)**.**

The traditional analytical approach of attaining Nash Equilibrium involves two stages. The first stage is to determine mathematical expressions for BRF of each firm involved in the game by maximizing its corresponding pay off function using calculus principles. The second stage involves solving the system of best responses of all firms to attain NE. Hence in the analytical approach, we must differentiate up to second order in order to find out the BRF of each firm and solve the resulting cumbersome equations. This involves a mathematical obligation to check the continuity and differentiability of each pay off function up to its second order. Games that involve complicated pay off functions with sophisticated demand and cost functions require extensive mathematical skills to solve the differential equations which arise analytically (Agiza & Elsadany, 2004; Daskalkis & Goldberg, 2009**;** Fabrikant, Papadimitriou, & Berkeley, 2004; Grant Schoenebeck, 2006**;** Maskin, 1986). This leads to the adoption of simplifying assumptions such as linearizing the non-linear functions or using adopting symmetrical functions (Cheng, Reeves, Vorobeychik, & Wellman, 2004) In addition, games are often solved with little consideration of constraints such as production capacity of each firm, production costs, and environmental obligations.

A number of authors have used numerical or algorithmic approaches to solve for the Nash equilibrium (Razi, Shahri, & Kian, 2007**;** Konak, Kulturel-Konak, & Snyder, 2015; Wang, 2013). Zhang & Zhou (2012) have built supply chain network equilibrium models as non-linear problem formulations (Zhao & Nagurney, 2008); they employed smooth Newton algorithm (Ma, 2011) for solving the non-linear games but struggled to prove the existence and uniqueness of the equilibrium. A three stage game theoretic linear and un-constrained supply chain model is built by Xiao, Jin, Chen, Shi, & Xie (2010) employed traditional analytical approach and carried out complex calculations to examine existence of Nash equilibrium. Chern, Chan, Teng, & Goyal (2014) have proposed a non-cooperative Nash equilibrium solution in a vendor buyer supply chain model that could capture delay in payments; though they presented an unconstrained mathematical model, they had to solve a system of partial differential equations to prove the existence an equilibrium.

In this paper we put forward an evolutionary algorithmic approach to obtain the Nash equilibrium. An evolutionary algorithm is used to find the optimal value of a payoff function subject to a given set of constraints. Evolutionary algorithms work on the principle of random search technique in finding out the optimal value of the payoff function. Those random search techniques are iterative in nature and are not concerned with the nature and complexity of payoff function that is being maximized. Particle swarm optimization (PSO) provides a heuristic to determine the optimal value of a non-linear function (Kennedy & Eberhart, 1995). It is a population-based heuristic drawing on swarm intelligence with origins in the study of bird flocking, fish schooling and other swarm-type behaviours found in the biological world. Several variants have been developed to tackle complex situations with improved speed of convergence and quality of solutions, avoiding local entrapment and premature convergence (Nezami, Bahrampour, & Jamshidlou, 2013). These include a hybridized PSO back propagation algorithm (J.-R. Zhang, Zhang, Lok, & Lyu, 2007) and a modified PSO for adaptive equalization (Al-Awami, Zerguine, Cheded, Zidouri, & Saif, 2011). Particle Swarm Optimization with Composite Particles (PSOCP) is the particular evolutionary algorithm employed in this paper (Liu, Yang, & Wang, 2010). Its qualities are described in the algorithm description in Section 4. Use of PSO as an optimization heuristic is still relatively novel in the field of production economics (Nearchou, 2011) but has been applied to assembly line (Nearchou, 2011), inventory classification (Tsai & Yeh, 2008) and combinatorial optimization problems (Kemmoé Tchomté & Gourgand, 2009).

We use our evolutionary algorithmic approach to solve a variety of duopoly problems. EAA can deal with nonlinearities, discreteness and constraints that would be intractable using analytical approaches. As we are using a meta heuristic in the solution procedure we initially arrive at very near optimal solutions and hence the approximate Nash Equilibrium (Constantinos Daskalkis, Paul W. Goldberg, 2009). We then improve this approximate solution to obtain a solution very close to the exact Nash Equilibrium. The results from the PSOCP model are benchmarked against analytically derived solutions. The solutions for the Nash equilibria obtained from the EAA computational experiments deviate negligibly from the analytically derived solutions (less than 0.1%). The paper is novel in that, to our knowledge, this is the first time that evolutionary algorithms have been used to solve for the Nash equilibrium in duopoly models.

2. Model and problem description

Let us consider that a homogenous commodity is being produced by n firms where the cost incurred to i^{th} firm for producing q_i units of good is $C_i(q_i)$, where C_i is an increasing function of q_i. The price of the product being sold is determined by the demand for that good in the market and the total output of the market (Q). As we presume that quantity of production is equal to the demand, specifically if the total output of the market is Q then the market price is determined through the function $P(Q)$ where P is called the "inverse demand function". If each firm produces and sells q_i units, total market output is given by (1).

$$Q = \sum_{i=1}^{n} q_i \tag{1}$$

The market price of that product is determined by the function $P(Q)$. Then revenue of i^{th} firm $\left(R_i\right)$ is calculated from equation (2)

$$\begin{aligned} R_i &= q_i \times P(Q) \\ &= q_i \times P(q_1 + q_2 + ... + q_n) \end{aligned} \tag{2}$$

We consider the profit function of each firm as its payoff function. So the profit function (π_i) of i^{th} firm is computed by deducting the total cost of production from the revenue it earns for q_i units in (3).

$$\begin{aligned} \pi_i(Q) &= q_i \times P(Q) - C_i(q_i) \\ &= q_i \times P(\sum_{i=1}^{n} q_i) - C_i(q_i) \end{aligned} \tag{3}$$

The nature and complexity of the payoff function of each firm will be determined based on the level of complexity involved with the inverse demand function and cost function. So, we expand our formulation to more complex scenarios by adopting a tier wise increase in complexity from the all linear case to a more sophisticated duopoly game in subsequent Sections 2.1 to 2.4.

2.1 Linear demand and linear cost function

We start with a simple scenario by considering a Cournot model of a duopoly game with linear cost and payoff functions. Suppose we consider the linear inverse demand function as in (4).

$$\mathrm{P(Q)} = \begin{cases} \lambda - \mathrm{Q} & \text{if } \mathrm{Q} \le \lambda \\ 0 & \text{if } \mathrm{Q} > \lambda \end{cases} \tag{4}$$

The cost function of each firm as in (5).

$$C_i(q_i) = (F_i + c_i \times q_i) \tag{5}$$

Where λ and c_i are positive constants and F_i is a fixed cost incurred to the i^{th} firm, and c_i represents unit price of the commodity of the i^{th} firm. So, based on these assumptions, payoff function of i^{th} firm becomes

$$\pi_i(q_1, q_2) = q_i \times P(\sum_{i=1}^{n} q_i) - (F_i + c_i \times q_i) \tag{6}$$

In case of duopoly, the payoff functions of the individual firms become (7).

$$\begin{aligned}\pi_1(q_1,q_2) &= q_1 \times P(q_1+q_2)-(F_1+c_1\times q_1)\\ \pi_2(q_2,q_1) &= q_2 \times P(q_1+q_2)-(F_2+c_2\times q_2)\end{aligned} \tag{7}$$

2.2 Non-linear demand function

It is useful to consider non-linear market demand functions as these may better depict certain market situations. We adopt two standard scenarios from (Martin, 2002). They include exponential and hyperbolic demand functions. The generalized forms of both models are as shown in equations 8 and 9 respectively.

$$\pi_i(q_1,q_2...q_n) = q_i \times \alpha_i \times e^{-1/\beta_i \left(\sum\limits_{i=1}^{n} q_i\right)^{\gamma}} \tag{8}$$

Where α_i , β_i, γ are positive constants. The above function delineates the demand function as an exponentially decreasing function which may be more representative of certain markets. The other scenario represents demand as hyperbolic in (9).

$$\pi_i(q_1,q_2...q_n) = \left(\frac{A_i}{k_i + \sum\limits_{i=1}^{n} q_i}\right)\times q_i \tag{9}$$

Where A_i, k_i are positive constants. It is quite possible for markets to have a sinusoidal payoff functions presented in (Razi, Shahri, & Kian, 2007). The nonlinearity in the payoff function may lead to multiple Nash Equilibria. The payoff functions are defined as follows:

$$\pi_1(q_1,q_2) = 21+q_1\times\sin(\pi\times q_1)+q_1\times q_2\times\sin(\pi\times q_2) \tag{10}$$

$$\pi_2(q_1,q_2) = 21+q_2\times\sin(\pi\times q_2)+q_1\times q_2\times\sin(\pi\times q_1) \tag{11}$$

Thus through the above examples, we have considered a diversified range of scenarios with non-linear demand function. However, it is also useful to consider non-linearity in the cost function.

2.3 Non-linear cost function

Though there exists different kinds of non-linearity in the cost function, we have adopted a standard form of non-linear cost function from (Martin, 2002) defined below, so that it takes care of both economies and diseconomies of scale of production:

$$C_i(q_i) = c_{i,1} - c_{i,2}\times q_i + c_{i,3}\times q_i^{2} - c_{i,4}\times c_i^{3} + c_{i,5}\times q_i^{4} \tag{12}$$

where $c_{i,j} \geq 0$, $\forall i$ and j .

The payoff function with linear demand and non-linear cost is shown in equation (13) below:

$$\pi_i(q_1,q_2) \;=\; q_i \times P(\sum_{i=1}^{n} q_i) - (c_{i,1} - c_{i,2} \times q_i + c_{i,3} \times q_i^{2} - c_{i,4} \times q_i^{3} + c_{i,5} \times q_i^{4}\ ) \qquad (13)$$

Equations (14) and (15) represent the scenario having non-linearity in both demand and cost functions:

$$\left.\begin{array}{l} \pi_i(q_1,q_2,..q_n) = q_i \times \alpha_i \times e^{-1/\beta_i \left(\sum\limits_{i=1}^{n} q_i\right)^{\gamma}} - C_i(q_i) \\ C_i(q_i) = c_{i,1} - c_{i,2} \times q_i + c_{i,3} \times q_i^{2} - c_{i,4} \times c_i^{3} + c_{i,5} \times q_i^{4} \end{array}\right\} \qquad (14)$$

$$\pi_i(q_1,q_2,..q_n) \;=\; \frac{A_i \times q_i}{k_i + \sum\limits_{i=1}^{n} q_i} - (\mathrm{c}_{i,1} - c_{i,2} \times q_i + c_{i,3} \times q_i^{2} - c_{i,4} \times q_i^{3} + c_{i,5} \times q_i^{4}) \qquad (15)$$

It is easy to observe that the non-linear cost function mentioned in (15) can also take care of cost function of linear nature by making all the coefficient terms except the first order term equal to zero.

2.4 Constrained functions

In this section we consider capacity limitations on the production quantities of each firm. It is quite possible to have a discrete natured cost function with respect to quantity of production because the unit cost of a product can be lesser for bulk production. Hence the cost function can also take the forms of linear, nonlinear and simultaneously both. This discrete natured cost function can be taken care of mathematically by subjecting our payoff function to the set of constraints derived from putting limits on the production quantity and discrete cost of production of a commodity. Suppose if the discrete cost function is split into k intervals as shown in equation (18) and capacity of each firm is set as expressed in inequality (17), the generalized form of the model will be as given below:

$$\text{Maximize:} \quad \pi_i(Q) \;=\; q_i \times P(\sum_{i=1}^{n} q_i) - \sum_{k=1}^{K} f_k^{i} \times C_i^{k}(q_i) \qquad \forall\ i = 1, 2 ... n^{th} \text{ firm} \qquad (16)$$

Subject to:

$$q_i \le Q_i^{cap} \qquad \forall\ i \qquad (17)$$

$$C_i(q_i) = \begin{cases} C_i^1(q_i) & \text{if } 0 < q_i \le Q_i^1 \\ C_i^2(q_i) & \text{if } Q_i^1 < q_i \le Q_i^2 \\ C_i^3(q_i) & \text{if } Q_i^2 < q_i \le Q_i^3 \\ . & \\ . & \\ C_i^k(q_i) & \text{if } Q_i^{k-1} < q_i \le Q_i^k \end{cases} \qquad \forall\, i \tag{18}$$

$$f_k^i = \begin{cases} 1 & \text{if } \mathrm{Q}_i^{k-1} < q_i \le \mathrm{Q}_i^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ i \tag{19}$$

$$\sum_k f_k^i = 1 \quad \forall i \tag{20}$$

where Q_i^{cap} is production capacity of i^{th} firm and inequality (17) ensures that the production quantity of each firm will be within its capacity. Equation (16) represents the fact that each individual firm tries to maximize its pay off within its feasible zone within the existing constraints frontier. Equation (18) allocates different costs for different ranges of production whose range is split into k intervals. The expression f_k in equation (19) can be regarded as an activating binary constant which enables the contribution of k^{th} interval of the cost function if q_i falls in the k^{th} interval range. The expression $C_i^k(q_i)$ is the cost function corresponding to k^{th} interval of the split which can be both linear and non-linear. Though the cost of production depends on the quantity that the firm produces, equation (20) allocates that particular cost corresponding to the production quantity.

Another constraint that we introduce into the model is that of pollution limits. In fact, it is quite likely that different regulatory constraints exist to control the pollution level to ensure environmental protection from pollution prone commodities. Here we follow the concept of river basin pollution game (Haurie & Krawczyk, 1997) with a numerical example given by Mathiesen (2008), where i^{th} producer emits pollution a_{pi} and maximum capacity b_i that is allowed on aggregate pollution. That restriction can be mathematically represented as in equations (21) and (22). Suppose there is a scenario in which the firms are located along a river and pollutants may be expelled into to the river, where they disperse. Emission e_i is assumed to be proportional to production level. A regulatory authority monitors concentration of pollution at two downstream stations. Decay and transportation coefficient of pollution from i^{th} firm to control p^{th} is denoted by u_{pi}, then the regulator's constraints can be portrayed as in (21) and (22):

$$\sum_i u_{pi} \times e_i \times q_i \leq b_i \qquad \forall p \tag{21}$$

$$\sum_i a_{pi} \times q_i \leq b_i \qquad \forall p \tag{22}$$

where b_i, e_i, u_{pi}, a_{pi} are assumed to be positive. This constrained model can approximate a variety of situations by adjusting the values of parameters as required. Though all our constraints are linear in nature in fact, they can be of any forms like linear, non-linear or a combination of both. Now it is quite challenging to solve these kinds of scenarios. This brings us the usefulness of an EAA in solving these complex models efficiently.

3. Solution methodology

In this section, we proceed by giving a brief description about traditional analytical approach and EAA to determine the Nash Equilibrium. The following illustrates the solution procedure:

Definition-1: Given the strategy of other competing players in the game, optimal strategy or a best response of a particular firm is the strategy that maximizes its payoff function subject to existing constraints;

*Definition-2***:** The Best Response Function (BRF) can be defined as a function that takes the given strategies of competing players as an input and gives as an output the optimal strategy to be followed by it in order to maximize its payoff. Analytically, BRF of i^{th} firm is an extrapolation of the optimal strategies for all firms across the range of strategies available in the game.

3.1 Analytical approach

The analytical approach is a two stage process. The first stage involves finding out the BRF of each firm by differentiating its payoff function with respect to its strategy and solving that first order derivative to get the zeroes of derivative function of each firm. Those zero functions have to be validated as its best response by checking the sign of double derivative. Second stage proceeds with solving the system of BRFs of the game to arrive at Nash Equilibrium. The advantages of this approach are that it is simple to understand and gives the exact Nash equilibrium. However, the analytical approach cannot be employed if any part of the payoff function is either discontinuous or non-differentiable up to second order. With regard to constrained games, ensuring the existence of Nash equilibrium is a major concern. To do that, the convexity or concavity of the payoff function of each firm has to be figured out before we find the Nash Equilibrium. Based on the nature of the solution space, there exist techniques to find out the solution in limited cases but these are complex and mathematically rigorous.

3.2 Evolutionary algorithmic approach

Unlike the analytical approach, EAA does not literally find the mathematical expressions of BRFs of each firm to find out Nash Equilibrium; rather it uses a graphical approach to do the same. Broadly the technical framework of EAA consists of two stages. The first stage involves drawing the plots of BRFs of each firm using the evolutionary algorithm. The second stage involves the finding of intersection points of the system of BRF plots to give the Nash Equilibrium.

Based on the beforehand approximate strategies of different players involved in the game, a wide range is defined that could cover all the strategy sets of every firm. To draw the BRF of a particular firm, the strategies of the remaining competing firms are fixed at N_p number of different strategic points across the defined wide range. Then the best response of that particular firm at each point of the N_p points is determined using the evolutionary algorithm. The EEA takes the strategies of competing firms at each point as input and gives the best response of corresponding firm finding maximum of its payoff function. The set of all the best responses with respect to its competing firms are plotted at the N_p different points across the wide range. By extrapolating all the optimal points we get the BRF of that particular firm. We repeat this procedure to determine the best reaction functions of the remaining firms. The intersection point of all the BRFs will turn out to be our required NE of the game.

As the concept of extrapolation is used in the process of finding out Nash Equilibrium, it is important to have a sufficient number of points for drawing out the plot of BRF of each firm. However, to reduce the time of computation in the whole search for Nash Equilibrium in the strategy space, a two tier search is employed. The first tier involves finding out the approximate vicinity of Nash Equilibrium in a wide range with a smaller number $(N_p < 10)$ of points for extrapolation. Once the promising vicinity is established from this wide range search, a more detailed search is carried out in the promising region around Nash Equilibrium. To determine the promising space for exclusive search, the Self-Adaptive Range (SAR) technique is employed. Though the SAR technique is empirical in nature, experimental results show that SAR enables us to move very close to the exact equilibrium with less time and complexity of computation. The overall design of EAA is represented in the form of pseudo code in Table 1.

<< Insert Table 1 >>

3.3 Self Adapting Range (SAR) technique

Based on the equilibrium that we get in the first wide range search, we estimate an upper boundary (UB) of the promising range of equilibrium using equations (23) and carry out a rigorous search by plotting BRFs with a larger number of points in the promising region to arrive at a point very close to the true Nash Equilibrium:

$$\text{UB of PR} = \max\text{imum}(NE_{wr}) + \mu \times 10^{\eta} \qquad (23)$$

Where NE_{wr} is the set of strategies of all the firms at NE obtained during initial wide range search i.e. $NE_{wr} = \left\{q_1^*, q_2^*, q_3^* ....q_n^*\right\}_{wr}$ and η are the least positive integers which satisfies the inequalities mentioned in equations (24).

$$10^{\eta} \geq \text{maximum}(NE_{wr}) \tag{24}$$

Where μ is a positive rational number less than one. The constant μ can be termed as the tolerance random number that regulates the extra precision around the true Nash equilibrium in which we conduct the detailed search. Through rigorous simulation, it was ascertained that μ should be set at values between 0.4 and 0.7 to obtain good results. If multiple Nash equilibria are found to exist during the wide range search, the upper bound of the promising region is set at the maximum value of strategies involved in the multiple Nash equilibriums with extra precision around it using equations 23 and 24. In general, most of the production firm will have non negative amounts of production units so the lower bound of promising region is assumed to be equal to zero. To relax this assumption and model the games with negative values one can build equations similar to equations 23 and 24 and solve. Once the promising range is determined, incremental steps are used to split the promising range and construct N_p different strategic points. Exploring the promising range with incremental steps ensures uniformity in the search, which facilitates in building up a reliable BRF of each firm.

$$\text{Incremental Step} = \frac{\text{Upper Bound of Promising Range } (PR)}{\text{Number of points used for plotting BRF } (N_p)} \tag{25}$$

During exhaustive search in the promising region, the number of points (N_p) considered to plot a BRF is to be defined based on the complexity and sensitivity involved in the problem. In general, results show that keeping N_p more than 20 gives us acceptable solutions. To illustrate the computational experiments presented in Section 5, we kept $N_p = 20$. If the game is sensitive to the precision involved in the values of Nash Equilibrium, we can increase N_p and compress the promising range near the Nash Equilibrium to conduct the search exhaustively around the true Nash Equilibrium. The same applies even if there exist multiple Nash Equilibria.

4. Technical description of PSOCP Algorithm

This section provides the fundamental technical blocks that build up the PSOCP algorithm that is used as evolutionary algorithm for finding the N_p number of best response points of each firm across the range of strategies available for the other competing firms. Those best response points are joined to get the BRF of corresponding firm. The best response is the strategy that maximises the payoff function of the firm given the strategies of competing firms. Evolutionary algorithms find the optima of a payoff function using random search techniques in an iterative manner. Particle swarm optimization is one such algorithm. PSO tracks the optimal value with randomly generated solution vectors known as particles. The number of swarm particles that participate in the search is called the swarm size (n). The dimension(d)or length of each particle in the swarm is generally equal to the number of variables involved in the payoff function. In Cournot games, payoff function of a firm is just a function of its production quantity or strategy. So the dimension $(d)$of each particle is one. Each particle will be passed into the payoff function to get the value of payoff known as fitness value in artificial intelligence literature. So, the swarm is simply a $n \times d$ matrix. The mechanism of tracking the optima of payoff function involves mathematical operations on the swarm.

The swarm particles reach the vicinity of optimal value of a payoff function by effective exchange of information. To begin the search a set of n swarm particles are randomly generated to cover the whole search space. Then mutual exchange of information takes place among the particles participating in the search regarding which particle has the best fitness value. The particle with the best fitness value out of the total swarm is called the global best position (GBP). Suppose that the k^{th}particle out of $n$particles has got the best fitness position; then the remaining $n-1$ particles try to move towards thek^{th}particle in an iterative manner. Exchange of information takes place among the swarm: during the movement some other particle may come across better fitness position than thek^{th}particle; this information is again exchanged and search goes on. The value of the global best position found among the swarm either improves by iteration or remains same but does not degrade. After several iterations all the particles converges to a near optimal region of the fitness function after which there will be no significant improvement in the best fitness value of the function. The GBP of the converged swarm is taken as the true optimal value of the payoff function. Another key concept involved in this exploring mechanism is recollecting of information. Each particle will have the ability to remember the best fitness position that it has come across up until the current iteration; this is called the local best position (LBP). The LBP is different for different particles during movement. If a particle moving towards GBP encounters a worse position than its LBP, it stays at its LBP rather than make a move. This ability of remembering and exchanging of information eventually brings all the swarm particles close to the existing true global optima of a given payoff function. The efficiency of the algorithm both in terms of quality and quickness of arriving at global optima depends on how the particles exchange information and how quickly they move towards the promising region. There is however scope for improving the efficiency of the algorithm giving birth to different variants of evolutionary algorithm. This paper uses one such variant of swarm intelligence called particle swarm optimization with composite particles (PSOCP) by Liu et al. (2010). PSOCP has desirable advancements in the technical framework of the algorithm allowing efficient exploration of optimal values of payoff functions to take place.

4.1 Particle Swarm Optimization with Composite Particles (PSOCP)

PSOCP is a variant of the evolutionary algorithm known as PSO (Kennedy & Eberhart, 1995) which works on swarm intelligence while tracking out global optima. It was initially developed by Liu et al. (2010) in order to tackle dynamic environments. The concept was inspired by the phenomenon of interaction of elementary members in each composite particle through a velocity-anisotropic reflection scheme (VAR) derived from the physics discipline. Compared to basic PSO, it uses the "concerted action" principle explained in Liu et al. (2010) while updating the elementary particles for next iteration but maintaining a similar procedure to that used for updating the pioneer particles of composite particles. This feature of integral movement of particles facilitates the swarm to share the beneficial information in a constructive manner which ensures that elementary particles move under correct guidance in accordance with the fittest particles. This enables the swarm to make an easy decision support system in the space to explore the promising region around exact optima while avoiding collision and velocity-slackening of elementary particles. A scattering operation is also adopted to ameliorate the fittest elementary particle available to a more desirable direction. When a composite particle generally converges, the scattering operation is triggered which enables the swarm avoid local entrapment. The VAR scheme helps the elementary particles to look for more extensive solution space.

4.2 Position modulation function

This function ensures that all variables should lie within their corresponding boundaries. The function basically takes the swarm to be modulated as its input and modifies that swarm according to the predefined limits and gives the modified swarm as the output. The approach is briefly depicted with the help of pseudo code in Table 2.

<< Insert Table 2 here>>

4.3 Composite Particles

As per the procedure mentioned in Liu et al. (2010), we adopted the same method to construct the composite particles in this paper. A composite particle is defined as a three particle group with similar features. We consider the Euclidean distance between one particle and another so that each group forms a near equilateral triangle. A "worst is first" criteria is utilized in constructing the composite particles. The procedure starts by arranging all the particles in ascending order of their fitness and the worst particle is selected as a first particle in the composite particle being constructed and the remaining two particles are selected based on Euclidean distance which are close to the weakest particle in that composite particle. This procedure iterates until all possible composite particles are constructed. The particles that remain unconstructed in the end are named as independent particles. These will ultimately turn out to be best fit particles. Composite particles usually handle two major concerns. The primary concern is to develop a method which would increase the interaction among elementary particles in such a way that more promising search space would be explored. A secondary concern is that it facilitates VAR and scattering operations on the composite particles being constructed from which we can improve the performance of the algorithm. The pseudo code shown in Table 3 depicts the procedure adopted for construction of composite particles.

<< Insert Table 3 >>

4.4 Scattering and VAR operations

To ensure local diversity within each composite particle, we adopted scattering and VAR operations which facilitates in avoiding local entrapment of our heuristic, a major drawback of meta heuristics in computational intelligence literature. The procedure proceeds by utilising the concept of convergence. A composite particle is said to be converged when the Euclidean distance between the worst particle and the other particles inside it becomes less than the threshold Φ. A scattering operation is performed whenever the convergence criteria are met with in the composite particle. If the position of the elementary particle having the best fitness value in the composite particle is represented by F and other two non-pioneer particles denoted by E_1 and E_2, then the particle with best fitness F scatters along the direction $\overrightarrow{E_1F}$ and $\overrightarrow{E_2F}$ in order to replace E_1 and E_2. Now the new composite particle contributes to the point F, S_1 and S_2 as shown in Fig. 1**.**

<< Insert Fig. 1 >>

A velocity anisotropic reflective (VAR) operation is proposed along with a scattering operation to make sure that the worst particles move in an improved direction towards a promising region in the search space. This enhances the exploration and better tracking of global optima in the search space. The procedure involves two sub steps with an adoptive step size and pioneer particle selection as defined in Liu et al., (2010). Both of these operations are proposed for better tracking of optima in dynamic environments. The pseudo code given in Table 4 depicts the schema for carrying out scattering and VAR operations.

<< Insert Table 4 >>

4.5 Overview of PSOCP for optimizing payoff functions

The innovative qualities of the algorithm - construction of composite particles, scattering and VAR - have been adopted to design a variant of PSOCP to tackle complex game theoretic scenarios smoothly. The algorithm begins with random initialization of the swarm and its initial velocity and runs in an iterative manner until the termination criterion is met. Each iteration involves tracking of the personal best for each pioneer particle, the global best particle, construction of composite particles, scattering operation, VAR scheme operation and updating the swarm for next iteration. At the end of each iteration the position of each updated particle is checked to ensure that it lies within its predefined boundaries using the function position modulation discussed in Section 4.2. Eventually the algorithm arrives at a near optimal solution of the fitness function which we are trying to optimize. Hence this algorithm can be used to predict the optimal decisions or best responses of all the firms corresponding to their pay off functions. An overview of the algorithm is presented in the form of pseudo code given in Table 5.

<< Insert Table 5 >>

4.6 Handling constrained games

Though there are many ways to address constrained scenarios, we have adopted a penalty function method to overcome these problems in EAA. The first reason is to do with the randomness involved in generating and updating the solutions in our evolutionary algorithm, which may not facilitate us in giving a feasible swarm in every iteration. The second reason has to do with the ease with which the penalty function approach handles complex constraints. A penalty function is constructed by penalizing each constraint with an appropriately large number in proportion to its violation and annexing all those penalty values as one function. Then this penalty function will be appended to the original objective function as a negative effect. As our algorithm tries to rehabilitate and improve the payoff function in every iteration this consolidated objective function ensures reduction in violation of all the constraints in later iterations, as violation of any constraint shows a negative effect in the payoff function. This enables us to end up with a feasible solution after substantial number of iterations whose penalty function value falls to zero due to continuous reduction in violation. So, ultimately this makes a clear path for our model to return the best response strategy of each firm at every stage by contemplating all the constraints directly without any further attention. Several approaches in constructing penalty function are defined in Yeniay (2005). Through this technique we curtail the difficulty concealed with handling all kinds of constraints that exist in the problem. Hence in EAA, we directly plot the points that are feasible with respect to all the constraints that frontiers the space and plot the required BRF. Hence penalty function approach is adopted to overcome constrained scenarios, which is a conventional behaviour of general competitive market.

4.7 Sensitivity analysis and optimal parameter setting

As we are using an evolutionary heuristic, it is necessary to look for the appropriate settings of the prime parameters that amend the efficiency of the algorithm in finding the better optimal solutions with less complexity in computation. The crucial parameters involved in our algorithm that may affect the dominant properties of the algorithm are inertia weight (w), accelerating coefficients (k_1, k_2), diversification parameter used in VAR operation (R_{step}), stretching parameters $(\mathrm{S}_{step\,\min,}\mathrm{S}_{step\,\max})$, and the threshold Euclidean distance limit (Φ) involved in the scattering operation. It is imperative to fix optimal settings of the parameters, which vary from situation to situation. As we deal with diversified scenarios, these parameter settings cannot be stated as being invariably the best. Hence in particular with solving the game theory models, we present the optimal data set of the above mentioned parameters obtained by conducting 30 repeated runs for each numerical example that is considered to represent the scenarios modelled above. Few parameter settings are adopted from the conclusions drawn with the numerous experiments presented in Liu et al. (2010).

We make an observation that we are getting $k_2 >> k_1$ at the optimal parameter setting representing the fact that it facilitates the PSO variants to undergo a biased search towards global optima than local optima. This may also endanger the fact that search may miss a better optimal solution which is encountered by the diversification operations adopted in our heuristic. The important parameter that determines the efficiency of the algorithm in different scenarios is the stopping criteria to be employed while figuring out the optimal points for tracing out each BRF. This is exercised in two ways, first one being limiting the number of iterations that algorithm runs and the other is by predefining a threshold converging distance beyond which the algorithm does not search. That threshold distance depends on the accuracy in the quality of solutions needed and also on the range of fitness values. Here we employed the first one by defining the constraining maximum number of iterations (E) as 100 which is verified from the 30 repeated experiments on each scenario. In fact attention should be paid here when we come across a complex situation if it requires more iterations than in the case of constrained optimization. Table 6 shows suitable range of parameters at which our test instances have been solved.

<< Insert Table 6 >>

5. Computational experiments

The discussion is divided into two sections. The first section presents the optimal parameter settings of the algorithm at which the algorithm is sufficiently robust and efficient to tackle complex payoff functions. Secondly, we present several, increasingly complex, numerical examples corresponding to the models presented in Sections 2.1 to 2.4. The solutions obtained through EAA are compared with those gained from an analytical approach to show the robustness, ease and efficiency of EAA in attaining the Nash Equilibrium, even in games with high levels of complexity.

5. 1 Linear demand and cost

If we consider a duopoly case in which linear inverse demand function of each firm as $P(q_1,q_2) = 3 - 0.01\times(q_1 + q_2)$ and cost function as defined in Section 2.1 with the parameters as $(F_1, \mathrm{c}_1) = (0.10, 0.01)$ *and* $(F_2, \mathrm{c}_2) = (0.12, 0.05)$ then payoff function of both firms becomes

$$\begin{aligned}\pi_1(q_1,q_2) &= q_1 \times (3 - 0.01(q_1 + q_2)) - (0.10 + 0.01\times q_1)\\ \pi_2(q_1,q_2) &= q_2 \times (3 - 0.01(q_1 + q_2)) - (0.12 + 0.05\times q_2)\end{aligned}$$

The above data is extracted from the data referred to in (Mathiesen, 2008**).** Results for Mathiesen’s problem 5.2 are presented in Table 7 and the BRFs of both firms and NE is shown in figure 2. The units for best strategy of all the firms presented in all the subsequent tables presented throughout Section 5 are the number of units of production quantities to be produced by the corresponding firm. NE_{wr} in the all the tables presented throughout Section 5 indicates initial Nash Equilibrium found in wide range search in EAA.

<< Insert Fig. 2 >>

<< Insert Table 7 >>

Hence from the above results we can observe that the relative error that we get through EAA is negligible at less than 0.05 per cent.

5.2 Non-linear demand

In this section, we present numerical examples corresponding to each scenario discussed in Section 2.2. The first scenario refers to exponential inverse demand function (Martin, 2002). It considers a duopoly situation whose payoff functions are of the forms shown in equation (8) with the parameters $\alpha_1 = \alpha_2 = 100, \beta_1 = \beta_2 = 10$ and $\gamma=2$. The payoff function of each firm becomes:

$$\pi_1 = q_1 \times 100 \times e^{-\frac{1}{10}\times\left(\sum\limits_{i=1}^{n} q_i\right)^{0.5}}$$

$$\pi_2 = q_2 \times 100 \times e^{-\frac{1}{10}\times\left(\sum\limits_{i=1}^{n} q_i\right)^{0.5}}$$

The solution for this situation is presented in Table 8 and Fig. 3.

<< Insert Fig. 3>>

<< Insert Table 8>>

5.3 Non-linear demand and non-linear cost functions

In this case, a situation has been taken up in which both inverse demand function and cost functions having non-linear natures. In this sense we provide a numerical replica for the model that is being described in Section 2.3 in which the demand functions takes a hyperbolic nature with respect to quantity. This following example is adopted from (Martin, 2002), Payoff function of corresponding firm as shown below.

$$\pi_1(q_1,q_2) = \frac{450 \times q_1}{2 + \sum\limits_{i=1}^{2} q_i} - (430 - 230 \times q_1 + 61.25 \times q_1^{2} - 6.25 \times q_1^{3} + 0.25 \times q_1^{4})$$

$$\pi_2(q_1,q_2) = \frac{450 \times q_2}{2 + \sum\limits_{i=1}^{2} q_i} - (430 - 230 \times q_2 + 61.25 \times q_2^{2} - 6.25 \times q_2^{3} + 0.25 \times q_2^{4})$$

Table 10 and figure 4 shows the computational results for the above the scenario. We have presented more elaborate results than Martin (2002) comprising the different optimal responses found by PSOCP in EAA which are used in plotting BRFs of each firm in Table 10.

<< Insert Fig. 4 >>

<< Insert Table 10 >>

Another non-linear benchmark scenario having multiple Nash equilibriums involving sinusoidal payoff functions (Razi et al., 2007) is presented. The respective payoff functions for that duopoly scenario are as shown below

$$\pi_1(q_1,q_2) = 21 + q_1 \times \sin(\pi \times q_1) + q_1 \times q_2 \times \sin(\pi \times q_2)$$
$$\pi_2(q_1,q_2) = 21 + q_2 \times \sin(\pi \times q_2) + q_1 \times q_2 \times \sin(\pi \times q_1)$$

Fig. 5 and Table 9 show the results of that bench mark scenario depicting all the possible multiple equilibrium (NE). This example reflects the resilience of the EEA algorithm, achieving solutions to complex games smoothly.

<< Insert Fig. 5 >>

<< Insert Table 9 >>

5.4 Constrained functions: Production and Environmental

In this case, a constrained scenario has been considered with linear payoff functions subject to constraints that regulate the quantity of production by each firm and environmental pollution levels. The constraints pertain to the production capacities of firms and environmental regulations that regulate emission of pollutants by both firms. The environmental pollution constraints are as described in Section 2.4.

Pay off functions of individual firm:

$$\pi_1(q_1,q_2) \;=\; q_1 \times (3 - 0.01(q_1 + q_2)) - (0.10 + 0.01 \times q_1)$$
$$\pi_2(q_1,q_2) \;=\; q_2 \times (3 - 0.01(q_1 + q_2)) - (0.12 + 0.05 \times q_2)$$

Subject to:

$$\left.\begin{array}{l} q_1 \le 20 \\ q_2 \le 10 \end{array}\right\} \text{ Production Capacity Constraints}$$

$$\left.\begin{array}{l} 0.5 * 6.5 * q_1 + 0.25 * 5 * q_2 \;\le\; 100 \\ 0.5 * 4.583 * q_1 + 0.25 * 6.250 * q_2 \;\le\; 100 \\ 3.25 * q_1 + 1.25 * q_2 \;\le\; 100 \\ 2.2915 * q_1 + 1.5625 * q_2 \;\le\; 100 \end{array}\right\} \text{ Environmental Pollution Constraints}$$

$$\left.\begin{array}{l} q_1 \ge 0 \\ q_2 \ge 0 \end{array}\right\} \text{ Non Zero Constraints}$$

Results for the above have been presented in three stages with respect to constraints. The first stage deals with effect of production capacity constraints, second stage deals with effect of pollution constraints and the last stage deals with combined effect on NE. The following figures 11, 12, 13 and tables 6, 7, 8 shows the results in sequence of stages.

<< Insert Fig. 6 >>

<< Insert Fig. 7 >>

<< Insert Fig. 8 >>

<< Insert Table 11 >>

<< Insert Table 12 >>

<< Insert Table 13 >>

Another constrained scenario can be built by considering the nonlinear payoff functions for each firm subject to the same set of constraints mentioned in Section 5.5. Consider the payoff functions with hyperbolic demand function and a polynomial cost function presented in Section 5.4 subject to the same constraints presented in Section 5.5. The following figure carries the results of this constrained non-linear scenario. We can observe that there is leftward shift in the Nash Equilibrium when we move from the unconstrained scenario in Section 5.4 to current constrained scenario. There does not exist multiple equilibrium in constrained scenario compared to unconstrained scenario. In the figure 9, BRF-1, BRF-2 represents best response functions in constrained scenario and BRF-1UC, BRF-2UC represents best response curves of unconstrained scenario. NE-UC indicates Nash equilibrium in the unconstrained scenario, NE-C indicates Nash equilibrium in the constrained scenario.

<< Insert Fig. 9>>

A comparison of the above test have been made with the possible exact solutions obtained through traditional analytical approach and these are presented in Table 14. Except scenario 1 and 2, it is difficult to find analytical solutions for the remaining scenarios. Hence no comparison could be done for those scenarios. In fact this shows the value of the proposed EAA as it can handle without difficulty complicated scenarios that cannot be solved using an analytical approach.

<< Insert Table 14 >>

6. Conclusion

This paper used an evolutionary algorithmic approach to determine the Nash equilibrium for situations of duopolistic competition involving nonlinearities, constraints and discrete functions. This is a novel way of determining the Nash equilibrium and a novel application of particle swarm optimization. The model was validated by comparing the results found using the composite particle swarm approach to results determined analytically for a variety of different cases. Closed form analytical solutions however can be found only for industrial competition problems specified for relatively simple, usually linear, formats. The power of the composite particle swarm approach is that solutions for the Nash equilibrium can be found for complex non-linear situations. Examples of non-linear competitive situations abound in the real world, for example where inverse demand functions are concave rather than linear, where firm costs demonstrate economies or diseconomies of scale, or where production capacity is constrained or formulated in discrete bands. A limitation of this research is that EAA has been tested using a limited number of cases and Nash equilibria determined only for pure strategies. Further research work may be carried out to extend the variety of cases examined, to consider more realistic competitive situations, and to determine Nash equilibrium in mixed strategies.

References

Agiza, H. N., & Elsadany, a. a. (2004). Chaotic dynamics in nonlinear duopoly game with heterogeneous players. *Applied Mathematics and Computation*, *149*(3), 843–860. doi:10.1016/S0096-3003(03)00190-5

Al-Awami, A. T., Zerguine, A., Cheded, L., Zidouri, A., & Saif, W. (2011). A new modified particle swarm optimization algorithm for adaptive equalization. *Digital Signal Processing*, *21*(2), 195–207. doi:10.1016/j.dsp.2010.05.001

Cheng, S., Reeves, D. M., Vorobeychik, Y., & Wellman, M. P. (2004). Notes on Equilibria in Symmetric Games. *Research Collection School of Information Systems*, (1995), 1–5.

Chern, M. S., Chan, Y. L., Teng, J. T., & Goyal, S. K. (2014). Nash equilibrium solution in a vendor-buyer supply chain model with permissible delay in payments. *Computers and Industrial Engineering*, *70*, 116–123. doi:10.1016/j.cie.2014.01.013

Daskalkis, C. & Goldberg, P.W. (2009). The Complexity of computing a Nash Equilibrium. *Society for Industrial and Applied Mathematics*, *39*(1), 195–259.

Fabrikant, A., Papadimitriou, C., & Berkeley, U. C. (2004). The Complexity of Pure Nash Equilibria. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computingthe thirty-sixth annual ACM symposium on Theory of computing* (pp. 604–612).

Grant Schoenebeck, S. V. (2006). The Computational Complexity of Nash Equilibria in Concisely Represented Games. In *Proceedings of the 7th ACM conference on Electronic commerce* (pp. 270–279).

Haurie, A., & Krawczyk, J. B. (1997). Optimal charges on river effluent from lumped and distributed sources. *Environmental Modeling and Assessment*, *2*, 177–189.

Kemmoé Tchomté, S., & Gourgand, M. (2009). Particle swarm optimization: A study of particle displacement for solving continuous and combinatorial optimization problems.

International Journal of Production Economics, *121*(1), 57–67. doi:10.1016/j.ijpe.2008.03.015

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of IEEE Conference on Neural Networks* (Vol. 4, pp. 1942–1948).

Konak, A., Kulturel-Konak, S., & Snyder, L. V. (2015). A game-theoretic genetic algorithm for the reliable server assignment problem under attacks. *Computers and Industrial Engineering*, *85*, 73–85. doi:10.1016/j.cie.2015.02.028

Liu, L., Yang, S., & Wang, D. (2010). Particle Swarm Optimization With Composite Particles in Dynamic Environments. *IEEE TRANSACTION SYSTEMS, MAN, AND CYBERNETICS*, *40*(6), 1634–1648.

Ma, C. (2011). A regularized smoothing Newton method for solving the symmetric cone complementarity problem. *Mathematical and Computer Modelling*, *54*(9-10), 2515–2527. doi:10.1016/j.mcm.2011.06.013

Martin, S. (2002). *Advanced Industrial Economics*. Blackwell Publishers.

Maskin, E. (1986). The in Existence of Equilibrium Economic Discontinuous Economic Games, I: Theory. *The Review of Economic Studies*, *53*(1), 1–26.

Mathiesen, L. (2008). Regulation of pollution in a Cournot equilibrium. *SSRN Electroinics Journal*, *1*(June), 1–19. doi:10.2139/ssrn.1552006

Nash, J. (1951). Non-Cooperative Games. *Annals of Mathematics*, *54*(2), 286–295.

Nash, J. F. (1950). Equilibrium Points in n-Person Games. In *Proceedings of the National Academy of Sciences of the United States of America* (Vol. 36, pp. 48–49).

Nearchou, A. C. (2011). Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization. *International Journal of Production Economics*, *129*(2), 242–250. doi:10.1016/j.ijpe.2010.10.016

Nezami, O. M., Bahrampour, A., & Jamshidlou, P. (2013). Dynamic Diversity Enhancement in Particle Swarm Optimization (DDEPSO) Algorithm for Preventing from Premature Convergence. *Procedia Computer Science*, *24*, 54–65. doi:10.1016/j.procs.2013.10.027

Razi, K., Shahri, S. H., & Kian, A. R. (2007). Finding Nash Equilibrium Point of Nonlinear Non-cooperative Games Using Coevolutionary Strategies. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)* (pp. 875–882). Ieee. doi:10.1109/ISDA.2007.41

Tsai, C.-Y., & Yeh, S.-W. (2008). A multiple objective particle swarm optimization approach for inventory classification. *International Journal of Production Economics*, *114*(2), 656–666. doi:10.1016/j.ijpe.2008.02.017

Wang, Z. X. (2013). An optimized Nash nonlinear grey Bernoulli model for forecasting the main economic indices of high technology enterprises in China. *Computers and Industrial Engineering*, *64*(3), 780–787. doi:10.1016/j.cie.2012.12.010

Xiao, T., Jin, J., Chen, G., Shi, J., & Xie, M. (2010). Ordering, wholesale pricing and lead-time decisions in a three-stage supply chain under demand uncertainty. *Computers and Industrial Engineering*, *59*(4), 840–852. doi:10.1016/j.cie.2010.08.011

Yeniay, Ö. (2005). Penalty function methods for Constrained optimization with Genetic Algorithms. *Mathematical and Computational Applications*, *10*(1), 45–56.

Zhang, J.-R., Zhang, J., Lok, T.-M., & Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, *185*(2), 1026–1037. doi:10.1016/j.amc.2006.07.025

Zhang, L., & Zhou, Y. (2012). A new approach to supply chain network equilibrium models. *Computers & Industrial Engineering*, *63*(1), 82–88. doi:10.1016/j.cie.2011.10.007

Zhao, L., & Nagurney, A. (2008). A network equilibrium framework for Internet advertising: Models, qualitative analysis, and algorithms. *European Journal of Operational Research*, *187*, 456–472. doi:10.1016/j.ejor.2007.03.038