Yugoslav Journal of Operations Research
29 (2019), Number 3, 359–373
DOI: https://doi.org/10.2298/YJOR180515016L

VARIABLE NEIGHBOURHOOD SEARCH FOR FINANCIAL DERIVATIVE PROBLEM

Nareyus I LAWRANCE AMALDASS
Brunel University, London, UK
nareyus22@gmail.com

Cormac LUCAS
Brunel University, London, UK
cormac.lucas@brunel.ac.uk

Nenad MLADENOVIC
LAMIH, University of Valenciennes, France and Mathematical Institute, SANU, Belgrade, Serbia
nenad@turing.mi.sanu.ac.rs

Received: May 2018 / Accepted: April 2019

Abstract: We propose a new matching problem for combinatorial optimization in financial markets. The problem studied here has arisen from the financial regulators that collect transaction data across regulated assets classes. Unlike previous matching problems, our focus is to identify any unhedged/unmatched derivative, Contract for Difference (CFD) with its corresponding underlying asset that has been reported to the corresponding component authorities. The underlying asset and CFD transaction contain variables like volume and price. Therefore, we are looking for a combination of underlying asset variables that may hedge/match the equivalent CFD variables. Our aim is to identify unhedged/unmatched CFD's. This problem closely relates to the goal programming problem with variable parameters. In this paper, we construct and implement a variant of Basic Variable Neighborhood Search (BVNS) with our newly constructed local search techniques that performs efficient neighbourhood search to solve these types of problems. Computational results show that the proposed approach achieve good solutions.

Keywords: Goal Programming problem, Contract for Difference, Equity, Metaheuristics, Variable Neighborhood Search.

MSC: 90B06, 90C05, 90C08.

1. INTRODUCTION

A CFD, or Contract for Difference, is a leverage derivative product that allows speculation/trade on price movements of the underlying instruments such as commodities, market indices, shares etc. Initiated by entering into a contract at an opening price of an underlying instrument and betting on whether the price of that instrument increases or decreases. To bet that the price increases means that you would go long by buying the CFD expecting the underlying instrument price to gain in value. If you bet that the price decreases, then you would go short by selling the CFD expecting the underlying instrument price to lose its value. In either case when the contract is closed, depending on the price difference between the open and close of the contract, you make a profit or a loss.

CFD is a leveraging product; in other words, a contract can be bought for a fraction of the market value of the underlying instrument. This fraction can be as small as 1% and the rest is covered by the CFD broker. Even though you have deposited only a fraction of the market value, you are enabled to gain 100% profit or loss when closing the contract.

Figure 1: CFD Structure(ws-alerts.com)

For example, Company A share price is x. If you buy a long CFD (buyer) say N shares of company A, then you pay a fraction (1 %) of $N.x$, with an expectation the share price of company A increases. If the price increases from x to $x + u$, then the seller of the CFD will pay you the price difference $u.N$. However, if the price decreases, you pay the seller at the close of the contract.

The buyer and seller of the CFD will enter into a contract. The buyer of the CFD is generally classified as the counter party (client) and the seller as a broker. In theory, a contract is opened when the broker sells a CFD to the counterparty/client. The broker (seller) is expected to buy or hold the equivalent shares (underlying). Similarly, the contract is closed when the counterparty (buyer) sells back the CFD to the broker. When the broker buys the CFD, the broker can go and sell the equivalent share (underlying) in the market.

Regulators around the world especially in Europe have an objective to ensure the financial market works well and to improve market integrity. European regulation has established market abuse regulations to increase market integrity and investor protection. As a result, monitoring the market across all regulated asset classes has been a key functionality for regulators.

In this paper, we will analyze the CFD derivatives for underlain equity market. We use intraday transaction data from January 2013 until January 2015 that have been reported to one of the key European regulators. The transactions report contains all intraday transactions that have taken place across various trading platforms. Our aim is to establish the corresponding match for CFD with its underlying equity hence, to detect the unhedged CFD with its underlying equity. We have developed two different local search methods and embedded them into BVNS to generate new variants, BVNS-LS- Type1 and BVNS-LS-Type2 to find a better solution.

This paper is organized as follows. In section 2, we review the literature. In sections 3 & 4, we illustrate our data source and the problem with a simple example. We describe the mathematical model presented in section 5. In section 6, we describe the BVNS methodology and our newly constructed local search technique, we further discuss its implementation on our problem. While section 7 discusses the numerical results, the conclusions are in section 8.

2. LITERATURE REVIEW

Financial Market Surveillance has become one of the key aspects among the regulators around the world. Various techniques have been used to detect market abuse behavior in the financial markets. Punnuyamoorthy, et al [1] introduced a hybrid data mining technique for detection of stock price manipulation. They used GA with an Artificial Neural Network technique to classify activities that would have potential manipulation. Pirrong [2] examined the Ferruzi Soy bean episode of 1989 and demonstrated how to detect manipulation in the commodity market. He concluded that the regulation in the US market was complex, confusing, and inefficient in futures and securities a market that has relied on costly preventative measures rather than on post deterrence. Ogut et al [3] investigated the best technique to detect stock price manipulation. They developed a data mining technique (ANN and SVM) and multivariate statistical technique (discriminant analysis, logistic regression) for the Istanbul stock market. They concluded that the performance of the data mining technique in terms of total classification accuracy and sensitivity statistics was better than those of the multivariate techniques. Comerton-Forde et al [4] demonstrated the impact on an equity market by using the close price manipulation cases. They further constructed an index to measure the probability and intensity of closing price manipulation and estimated its classification accuracy. David et al [5] modeled cross border market surveillance activities as service systems interacting in a service oriented economy. The market surveillance activities are described as user or customer driven service value networks. The cases were considered as configuration of value networks and value propositions in which the provider and the customers of the service were assumed to be the regulator. Toumi et al (2015) [6] proposed an efficient method two variants of the variable neighborhood search (VNS) heuristics to solve the (0-1) quadratic knapsack problem. They compared large size instance with 1000 and 2000 binary variables and compared the results with other results in the literature. Pererira al et [7] investigated a test assembly design problem. They solved the problem by implementing various neighborhood and variable neighborhood search methods and found that their results outperformed the results obtained in the previous literature. Puchinger et al [8] proposed Relaxation guided variable neighborhood search, which is based on a general VNS scheme and a new variable neighborhood descent (VND) algorithm. The relaxations are used as the indicator for the potential gains of searching the corresponding neighborhood. The algorithm was tested on multiple dimensional knapsack problems and obtained promising results. Durate et al [9] explored the adaptation of VNS to solve multi-objective combinatorial optimization problems. They described how to design the shake procedures, the improvement methods and acceptance criteria for different VNS algorithms for more than one objective; they validated their proposed design on multi-objective combinatorial problems.

3. PROBLEM EXAMPLE

We illustrate our problem with a simple example. In figure 2, we have 6 trades and 5 CFD's on an equity, which contains volumes and prices. Our aim is to match all the 5 CFD with the trades and list all the unmatched CFDs, if the trades are not matched. Note, once trades are match to a CFD they cannot be reused for any another CFD.

In our illustrative example, we generate an initial solution set of trades with the volume and price. We then use the first and the second neighbourhood structure as our shaking procedure. In figure 3, we can see that volumes 1000 and 3000 are removed and volumes 200 and 100 are added. Similarly, their corresponding prices 5.35 are removed and 5.10 are added. We calculate the trade volume 300 by adding trade volumes 200 and 100 since their mean price is 5.10, which exactly matches the first CFD volume 300 and price 5.10. Hence, the trade volumes 100, 200 and their corresponding price 5.10 will be matched against CFD volume 300 and price 5.10. But our aim is to find the unmatched CFD, hence the matched volume and price for trades are removed from set $\acute{C}$, and the rest of the trade volume and price are considered for the next CFDs.

By repeating the above shaking procedure for each CFD in our example, we get the below solution after the shaking procedure.

TRADE

Volume	Price
1000	5.35
100	5.10
3000	5.35
200	5.10
1500	5.35
200	5.12

CFD

Volume	Price
300	5.10
1100	5.35
1500	5.35
200	5.11
3000	5.45

Figure 2: Example Trade and CFD's

Trade Volume	1000	100	3000	200
Trade Price	5.35	5.10	5.35	5.10

Figure 3: Example

- CFD volume 300 and price 5.10 are matched with two TRADE volume (100 + 200) and its price 5.10.
- CFD volume 1500 and price 5.35 are matched with one TRADE volume 1500 and its price 5.35.
- CFD volume 1100 and price 5.35 are unmatched with any TRADE, even though the price 5.35 matches with the CFD price but the volumes 1000 and 3000 do not match.
- CFD volume 200 and price 5.11 are unmatched with any TRADE, even though the volume 200 matches with the volume of CFD, the price 5.12 doesn't match.
- CFD volume 3000 and price 5.45 are unmatched with any TRADE, even

though the volume 3000 matches with the volume of CFD, the price 5.45 doesn't match.

In order to improve our solution from the shaking procedure, we use our two newly developed algorithms as our local search procedure. We concentrate more on the unmatched CFD's in our local search procedure. We have tuned our local search algorithm to identify the reason of the unmatched CFD's. This reason could be either an over/under volume or over/under price. Further we would match the best possible trades inorder to have the minumum mis-match value for the unmatched CFDs. In our example, Figure 4 represents the list of unmatched CFD's and the best possible trade match that has minimum mis-match value with the reason for the unmatch.

Unmatched CFD

Volume	Price	Reason
1100	5.35	Under Volume by 100
200	5.11	Over Price by 0.01
3000	5.45	Under price by 0.10

Figure 4: Unmatched

- In the above table,CFD volume (1100) is unmatched, the reason is the under volume trade of (1000). Hence the minimum difference is 100 or the match is missed by 100.
- In the above table, CFD price (5.11) is unmatched, the reason is the over price trade of (5.12). Hence the minimum difference is 0.01 or the match is missed by 0.01.
- In the above table, CFD price (5.45) is unmatched, the reason is the under price trade of (5.35). Hence the minimum difference is 0.10 or the match is missed by 0.10.

4. DATA

The data used in this paper are provided by one of the key European regulators. The transaction data are got after the implementation of transaction reporting that has been described in the Markets in Financial Instruments Directive (MiFID). It is an obligation for trading firms to report their trades to their local regulators that have been set out in the office of the journal of the European Union, commission regulation EC No. 1287/2006 (Article 13/Annex 1). Firms must report transactions when they execute a trade that is reportable. The report must contain mandatory details of their transactions by the end of the following business day (T+1) as specified in (Article 13/Annex 1). Transaction reports received from firms are loaded into the transaction monitoring system. The purpose of the transaction reporting is to detect and investigate suspected market abuse, and also to maintain confidence in financial markets and reduce financial crimes.

Our primary analysis is based on intraday transaction data for all the FTSE 100 stocks over the period January 2013 until January 2015 in the UK equity market. For our analysis, we have considered only the CFD transaction data. The transaction data are reported on a stock by stock basis that consists of all the executed trades across multiple regulated platforms and are reported in seconds. Further, we have used the price mode function that would convert the price currency to GBP in case they were reported in a different currency or pence.

5. MATHEMATICAL PROGRAMMING FORMULATION

Here we set out our mathematical model of the CFD matching problem.

Sets

- $C = \{1, 2, \ldots, c, \ldots, n\}$ denotes a set of CFD's,
- $T = \{1, 2, .., i, ., t\}$ denotes a set of trades.

Data
- V_t be the volume of trades.
- P_t be the price of trades.
- V_c be the volume of CFD.
- P_c be the price of CFD.

Variables
- $\dot{V}_c$ be the over volume of CFD.
- $\ddot{V}_c$ be the under volume of CFD.
- $\dot{P}_c$ be the over price of CFD.
- $\ddot{P}_c$ be the under price of CFD.

$$Y_{tc} = \begin{cases} 1 & \text{if trade } t \text{ is used in balancing CFD } c \\ 0 & \text{otherwise.} \end{cases}$$

The complete mathematical programming formulation can be written as:

$$Minimize \sum_{c=1}^{c} (\dot{V}_c + \ddot{V}_c + \dot{P}_c + \ddot{P}_c) \tag{1}$$

subject to

$$\sum_t V_t Y_{tc} = V_c + \dot{V}_c - \ddot{V}_c,\ \forall c \tag{2}$$

$$\sum_t V_t P_t Y_{tc} = V_c(P_c + \dot{P}_c - \ddot{P}_c),\ \forall c \tag{3}$$

$$\sum_c Y_{tc} \leqslant 1,\ \forall t \tag{4}$$

The objective function is to minimize the total CFD mismatch . The constraint group(2) determines the over and under volume.(3) determines the over and under volume with price of the cfd and (4) specific that a trade can be used for at most one CFD.

6. BASIC VARIABLE NEIGHBOURHOOD SEARCH (BVNS)

Variable Neighbourhood Search (VNS) is a metaheuristic algorithm for solving combinatorial and global optimization problems introduced in [10]. The special feature of VNS is the systematic changes of neighbourhoods within the local search to attain better solutions. The VNS is based on three factors. (i) A global solution is a local minimum for all neighbourhood structures, (ii) A local optimal solution in one neighbourhood may not be the optimal solution of another neighbourhood, (iii) Local optimal solutions are relatively close to different neighbourhoods.

There are several variants of VNS that have been used in various combinatorial optimization problems. In our problem, we are using Basic Variable Neighbourhood Search (BVNS)[10]. It uses a process to find the next optimal solution from the most fitting neighbourhood structure and then, the solution is further refined and improved by using a local search technique. This improved solution will be a current solution from the neighbourhood in the iteration. This process will provide a good solution and save computational time without analyzing the full neighbourhood structure.

Our proposed BVNS, initially generates random solution S, then it uses two neighbourhood structures, namely Remove Fill and Add Remove, as a shaking procedure to generate a solution $\dot{s}$ and a local search to improve the shaking solution $\dot{s}$ as the input solution to get a newly improved solution $\ddot{s}$. We then compare $\ddot{s}$ solution with the S in term of the objective function. If there is an improvement, we replace the current solution S with $\ddot{s}$. We define the stopping criteria to be the maximum number of iteration, 500 for shaking and local search.

Let us assume K neighbourhood structures $N_1, N_2, ., N_k$. The process starts with the initial solution S, we obtain the next solution, s, from the neighbourhood such that $N(s) \subset S$. Performing the shaking procedure for local changes in the neighbourhood, we can obtain a better solution $\dot{s}$ from $N(s)$. We perform a local search procedure with different neighbourhoods until a local optimum is obtained.The general working algorithm for BVNS is given below.

Algorithm 1: BVNS

Initialization: select the neighbourhood structure sets $N_S, S = 1, 2, ..., S_{max}$;
Generate a random initial solution s in S;
Set $S = 1$;
Repeat the following steps until $S = S_{max}$;
Shaking: generate a point $\dot{s}$ randomly from $N_s(S)$;
LS: implement Local search method to obtain local optimum $\ddot{s}$ from $\dot{s}$;
if *$\ddot{s}$ is better than $\dot{s}$* **then** set $s = \ddot{s}$ and $s = 1$;
else $s = s + 1$;
stop ;

6.1. Neighbourhood Structure

Initially, we group the trades into various random subsets. We match these subset groups to the CFD. Define C as the set of trades that are considered to match each CFD and $\acute{C}$ is its compliment. We remove and add certain trades to match the CFD in order to obtain the solution s. We use two types of neighbourhood to perform shaking.

- The First neighbourhood, Remove and Fill, we enforce trade volume V_t and trade Price P_t, to be removed from C and by adding V_t, P_t to $\acute{C}$ to get a better match.
- In the second neighbourhood, Add and Remove, we enforce trade volume V_t, trade Price P_t that are not used from $\acute{C}$ to match the CFD by removing a combination of tradesV_t, P_t from C.

6.2. Local Search Neighborhood

We construct two new search algorithms in order to improve the solution. We implement these algorithms as a local search procedure in BVNS. We then compare the results of these two local searchs and report the solutions in the results.

6.3. Local Search Type- 1

In this local search type 1 approach, we improve the solution by matching the trades to the CFDs to generate new solutions that contain various CFD mismatches. We weight these mismatches with a cost function. Later, we start with the worst CFD cost to re-match the CFD with the trades to either attain a better solution or we achieve the same solution. The pseudo code for the Local Search (LS-TYPE-1) is given below.

Algorithm 2: LS-TYPE-1

Initialize Set $T = 0$;
Set $\acute{T} = \infty$;
Set $O_i = i$;
for *each i in CFDs* **do**
  solve(O_i);
  Let $E_{O_i} = Z$;
  $T = T + Z$;
end
if $T \geq \acute{T}$ **then** break ;
else $\acute{T} = T$;
Let $\acute{O}_i$, $\acute{E}_i$ = sort(E_i)and O_i;
Let $O_i = \acute{O}_i$;
Display $\acute{T}$;

In the pseudo code, O_i is the ordering of disaggregated sequence of each CFD's. EO_i is the new seqeunce order. Depending on the CFD mismatch cost, we sort the EO_i to get the maximum CFD mismatch cost sequence $\acute{O}_i$, $\acute{E}_i$. We rematch the trades T to CFD's E_i in order to obtain a better sequence or to minimize the CFD mismatch cost.

Algorithm 3: BVNS-LS-TYPE-1

Initialization: select the neighbourhood structure sets $N_S, S = 1, 2, ..., S_{max}$;
Generate a random initial solution s in S;
Set $S = 1$;
Repeat the following steps until $S = S_{max}$;
Shaking: First, Second neighbourhood structures;
LS: Local Search: Algorithm LS-TYPE 1;
if *$\ddot{s}$ is better than $\dot{s}$* **then** set $s = \ddot{s}$ and $s = 1$;
else $s = s + 1$;
stop ;

6.4. Local Search-Type 2

We reconstruct a different type 2 local search to solve each CFD independently. Not every trades used in matching the CFD is available for the next CFD. If no mismatch is found, we exit, otherwise, we sort the CFD mismatch from the largest to the smallest. This constitutes the new ordering. This process is repeated until either no mismatch is achieved or the current sequence has occurred previously; in which case we know that we have a mismatch and we also know its minimum value.The pseudo code for the type 2 local search is given below.

Algorithm 4: LS-TYPE-2

Initialize Set $c = 0$;
for *each i in* CO_i **do**
    Z= solve(CO_i);
    $CS_i = Z$;
end
if $\sum_{j=1}^{c} CS_i = 0$ **then** break ;
else sort(CS_i, CO_i) ;
for k *in* 1 *to* $c - 1$ **do**
    if $CO_i = OO_k$ **then** break ;
    else sort(CS_i, CO_i) ;
end

In the pseudo code, CO_i is the ordering of disaggregated sequence of each CFD's. Z is the objective function of solving the disaggregated CFD's, if the CS_i is zero, we break and exist the loop. We then sort the CS_i to find the worst cost of the non matched CFD. If there is a match, we break, if not, we repeat the process until we find a better solution. The below psuedo code gives the new VNS variant with LS-TYPE-2:

Algorithm 5: BVNS-LS-TYPE-2

Initialization: select the neighbourhood structure sets $N_S, S = 1, 2, ..., S_{max}$;
Generate a random initial solution s in S;
Set $S = 1$;
Repeat the following steps until $S = S_{max}$;
Shaking: First, Second neighbourhood structures;
LS: Local Search Algorithm LS-TYPE 2;
if $\ddot{s}$ *is better than* $\dot{s}$ **then** set $s = \ddot{s}$ and $s = 1$;
else $s = s + 1$;
stop ;

7. NUMERICAL RESULTS

Our experiments are performed on an intel core i5 processor, 3.20GHZ, windows 7 with 64 bit operating system. We collected the transaction data (section 3) from the European regulator. Clearly, we are only interested in solving the problem for a security with more than one CFD. The reason for including a problem set with one CFD is to validate our results. We deliberately included datasets that contain unhedged/unmatched data and grouped the data to increase the trade size. We used AMPL as our coding language and our intention is to solve these problems using solver. Table 1, below, contains the results which represent the problem size, the total number of trades in an instance, the total number of CFD's, the optimal CFD error cost for each problem instance, and the CPU seconds are given in brackets for BVNS-LS-TYPE-1, BVNS-LS-TYPE-2,cplex .

Problem Instance	Trades	CFD	BVNS-LS-TYPE-1	BVNS-LS-TYPE-2	CPLEX
$new - CFD - 1$	26831	8	0.0249122 (68.569)	0.00444 (39.39)	0.00444 (2278.63)
$new - CFD - 2$	11988	4	0.0084 (25.191)	0.00084 (8.127)	0.00084 (727.511)
$new - CFD - 3$	6622	6	0.067 (18.92865)	0.0622 (13.135)	0.0622 (10687.1)
$new - CFD - 4$	12467	9	0.0417 (48.4523)	0.0415 (68.843)	0.0418 (10803)
$new - CFD - 5$	53647	6	0 (34.32094)	0 (13.712)	0.00025 (10378.1)
$new - CFD - 6$	10432	11	0.0476 (36.9594)	0.00279 (72.228)	0.01016 (10197.5)
$new - CFD - 7$	1132	7	0 (10.5535)	0 (4.992)	0 (10408.2)
$CFD - 9$	598088	4	0.00044 (2973.8)	0.000346 (1258.66)	0.000946 (3859.59)
$CFD - 6$	9209	5	0 (702.8)	0 (702.8)	0.00515 (10659.8)
$CFD - 11$	623	3	0.00485 (12.4937)	0.000485 (7.846)	0.000485 (655.625)

Table 1: Best Solution for CFD-Trades Matching

We have compared the results between two different local search types; we found that type 2 local search is more efficient than type1 local search. In all instances, we have validated our results in the following way. For every problem, we have fixed the binary variables to the values determined by our algorithm and then solved. After, we unfix our assignments and resolve using the previous optimal basis as a warm start for this resolve. For all the problems, these two separate solutions have produced the same results as our local search type 2 algorithm produced. Because of the combinatorial nature of these problems, we decided not to parameterize benchmark results. We report the best solution found by cplex. The stopping criterion is a mixed integer solution limit of 100 and a default time limit. In majority of cases, cplex is unable to find a better solution; in case if a solution is found, the time is extensive.

CFD	Match	Trades
1	Y	11
2	Y	22
3	Y	1
4	Y	7

Table 2: new-CFD-2

CFD	Match	Trades
1	Y	11239
2	N	6578
3	Y	712
4	Y	9987

Table 3: CFD-9

CFD	Match	Trades
1	Y	23
2	N	73
3	N	11
4	N	5
5	Y	1
6	Y	1

Table 4: new-CFD-3

CFD	Match	Trades
1	Y	25
2	Y	16
3	Y	15
4	Y	8
5	Y	19
6	Y	14

Table 5: new-CFD-5

CFD	Match	Trades
1	Y	19
2	Y	22
3	Y	16
4	N	9
5	Y	1
6	Y	17
7	Y	19
8	Y	18

Table 6: new-CFD-1

CFD	Match	Trades
1	Y	10
2	Y	8
3	Y	8
4	Y	11
5	Y	11
6	Y	6
7	Y	2

Table 7: new-CFD-7

CFD	Match	Trades
1	Y	9
2	Y	12
3	N	10
4	Y	9
5	Y	8
6	Y	6
7	N	3
8	Y	17
9	Y	2

Table 8: new-CFD-4

CFD	Match	Trades
1	Y	10
2	Y	10
3	Y	7
4	Y	9
5	Y	2
6	Y	3
7	N	6
8	Y	1
9	Y	10
10	Y	22
11	Y	16

Table 9: new-CFD-6

CFD	Match	Trades
1	Y	14
2	Y	852
3	Y	29
4	Y	50
5	Y	241

Table 10: new-CFD-6

CFD	Match	Trades
1	Y	11
2	Y	6
3	N	5

Table 11: new-CFD-11

TABLES 2,3,4,5,6,7,8,9,10,11 contain the results of individual problem instances. The CFD column represents the number of CFD's to be matched. The match column represents whether the CFD volume and price are matched with its corresponding trades, *Y* indicates the match is successful, and *N* indicates the match is unsuccessful. The trades column represents the total trades used to match the CFD. In case they are unmatched, the trades represent the nearest mismatch of the CFD.

8. CONCLUSION AND FUTURE WORK

We have introduced a neighbourhood structure for shaking and two different local search approaches. We have combined each of these local search types with our shaking neighbourhood and attained two new Variable Neighbourhood Search (VNS) variants for these types of matching problems. We further compared the results of these two search methods by calculating the CFD cost and starting with the worst CFD cost to determine the optimal solution. From our comparison, the type 2 local search approach identifies the most optimal CFD trades match and quickly solves the problem in good computation time. Future work may include development of more advanced VNS based techniques such as skewed general VNS and an advanced metaheuristic hybrid approach.

REFERENCES

[1] Punniyamoorthy, M., and Thoppan, J.J., "ANN-GA based Model for Stock market Surveillance", *Journal of Financial Crime*, 20 (1) (2013) 52-66.

[2] Pirrong, C., "Detecting manipulation in futures markets:: the ferruzzisoybean episode". *American Law and Economics Review*, 6 (1) (2004) 28-71.

[3] Ogut, H., Doganay, M., and Aktas, R., "Detecting stock price manipulation in an emerging market the case of turkey", *Expert Systems with application*, 36 (9) (2009) 11944-11959.

[4] Comerton-Forde, C., and Putnins, T. J., "Measuring of closing price manipulation", *Journal of Financial Intermediation*, 20 (2) (2009) 135-58 .

[5] David, D., Theodoulidisand, B., and Eliza, A., "Cross broder challenges in financial markets monitoring and surveillance. A case study of customer driven service value networks", *Annual SRII Global Conference*, (2012) 146-157.

[6] Toumi, S., Cheikh, M., and Jarboui, B.,"0-1 Quadratic knapsack problem solved with VNS algorithm", *Electronic Notes in Discrete Mathematics*, 47 (2015)269-276.

[7] Pereira, J., and Vila, M., "Variable neighbourhood search heurisitcs for a test assembly design problem", *Expert Systems with application*, 42 (10) (2015) 4805-4817.

[8] Puchinger, J., and Raidi, R.G., "Bringing Order into the Neighbourhood: relaxation guided variable neighbourhood search", *Journal of Heurisitics*, 14 (5) (2008) 457-472 .

[9] Duratem, A., Pantrigom, J. J., Pardo, G. E., and Mladenovic, N., "Multi-objective variable neighbourhood search : an application to combinatorial optimization problems", *Journal of Global Optimization* , 63 (2015) 515-536.

[10] Mladenovic, N., and Hansen, P., "Variable neighborhood Search", *Computers and Operations Research*, 24 (11) (1997) 1097-1100.