Abstract—This paper examines a simulation of the Beer Distribution Game and a number of optimisation approaches to this game. This well known game was developed at MIT in the 1960s and has been widely used to educate graduate students and business managers on the dynamics of supply chains. This game offers a complex simulation environment involving multidimensional constrained parameters. In this research we have examined a traditional genetic algorithm approach to optimising this game, while also for the first time examining a particle swarm optimisation approach. Optimisation is used to determine the best ordering policies across an entire supply chain. This paper will present experimental results for four complex customer demand patterns. We will examine the efficacy of our optimisation approaches and analyse the implications of the results on the Beer Distribution Game. Our experimental results outline our optimisation approaches and analyse the implications of the results on the Beer Distribution Game. We will outline a direct comparison of these results, and present a series of conclusions relating to the Beer Distribution Game.

I. Introduction

This paper examines the well known Beer Distribution Game (BDG), which is a commonly used supply chain analysis tool. This simple game has been widely used for almost five decades to illustrate human decision making and the concepts of supply chain management [1], [2]. The traditional game is normally played by four players, representing four individuals, a retailer, a wholesaler, a distributor and a manufacturer. Each individual faces a decision making challenge involving how they manage their current stock inventories. Each participant in the game seeks to minimise their total cost by managing their inventories in the face of uncertain demand. It has been shown that this simple game provides complex and often non-linear dynamics due to feedbacks and time delays. It has been shown through simulation and also real life experiments that game participants find it extremely difficult to perform well in this game. Their decisions commonly result in large divergences which are far from optimal behavior. These result in large oscillations, deterministic chaos and other forms of complex behavior [3].

The primary goal of this paper is to examine optimal solutions for the BDG. This becomes increasingly difficult when more complex customer demand patterns are examined. We define optimality as the best set of strategies for the overall efficiency of the supply chain, thereby, minimising the total costs for all participants of the game. Determining the most optimal ordering policy is always a difficult problem for any participant in the Beer Distribution Game. A number of possible approaches will be examined in this paper. We will present a particle swarm optimisation (PSO) approach. Such PSO approaches have been successfully used in other similar contexts to find optimal solutions, yet, this is the first time it has been applied to the BDG. We will also examine optimal solutions found using a genetic algorithm (GA) approach. This paper differs from existing research which is primarily based on very simple customer demand patterns [1], [4]. This paper will address a number of important research questions:

1) How effective are PSO and GA approaches to optimising the BDG?
2) What are the differences between our PSO approach and other optimisation approaches to this problem?
3) What are the effects of more complex customer demand patterns on the results?
4) What are the differences between common and distinct ordering policies among game participants?

These research questions will be referred to regularly throughout this paper and answered directly in the Conclusions section. The following sections of this paper are structured as follows. In Section II, we will discuss background research and in Section III we will outline our experimental setup. Section IV will provide a detailed examination of our experimental results. In Section V we will outline our conclusions, while finally in Section VI we will briefly summarise the contributions of this paper and outline some future work.

II. Background Research

Our interpretation of the BDG is based on the specification outlined by Sterman [1]. Later sections will provide a detailed overview of Particle Swarm Optimisation, and Genetic Algorithm search techniques. These sections will also outline previous optimisation approaches to the BDG.

A. Introduction to the Beer Distribution Game

BDG is a classic supply chain optimisation problem [5]. It has been widely used in the domain of supply chain management [5], [6], [7]. This game offers a simplified implementation of common real world production and distribution systems. This system consists of four participants: Retailer, Distributor, Wholesaler and Manufacturer. As shown in Fig. 1, each participant has control and responsibility for its own inventory.

- **Retailer:** The retailer receives orders from customers though the “customer demand”. Subsequently the Retailer must order beer from the Wholesaler to replenish
This phenomenon is known as the bullwhip effect [8]. Delays are an intrinsic part of the BDG. Their presence causes significant challenges in attempting to maintain optimal levels of inventory. Each week, every participant must decide how much to order from its respective supplier or how much to brew in order to meet current and future demands. These processes are affected by a number of delays. Delays represent shipping delays and order receiving delays. These involve the time it takes to receive, process, ship and deliver orders. In the case of the manufacturer these delays represent the amount of time required to replenish its inventory. These delays are an intrinsic part of the BDG. Their presence causes significant challenges in attempting to maintain optimal levels of inventory. Each week, every participant must decide how much to order from its respective supplier or how much to brew in order to meet current and future demands. It is common for each participant to order more (or less) than actually needed due to instabilities in the supply chain. Furthermore, demand information is often distorted in the supply chain from the end customer to the manufacturer. This phenomenon is known as the bullwhip effect [8].

As we have stated previously, the objective of participants is to minimise cumulative costs over a 150 week period by keeping inventories as low as possible while avoiding out-of-inventory conditions which cause backlogs. The BDG commonly uses the following costs to penalise inventory holding and backlogs. The cost of inventory holding is $0.5 per case of beer per week and the cost of backlogs is $2.0 for each case of beer per week. It is intuitive for a player to order more beer when inventory falls below a desired level. Similarly a player is likely to order less beer when stocks begin to accumulate.

B. The Beer Distribution Game Ordering Policy

Since all the game participants experience time delays, it becomes quite difficult to manage their inventory levels. To deal with this challenge, Sterman proposes that the participants adopt three ordering principles [1]. Sterman’s ordering principles are as follows, and state that game participants should place sufficient orders to:

- **Satisfy Expected Demand**: The participants should order enough to satisfy the demand. However, to predict the exact future customer demand is often a difficult task. In this paper, we use an “adaptive expectations” formulation (see Equation (3)) to model future customer demand.
- **Adjust Inventory Levels**: There is a chance of prediction errors from the previous principle. Therefore, it is necessary to adjust orders above or below the expected orders. This serves to correct actual inventory levels in line with desired inventory levels.
- **Adjust for Orders Currently in the Supply Line**: Orders currently in the supply line should be factored into future ordering decisions. Therefore, a participant should be capable of ensuring a stable response to rapid changes in customer demand. It is pointless to place orders for items already ordered in a previous time step.

The following equations formalise an intuitive ordering policy based on the principles discussed above. We will refer to this as the “Stock Management Structure” (SMS). First, orders must be nonnegative:

$$OP_t = \max(0, OP_t^*)$$ (1)

In Equation (1), $OP_t$ represents the actual orders placed at week $t$, whereas $OP_t^*$ represents the orders calculated through the “ordering heuristic” at week $t$. This is defined as follows:

$$OP_t^* = ED_t + \alpha(INV_t^* - \beta * OSL)$$ (2)

In Equation (2), $ED_t$ represents the expected demand at week $t$ (see Equation (3)), $INV_t^*$ represents the discrepancy between desired and actual inventory at week $t$ (see Equation (4)), $OSL$ represents the orders in the supply line; $\alpha$ and $\beta$ represent the adjustment parameters for the inventory and the supply line, respectively; The expected demand ($ED_t$) in Equation (2) is expressed as:

$$ED_t = \theta * CD_{t-1} + (1 - \theta) * ED_{t-1}$$ (3)
In Equation (3), \( ED_t \) and \( ED_{t-1} \) are the expected demand at week \( t \) and \( t-1 \); \( CD_{t-1} \) is the customer demand at week \( t-1 \), \( \theta \) describes the rate at which the demand expectations are updated. The parameter \( \theta \) is typically set to 0.25 [1]. The discrepancy between desired and actual inventory (\( INV_t^j \)) in Equation (2) is formulated as follow:

\[
INV_t^j = Q - INV_j + BL_t
\] (4)

In Equation (4), \( Q \) represents the desired inventory level, \( INV_t^j \) represents the current inventory and \( BL_t \) represents the backlogs at week \( t \). It is clear from Equation (2), that the SMS comprises of three factors: expected demand, inventory level and orders in the supply line. In the BDG, Equation (1) is used to determine how much beer to produce in the case of the manufacturer. Furthermore, in all other cases Equation (1) is used to determine how much beer to order from one’s respective supplier.

It has been proposed that two parameters are used by the game participants to determine their orders [9].

(a) This represents the discrepancy between desired and actual inventory ordered. This parameter is usually represented in the range. \( 0 \leq \alpha \leq 1 \)

(b) This represents the fraction of the supply line taken into account. This parameter is usually represented in the range. \( 0 \leq \beta \leq 1 \) If \( \beta = 1 \), the participant factors in all orders in the supply line or conversely, if \( \beta = 0 \), the participant factors in no orders in the supply line.

The combination of the adjustment parameters (\( \alpha, \beta \)) corresponds to a set of behaviours for a given participant. These parameters are fundamental to our analysis of the BDG. They will provide the main basis for our PSO and GA optimisation approaches. The following two experimental scenarios will be examined in this paper.

- **Scenario A**: all participants in the supply chain use the same ordering policy (the same \( \alpha \) and \( \beta \) values). Therefore there are the only two parameters involved. We will refer to this scenario as \( S_a \).

- **Scenario B**: each participant uses different ordering policies (different \( \alpha \) and \( \beta \) values). Therefore there are eight parameters involved as there are four individuals in the supply chain and each of these has two strategies parameters (\( \alpha \) and \( \beta \)). We will refer to this scenario as \( S_b \).

C. The Beer Distribution Game Objective Function

In the BDG, all the participants wish to minimize their total cost. This objective can be formulated as:

\[
totalCost = \sum_{i=1}^{m} \left( \sum_{j=1}^{n} (2.0 \cdot BL_j^i) + 0.5INV_j^i \right)
\] (5)

In Equation (5), \( m \) is the total number of weeks and \( n \) is the total number of participants, in our case \( n = 4 \). The cost of holding inventory is 2.0 while the cost of maintaining a backlog is 0.5. \( BL_j^i \) is the \( j^{th} \) participant’s backlog at week \( i \); \( INV_j^i \) is the \( j^{th} \) participant’s inventory at week \( i \).

We use equation (5) as the objective function and employ both our PSO and GA to determine the optimal \( \alpha \) and \( \beta \) for the entire supply chain.

D. Particle Swarm Optimization

Particle swarm optimization (PSO) was first described in 1995 by Kennedy and Eberhart [10], [11]. This approach was first inspired by the phenomena of bird flocking and fish schooling. It has been successfully applied to many problems including the economic dispatch [12], and also reactive power and voltage control [13]. However, PSO has not been widely used in the case of supply chain problems.

For each particle {
Initialise it in the search space (it must satisfy all the constraints);
}
Do {
For each particle {
Evaluate its position according the objective function;
If the current position is better than the previous personal best position (pBest), then update pBest;
Choose the particle with the best position of all the particles according the objective value as the gBest;
Update its position according the equation(6);
Check if its position is still in the search space. If not, reinitialise a new particle that satisfies the constraints;
} while it does not reach a stop criteria(i.e. maximum iterations)
}

Fig. 2. The Modified PSO

In PSO, there is a population of solutions referred to as particles. Particles fly around the D-dimensional solution space, and are evaluated according to a fitness criteria after each iteration. The \( i \)-th particle’s position is represented as \( X_t = (x_{11}, x_{12}, \ldots, x_{1d}, \ldots, x_{nD}) \). The flying velocity for a particle \( i \) is represented as \( V_t = (v_{11}, v_{12}, \ldots, v_{1d}, \ldots, v_{nD}) \). In every iteration, each particle’s flying velocity is updated according to the following two positions. The first one is the position at which its best fitness has achieved so far. This position is a “personal best” position and recorded as \( pBest \). The second position is the best position obtained so far by all particles in the population (or by its local neighborhood, in the local version of PSO). This position is a “global best position” (or local best) and called \( gBest \). After
E. The modified PSO

Particle Swarm Optimisation has been shown to be very effective for many optimisation problems. For the BDG problem, however, the PSO needs to be modified to deal with the game constraints.

Our modification involves reinitialising unfeasible particles. Therefore in each iteration, once a particle fails to satisfy the game constraints, it is reinitialised and positioned randomly in the valid search space. Therefore, this particle will again satisfy all the game constraints while also retaining its memory of (a) its personal best position to date pBest, and (b) the location of the global best position gBest. This modification only concerns those particles which break the game constraints, therefore all other particles are updated as normal as in the traditional PSO.

This modification is similar to that proposed by Hu and Eberhart [15] which was applied to constrained non-linear optimisation problems. Our modification benefits the overall diversity of particles, and maintains a good search capability since the total number of particles does not reduce. An overview of the modified PSO is shown in Fig. 2\(^1\).

F. Genetic Algorithms

Genetic Algorithms (GAs) are inspired by evolutionary biology such as selection, crossover (also called recombination) and mutation. GAs have been successfully applied to optimisation problems like wire routing, scheduling, adaptive control, supply chain management, etc [16].

An implementation of a GA begins with a population of chromosomes that encode candidate solutions to a problem. For each generation \( G \), each chromosome is evaluated and assigned a fitness value. After evaluation, Selection is applied to the population according their fitness value and an intermediate population is created. The algorithm applies recombination and mutation to create the next population in \( G + 1 \). Once evaluation, selection, recombination and mutation is completed, this new population is then used and the process repeats over successive generations. This process continues for a predefined period of time (\( G \)), or until the most optimal solutions are found.

Previous research has applied a GA to the BDG problem. O’Donnell et al. [17] and Lu et al. [18] have successfully used GAs to reduce bullwhip effect for the BDG. Strozzi et al. [4] also used a GA to optimise the BDG ordering policies for an entire supply chain. However, these papers examined simple customer demand patterns. In this paper, we will examine a series of more complex demand patterns and investigate the implications of these demand patterns. Furthermore, we will use a real-coded GA which Wright has claimed should increase efficiency and precision [19]. The previous GA approaches mentioned here use binary encoded GAs which differs from our approach. Our real-coded GA uses the Java Genetic Algorithms Package (JGAP) [20].

III. Experimental Setup

Our experiments examine a number of customer demand patterns. These demand patterns use a common mean value which is set to 8. Then we can compare the participants’ behaviour when they use the best policies obtained from PSO or GA. As has been outlined by Yan and Woo, we can model a number of alternative customer demand patterns [21]. These demand patterns are also illustrated in Fig. 3, and less than 150 data points are used for clarity purpose. These demand patterns are as follows:

- **One Step Demand (OSD):** This demand changes only once in the period of this simulation. It is typically set that the customer demand is four cases of beer per week until week 4 and then steps to eight cases of beer per week.
- **Stationary Demand (SD):** The mean demand remains constant over time. The distribution of the demand conforms to a normal distribution. We use a mean of 8 and the standard deviation of 2.
- **Uniform Demand (UD):** This demand fluctuates randomly and is generated using a uniform distribution in the range of \([0, 16]\).
- **Cyclic Demand (CD):** This pattern varies cyclically over time, usually in response to some seasonal effect of season or the standard business cycle. The mean value of demand changes periodically. The cycle of the demand is 50 weeks. In every cycle of the first 25 weeks, we use normal distribution with the mean of 10 and the deviation of 2; In the following 25 weeks, we use normal distribution with the mean of 6 and the deviation of 2.

As explained we will examine four distinct customer demand patterns. The One Step Demand pattern is the most commonly used demand pattern when analysing the BDG. These other three demand patterns are more realistic and reflect a closer representation of real market dynamics. For example, demand for goods such as rice normally follows the SD pattern, demand for Christmas ornaments normally follows the CD pattern. New product demand normally follows the UD pattern.

As mentioned previously we will examine two distinct experimental scenarios. Scenario A: Where all individuals use the same \( \alpha \) and \( \beta \) strategies. We will then examine Scenario B: Where all individuals can use different \( \alpha \) and \( \beta \) values. These two distinct scenarios will be used to compare the performance of our PSO and GA solutions. These scenarios

\(^1\)Please note that our modification is shown in italic bold.
fundamentally change the optimisation problem involved. In Scenario A, all individuals use the same ordering policy and therefore only two parameters must be optimised. In Scenario B, four individuals use different policies and therefore eight parameters need to be optimised.

In the case of all these experiments, optimality is considered as the collective optimal solution for the entire supply chain. This differs from using individual optimality for an individual participant in the BDG. For example, the most optimal solution for the Distributor may well be different, if we were to assume he is acting in a selfish manner. However, in this case the collective performance of the supply chain among all its participants is our fundamental concern. Therefore we define the most optimal solution as “those parameters which provide the most optimal performance across the entire supply chain”. This optimal policy is formally specified through the minimum cost as calculated through Equation (5).

A. Beer Distribution Game Parameters

The parameters involved in our BDG are as follows. The total simulation length was 150 weeks. The \( \alpha \) and \( \beta \) values exist in the range of \([0, 1]\). The \( \theta \) value used was 0.25 (see Equation (3)). Delays equal 4 weeks and the desired inventory levels, \( Q = 17 \).

B. PSO Parameters

Our PSO implementation is based on an existing PSO implementation by Hu and Eberhart [15]. The population size used was \( N = 50 \) and the total number of iterations was 500. The inertia weight was \( w = 0.5 + \text{rand}()/2.0 \), while the maximum allowed velocity \( V_{\text{max}} \) was set equal to range of the dimension. Each dimension was set to \( V_{\text{max}} = 1 \), since the \( \alpha \) and \( \beta \) values were limited to the range of \([0, 1]\). Finally the constants \( c_1 \) and \( c_2 \) were set to 1.49445.

C. GA Parameters

As mentioned earlier, our real-coded GA was implemented using JGAP [20]. The population size was set to \( N = 50 \) and the total number of iterations (Generations) used was \( = 500 \). The crossover rate was 0.8 and the mutation rate was 0.01.

### TABLE I

<table>
<thead>
<tr>
<th>Demand Type</th>
<th>Method</th>
<th>Worst</th>
<th>Best</th>
<th>( \mu )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSD</td>
<td>PSO</td>
<td>1854.0</td>
<td>1772.0</td>
<td>1834.6</td>
<td>27.5</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>1896.0</td>
<td>1791.0</td>
<td>1847.0</td>
<td>28.6</td>
</tr>
<tr>
<td>SD</td>
<td>PSO</td>
<td>2527.0</td>
<td>2459.0</td>
<td>2472.8</td>
<td>15.0</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>2765.0</td>
<td>2593.0</td>
<td>2549.19</td>
<td>61.5</td>
</tr>
<tr>
<td>UD</td>
<td>PSO</td>
<td>7409.5</td>
<td>7488.0</td>
<td>7501.2</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>8041.5</td>
<td>7988.0</td>
<td>7965.4</td>
<td>147.7</td>
</tr>
<tr>
<td>CD</td>
<td>PSO</td>
<td>13533.0</td>
<td>12015.0</td>
<td>12579.2</td>
<td>700.4</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>13702.5</td>
<td>12020.0</td>
<td>12946.9</td>
<td>727.2</td>
</tr>
</tbody>
</table>

IV. Experimental Results

In this section we will provide a series of detailed experimental results, involving the BDG, and our different optimisation approaches.
A. Simulation Results in Scenario A

The data in Table I shows the results obtained by our modified PSO and GA. These results are from 50 individual runs of our PSO and GA. Data is shown representing the maximum cost (Worst), the minimum cost (Best), the average cost ($\mu$) and standard deviation ($\sigma$) for 50 individual runs. Compare the results obtained by PSO and GA, we observe that the results from the PSO are marginally better than the results from the GA. Only two best results are the same when the participants face the SD and UD. So it seems that PSO provides a better solution for the BDG in $S_a$.

In order to clarify our results from $S_a$, we compare the results obtained from using our PSO and GA implementations with the results we have obtained from a bruteforce search of the $\alpha$, $\beta$ strategy space. This corresponds to an enumeration method, and therefore provides us with the optimal solutions for this problem. Here, we used a search resolution of (0.0001) in $S_a$. This resolution is high enough to obtain the global optimum due to the fact that there were no further improvements when higher resolutions were used. However, the bruteforce method was not feasible in $S_b$ due to the size of the search space. Table II shows the optimal results from the enumeration method, the best results from the PSO and GA and the number of hits to the optimal in 50 individual runs of the PSO and GA which we record as (No. PSO) and (No. GA). From Table II, we can see that the PSO find the optimal results in all customer demands while GA does not find all the optimal results. However, the results obtained by GA are very close to the optimal when the participants face the OSD and CD that GA does not obtain the optimal results. Furthermore, the PSO hits the optimal much more times than the GA. It should be noted that these differences do not indicate a clear benefit for using a PSO over a GA approach. Instead they clearly show the differences in the specific case and for the specific parameters used in our PSO and GA algorithms.

Our results in Table I and II show that the most optimal strategies differ significantly when alternative customer demand patterns are used. This means that the demand patterns have a significant effect on the overall performance of the supply chain. Total costs vary significantly across the various customer demand patterns. We believe this correlates strongly with the degree of complexity involved in interpreting each of these demand patterns. It is plausible to argue that where complexity is represented as $C_{OSD} < C_{SD} < C_{UD}$. It appears that Cyclical Demand (CD) is more complex, or more difficult to interpret than Uniform Demand (UD). While the standard deviation of the actual customer demand in each of these patterns would lead us to believe the converse, the actual amplification effects on inventory levels is the primary driver of this behaviour [21]. Furthermore, this problem is influenced through our optimisation approaches “overfitting” for a particular period of the cycle in (CD). This does not occur when interpreting the Uniform Demand customer demand. This effect is reduced when the period used in Cyclical Demand is reduced and therefore CD changes more regularly over shorter periods. We can therefore conclude that $C_{UD} < C_{CD}$. The total cost becomes larger as the demand pattern becomes more complex. So the simulation results show that the more complex the demand pattern, the more difficult to manage the inventory when the participants use the same strategies.

Table III shows the best ordering parameters for all participants obtained by the PSO and GA in 50 runs. The results achieved using both PSO and GA approaches are quite similar. It is clear that the $\alpha$ and $\beta$ values do not follow conventional wisdom as stated by Sterman [1] once more complex customer demand patterns occur. From the values shown in our table it is clear that as uncertainty increases about the customer demand, game participants are better to use lower $\alpha$ and $\beta$ values. Thereby not factoring in all orders in their supply chain, or trying to fully correct their inventory levels. This is a symptom of individuals performing better by not reacting to every small change in demand.

From the results shown and the our discussion above, we have shown that both the PSO and GA approaches achieve high quality solutions for the BDG in $S_a$.

B. Simulation Results in Scenario B

The data in Table IV shows the results obtained from using our modified PSO and GA optimisation approaches. This data represents the maximum cost (Worst), the minimum cost (Best), the average cost ($\mu$) and standard deviation ($\sigma$) over 50 individual runs. The results indicate that the GA was more successful than the PSO in its search for an optimum solution. It should be noted that the difference between the PSO and the GA approaches are not very significant and could be due in part to some of the parameters used in the two approaches. It is clear from analysing the results from both optimisation approaches, that more optimal solutions are achieved in $S_b$ than in $S_a$. Furthermore, it is clear from Table IV that total costs increased as more complex customer

---

**Table II**

Comparison results with enumeration methods in Scenario A

<table>
<thead>
<tr>
<th>Customer Demand</th>
<th>Enumeration Method</th>
<th>PSO</th>
<th>GA</th>
<th>No. PSO</th>
<th>No. GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSD</td>
<td>1772.0</td>
<td>1772.0</td>
<td>1791.0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>SD</td>
<td>2459.0</td>
<td>2459.0</td>
<td>2459.0</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>UD</td>
<td>7488.0</td>
<td>7488.0</td>
<td>7488.0</td>
<td>43</td>
<td>6</td>
</tr>
<tr>
<td>CD</td>
<td>12015.0</td>
<td>12015.0</td>
<td>12020</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table III**

The best $\alpha$ and $\beta$ for all participants in Scenario A

<table>
<thead>
<tr>
<th>Customer Demand</th>
<th>Method</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSD</td>
<td>PSO</td>
<td>0.8855</td>
<td>0.8478</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.8866</td>
<td>0.8470</td>
</tr>
<tr>
<td>SD</td>
<td>PSO</td>
<td>0.2039</td>
<td>0.8132</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.2018</td>
<td>0.8132</td>
</tr>
<tr>
<td>UD</td>
<td>PSO</td>
<td>0.0296</td>
<td>0.7274</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.0290</td>
<td>0.7252</td>
</tr>
<tr>
<td>CD</td>
<td>PSO</td>
<td>0.0249</td>
<td>0.1349</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.0246</td>
<td>0.1323</td>
</tr>
</tbody>
</table>
demand patterns were simulated. This reinforces this same observation which we discussed in $S_a$.

Table V shows the best ordering parameters for each participant as obtained by our approaches in 50 attempts. Clear differences are identifiable in the performance of the two optimisation approaches. These stem from a number of factors. Firstly, the search space is much larger, and more complex due to numerous local minima. Secondly, the specific parameters used by each approach may well influence their ability to find the most optimal solutions in the complex landscape. Apart from these specific issues, it appears quite clear that $S_b$ offers a much more expressive landscape of solutions with greater potential for identifying more optimal solutions than those available in $S_a$. This more expressive landscape is also much more difficult to search, and therefore this causes certain differences in the solutions identified by our optimisation approaches.

C. Comparison Results in both Scenarios

In this section, we will outline the differences and similarities between the results shown from $S_a$ and $S_b$. Table VI clearly shows a reduction in total cost in $S_b$ compared with $S_a$. This stems from the increased $\alpha$, $\beta$ landscape. In $S_a$ these strategy values are constrained so that the same $\alpha$, $\beta$ values are used by all participants in the supply chain. $S_b$ differs significantly, as all individuals are free to use different $\alpha$, $\beta$ values to their peers on the supply chain. This results in a much larger set of possible optimal solutions in the game. This larger, more complex landscape is further augmented when more complex customer demand patterns are simulated. This results in a heightened degree of difficulty when attempting to determine the most optimal $\alpha$, $\beta$ values. This observation is reinforced in the results shown, where it is clear that more complex customer demand patterns result in ever increasing differences between $S_a$ and $S_b$. For more complex demand patterns, $S_b$ offers greater potential to approximate to the global optimum than $S_a$.

### Table IV

<table>
<thead>
<tr>
<th>Customer Demand</th>
<th>Method</th>
<th>Worst $\mu$</th>
<th>Best $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSD</td>
<td>PSO</td>
<td>1664.5</td>
<td>1422.0</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>1560.0</td>
<td>1351.5</td>
</tr>
<tr>
<td>SD</td>
<td>PSO</td>
<td>1866.0</td>
<td>1651.0</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>1837.5</td>
<td>1635.0</td>
</tr>
<tr>
<td>UD</td>
<td>PSO</td>
<td>4874.5</td>
<td>4378.0</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>4537.0</td>
<td>4282.5</td>
</tr>
<tr>
<td>CD</td>
<td>PSO</td>
<td>6976.5</td>
<td>5640.5</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>6644.0</td>
<td>5335.5</td>
</tr>
</tbody>
</table>

V. Conclusions

The research outlined in this paper investigated optimised ordering strategies in the BDG when a number of complex customer demand patterns are simulated. This research holds particular significance for those interested in classic supply chain problems. Earlier in this paper, we posed a number of important research questions. In response to the first research question, both optimisation approaches were clearly very successful in achieving their task. The optimised $\alpha$ and $\beta$ values in $S_a$ and $S_b$, indicate optimal ordering strategies when more complex customer demand patterns occur. While global optima were only found in a limited set of circumstances, it is clear that both optimisation approaches performed well despite the complexity of the initial game, and the added complexity of the customer demand patterns examined in this paper. From examining previous research, and the results outlined here we can confidently conclude that both optimisation approaches performed well in their optimisation tasks.

It is also worth noting that previous research examining this optimisation problem has used genetic algorithm approaches to this problem. Existing research has predominantly examined much simpler customer demand patterns. This paper has extended this existing research to include much more complex customer demand patterns and also introduces for the first time a PSO approach to this optimisation problem. Our second research question refers to this extension. This PSO approach offers a new and alternative approach to these BDG optimisation problems, while specifically in the case of $S_a$ it has been shown to perform very well. It found the optimum solution more often than the GA approach (Table II).

Our third research question refers to the effects of more complex customer demand patterns. It is apparent that as these patterns become more complex the total cost increases as game participants struggle to cope with the increased difficulty. Furthermore, our results show that this increased difficulty also resulted in higher variances across all results. It is clear that this increased complexity posed significant challenges for our optimisation approaches, as they struggled to find optimal $\alpha$ and $\beta$ values. The more complex customer demand patterns used in this research demonstrates a more realistic interpretation of the BDG. Furthermore, the results also show the benefits of low $\alpha$ and $\beta$ values when determining an optimal solution. When customer demand patterns become more complex and difficult to interpret, perhaps its best to simply react rapidly without too much regard for previous trends.

Our final research question involves the use of $S_a$ and $S_b$ in our experimental simulations. As explained previously...
the strategy space in $S_a$ is significantly limited since all individuals must use the same $\alpha$ and $\beta$ values. Conveniently this provided us with a simplified strategy space which was easier to analyse. Furthermore, this also made optimisation much easier. $S_b$ provided a much more complex strategy search space where the $\alpha$ and $\beta$ values were not fixed across all individuals in the game. It is clear that this initially simple game results in some very complex dynamics which are inherently difficult to analyse. These are quite apparent from our results outlined for $S_b$. This complexity is further magnified when more complex customer demand patterns are introduced. Despite this, we identified lower costs in $S_b$ over $S_a$ across all customer demand patterns. This stems from $S_b$ offering greater freedom to individuals and their specific strategy choice. This effect is further magnified when more complex customer demand patterns are encountered.

VI. Summary and Future Research

This paper has examined the BDG, its optimisation and the effects of complex demand patterns. A number of fundamental factors influence this study. Firstly, the degree of autonomy offered to game participants and the scope of their individual rationality. This has significant implications for this paper. Even in the limited case of this paper, we have seen the effects of reduced and increased freedom to determine $\alpha$ and $\beta$ strategies. The second significant factor influencing this study involves the complexity of customer demand patterns. We have provided an initial study of these factors, yet much more work is required.

Acknowledgment

The authors would like to gratefully acknowledge the continued support of Science Foundation Ireland.

References