Non-Dominant Sorting Firefly Algorithm for Pricing American Option

Gobind Preet Singh, Ruppa K. Thulasiram, Parimala Thulasiraman
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
{gobind, tulsi, thulasir }@cs.umanitoba.ca

Abstract—**An option, a type of a financial derivative, is a contract that creates an opportunity for a market player to avoid risks involved in investing, especially in equities. An investor desires to know the accurate value of an option before entering into a contract to buy/sell the underlying asset (stock). There are various techniques that try to simulate real market conditions in order to price or evaluate an option. However, most of them achieved limited success due to high uncertainty in price behavior of the underlying asset. In this study, we propose a new variant of multi-objective Firefly algorithm to compute the accurate worth of an American option contract and compare the results with the popular option pricing models: Binomial Lattice and Monte-Carlo using the real market data.**

In our study, we have first modelled the option pricing as a multi-objective optimization problem, where we introduced the pay-off and probability of achieving that pay-off as the main optimization objectives. Then, we proposed to use a latest nature-inspired algorithm that uses the bioluminescence of Fireflies to simulate the market conditions. In this paper, we introduce the non-dominant sorting Firefly algorithm to find the Pareto optimal solutions for the option pricing problem. Using our algorithm, we have successfully computed complete Pareto front of option prices for a number of option contracts from real market. Also, we have shown that one of the points on the Pareto front represents the option value within 1-2% error of the real data (Bloomberg).

Moreover, with our experiments, we have shown that any investor may utilize the results in the Pareto fronts for deciding to get into an option contract and can evaluate the worth of a contract tuned to their risk ability. This implies that our proposed multi-objective model and Firefly algorithm could be used in real markets for pricing options at different levels of accuracy. To the best of our knowledge, modelling option pricing problem as a multi-objective optimization problem and using newly developed Firefly algorithm for solving it is unique and novel.

I. Introduction and Background

Today's real-world applications have become more complex, dynamic and require fast response time. In this research, we have focused on one such application: financial option pricing. In financial market, every investor desires rapid information and accurate solution to earn profits. However, problems in finance are too complex to be solved by a deterministic algorithm(s) in reasonable computing time. Therefore, researchers in the field of computational finance are constantly working towards developing efficient models and algorithms to help investors in their decision-making process and provide them with accurate information.

Financial option is a type of derivative contract and is used for a variety of purposes from risk analysis to portfolio management. Every investor in the financial market desires to know the accurate worth of an option contract. However, due to the dynamic and volatile nature of the financial market, pricing them accurately and efficiently is very difficult.

In 1973, Fischer Black and Myron Scholes [1] along with Robert Merton [2] revolutionized the options market by introducing a closed-form solution to price options. However, this Black-Scholes-Merton (BSM) model did not depict the market scenario well due to assumptions such as constant volatility and its restricted applicability to only simple options such as European option which allows expiration only at the expiration date. Following this, Cox-Ross Rubenstein [3] proposed a discrete time approach known as binomial lattice model to value an option in the risk-neutral regime put forth by BSM model. Due to time discretization, this model allowed pricing other styles of options as well, such as an American option (which allows exercising before the expiration date). However, in order to estimate the accurate worth of the option contract using binomial lattice model, we need to do large amount of computation. Therefore, there is always a demand for techniques, which prices option accurately and also within reasonable computing time. There are many other numerical techniques such as Monte-Carlo simulation, Fast Fourier transform (see, for example, [4]–[7]) in the literature to price options. However, these techniques also have some drawbacks and lack in appropriately capturing the real market conditions.

Due to lack of accurate and efficient approaches to price option, researchers started looking for other unconventional technique (s) aiming to capture real market scenario more accurately. One such class of technique is nature-inspired meta-heuristic algorithms [8]. In the recent past, various meta-heuristic algorithms, such as evolutionary and nature-inspired algorithms have been proposed to solve the option pricing problem [9]–[12] and showed satisfactory results. By getting motivated from these studies, we used another latest nature-inspired technique, Firefly algorithm to find solution for option pricing problem. In contrast to previous studies, we have mapped option pricing problem as a multi-objective optimization problem rather than a single objective problem and have developed a multi-objective variant of Firefly algorithm to find its solution.

The rest of this paper is organized as follows: In the next three sections, we briefly describe the option pricing and Firefly algorithm followed by the related work in the general topic of nature-inspired algorithms for financial applications. In section V we explain our proposed strategy to map the Firefly algorithm to option pricing, and describe the design of our algorithm. We implement this algorithm and explain the results of our implementation in section VI along with evaluation and error analysis. We conclude our study in section VII.

II. Financial Options

Financial option is a standard derivative contract that gives a holder the right without any obligation to buy or sell an underlying asset (such as a stock) at a predetermined price (called the strike price) for a specific period of time (contract period with an expiration time). The person who buys the contract is called the buyer/holder of the option and the person who sells the contract is called the writer. Options are categorized into two types: *call* and *put*. With call (put) option, the buyer holds the right to buy (sell) the underlying asset at a specific price during the contract period. Depending on when and how the option is exercised (i.e., style), options can be categorized as either vanilla (European, American) or exotic options (Asian, Russian, etc). In this paper, we focus on the American option that allows the holder to exercise the option anytime before the maturity date.

The goal of option pricing is finding the *worth* of a contract. Five basic parameters that influence the price of an option are: underlying stock price *(S)*, strike price *(K)*, interest rate *(r)*, volatility (σ) and time until expiration *(T)*. The local pay-off or the worth of a contract is computed as max(S-K,0). In the European style of an option, *S* is the price of stock during the expiration where as in American style of an option, *S* is the maximum value of stock during the contract period.

Volatility is the degree of price movement over the contract period measured using the standard deviation of the underlying asset price. Interest rate is the rate of return of an investment with zero risk.

For further description of options and other basic models (such as BSM, binomial lattice model, Monte-Carlo simulation, etc.), the readers may refer to [4], [13].

III. Firefly Algorithm

Firefly algorithm introduced in [8] is inspired by the flashing lights of Fireflies. Flashing light emitted by Fireflies is used as a communication system to attract mating partners or warn potential predators. First signalers are typically males to attract females, which in turn respond by emitting continuous flashing lights. Both the mating partners emit distinct flashing signals that are precisely timed to encode information like sex or species identity.

There are two important issues defined in the Firefly algorithm [8]: variation of light intensity (brightness) and attractiveness. Brightness and attractiveness of Fireflies are proportional to each other. A Firefly moves towards a brighter Firefly. The intensity of the light (I) varies with respect to the distance r. As r increases, the intensity (and attractiveness) decreases and vice versa. The intensity is found to be inversely proportional to the square of the distance $I(r) = I_o e^{-\gamma r^2}$, where I_0 denotes the initial light intensity and γ, the fixed light absorption coefficient. Attractiveness, β is therefore, $\beta = \beta_o e^{-\gamma r^2}$, where β_o is attractiveness at r=0. In addition, we can view the brightness of light intensity as being connected with the objective function. That is, the light intensity is proportional to the solution's fitness value.

In this paper, the distance between two fireflies i and j at x_i and x_j, respectively, is calculated using the Euclidean distance:

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^{k=n} (x_{ik} - x_{jk})^2} \tag{1}$$

where n is the dimensionality of the problem. The position of the i^{th} firefly is updated using the position of the j^{th} firefly, the attractiveness β and a randomization parameter α:

$$x_i = x_i + \beta_o e^{-\gamma r^2}(x_j - x_i) + \alpha \varepsilon_i \tag{2}$$

where ε_i in our algorithm, is a random number obtained from a random number generator of uniform distribution [0,1].

IV. Related Work

Heuristics and meta-heuristics approaches have been used to find solutions for problems in finance such as: determining implied volatility [14], time series forecasting of stock prices [15], or portfolio selection [16]. These problems are too complex to be solved by deterministic or analytic approaches in reasonable time. Hence, non-deterministic approaches like nature-inspired algorithms were tested on these problems and got appropriate results.

In 1994 Hutchinson et al. [17] applied artificial neural network for solving the option pricing problem. Chidambaran et al. [11] used genetic programming to calculate the value of an option. They tried to find the relationship between option price, option contract term and the properties of underlying asset. The authors randomly generated functions (programs) relating the set of input (properties of options contract) and underlying asset with one single output (option price). This approach had one benefit that incorporated previously known formulas such as BSM in its search to find better approximated results. Also, the authors claimed that in most of the trials their algorithm performed better than BSM model. Yin and Brabazon [18] proposed an improved adaptive genetic programming algorithm to price option and claimed that their strategy was more efficient than using fixed genetic programming for option pricing.

One of the most important and continuously studied problems in derivatives market is to find the accurate worth of an option contract. Some of the studies (see for example, [4], [5], [7]) were able to capture the parameters required to evaluate the option contract and to find accurate worth of these contracts. However, most of these techniques incurred high computational cost, or were not able to find accurate solutions for the option contract.

Keber and Schuster [19] used swarm-intelligence technique called generalized ant programming for valuation of American put options for non-dividend paying stocks. They claimed to have found accurate approximation results. Kumar et al. [9] studied the suitability of ant colony optimization (ACO) [20] algorithm for options valuation. The authors developed two new ACO based algorithms for pricing options namely sub-optimal path generation algorithm and dynamic iterative algorithm. They also studied the efficiency of their techniques to price exotic options (Asian or barrier option) and claimed their algorithm performed better than other numerical techniques like binomial lattice and Monte-Carlo simulation. However, due to some limitations with ACO in finding a path rather the best node (time) to exercise the option, the authors in [10] considered another popular and efficient meta-heuristic algorithm, particle swarm optimization (PSO) [21] for pricing options. The authors [10] mapped each particle of PSO to capture both the profit (pay off) and the exercise time. They also incorporated varying volatility in their algorithm to represent real market conditions to price both European and American style options.

In this paper, we model the option-pricing problem as a multi-objective optimization problem, solve the problem using multi-objective Firefly algorithm and validate our results using real market data. To the best of our knowledge this is a unique and novel model for pricing options.

V. Option Pricing: Multi-Objective Optimization Model and Firefly Algorithm

In this section, we model American option pricing as a multi-objective optimization problem and describe the solution methodology used to find it's solution. In an earlier work we reported results from our preliminary model for simple European option pricing, which were quite encouraging. This motivated us to carry on with this work for improving the model that can handle more complicated options such as American option. In section V-A, first we present the mapping of option pricing as multi-objective optimization. Then, in section V-C, we explain the non-dominant sorting Firefly algorithm in detail. Moreover, we not only extend our preliminary model for the current study but also devise a new non-dominated sorting Firefly algorithm to price American options.

A. Mapping Firefly to Option Pricing problem

This is one of the most important contribution of our research. In our previous work [22], we used a similar model for pricing European option and found very promising results. This model is a novel idea and has multiple advantages as compared to other traditional techniques.

We consider an optimization search space for option pricing problem as a 2-dimensional space where each point in search space represents a solution equal to the worth of an option contract (maximum possible profit expected on exercising the contract).

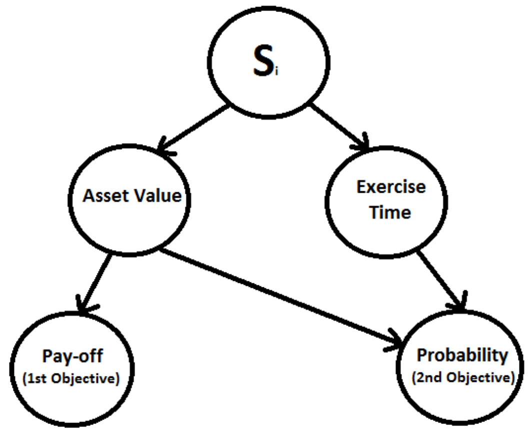

Fig. 1. Mapping

Each solution in the search space consists of two option parameters (dimensions): value of the underlying asset and exercise time. The first dimension, the value of the underlying asset, represents any feasible value that the underlying asset can reach during the life of an option. The second dimension, the exercise time, represents any value between the start of the contract and expiration time.

To capture appropriate behavior and accurate solution, we consider maximum possible pay-off from the contract and probability of attaining that payoff as two main objectives for the problem. Pay-off is the profit earned on exercising the contract and probability is the chance of earning that profit in current market conditions. Pay-off is calculated using max(S-K, 0) where S is the asset price at the time of exercise, K is the strike price and probability is calculated using the algorithm explained in the next section. To the best of our knowledge, none of the nature-inspired techniques in the past have considered probability of achieving a particular pay-off as an objective in pricing options. Therefore, our aim is to maximize the combination of both pay-off and probability in a holistic way accurately price an option. This makes our approach unique. We construct the fitness function of a solution representing maximum profit as function of two optimization objectives: profit (pay-off) and probability of achieving that profit.

$$FitnessValue(F(x)) = F(Payoff, Probability) \tag{3}$$

In this function we aim to maximize both the profit and probability of achieving the pay-off simultaneously, to obtain the accurate worth of the option. This is modelled as a multi-objective optimization problem. Interesting phenomenon to note is that these two objectives behave in such a manner that when probability is high, the rate of change in pay-off is low and vice-versa.

Therefore, to efficiently optimize both the objectives and compute accurate solutions, we designed and developed a novel and efficient multi-objective non-dominant sorting Firefly algorithm to find trade-off between these objectives and compute Pareto optimal solution for American option pricing problem. This is explained further in Algorithm 3 in the following subsection.

B. Probability Computation

Investors would like to get as accurate information as possible on the probability of any underlying asset reaching a particular value in future (during the contract period). However, according to efficient market hypothesis [23], it is assumed that asset prices in financial market follow a random walk, making it very difficult to predict the future value of an asset. In this work, we use Monte-Carlo simulation, to compute the probability of an asset reaching a particular value in future. This is described in Algorithm 1.

Algorithm 1 Probability estimation using Monte-Carlo simulation

1: **Input :** Intial Value(S_o), Target Value(V), Number of Trading Days(T), Volatility (σ), Drift rate(μ)
2: **procedure** PROBCAL
3: Initialize number of Simulation
4: t=0, counter=0
5: **while** $t < NumberofSimulation$ **do**
6: Generate a Normally Distributed Random Number (ε) from range (0,1)
7: Calculate Asset Value using equation $S(T) = S_o e^{(\frac{\sigma}{2})(2(1+\varepsilon\sqrt{T})-\sigma)}$
8: **if** $S(T) \geq V$ **then**
9: Counter = Counter +1
10: **end if**
11: **end while**
12: Probability = Counter / Total Simulations
13: **return** Probability
14: **end procedure**

The input to the algorithm consists of five parameters: initial asset value, target asset value for which probability is to be calculated, number of trading days and volatility. To estimate the probability, we compare the target asset value to the final simulated price from Monte-Carlo simulation (Line 7). Line 8, records the number of times the simulated random walk of an asset exceeds the target asset value. This value is divided by the total number of simulations giving the probability of an asset to reach or exceed the target value (Line 12 and 13). This approach of using Monte-Carlo simulation to compute the approximate value of probability is one of the most popular techniques in the literature [13] and is used by many real practitioners to make decision in their day-to-day trading cycle.

C. Non-dominant sorting Firefly algorithm for pricing American option (NSFA)

In this section we explain our newly designed non-dominated sorting Firefly algorithm (NSFA) to find Pareto optimal solution for multi-objective option pricing problem. This algorithm is inspired from NSGA-II (non-dominant sorting genetic algorithm) [24] in a way that we use non-dominated sorting and crowding distance approach to find Pareto optimal solution for multi-objective option pricing problem. Non-domination sorting is the sorting of the population on the basis of their non-domination and crowding distance metric is a technique used to main the diversity within the population. But as Firefly algorithm is very different from Genetic algorithm, the way these approaches are used is quite different in my aproach.

First, in order to simulate NSFA for option pricing problem, we have to set the initial values for Firefly parameters $(\alpha, \gamma, \beta_o n, t)$ and financial option parameters (S, T, K, σ, r). Then, on the basis of input parameters for financial option, upper bound and lower bound for both the variable S and T of a Firefly is set using equation defined in line 3a and 3b of Algorithm 3. Here, we use volatility to set bounds for the asset price (S). Explanation behind this approach is that we are trying to incorporate the real market condition in our search and volatility is the annualized standard deviation of price movement for an asset. In our implementation we use twice the value of volatility (2*σ) rather than using just volatility, in order to increase the size of the search space.

After initializing all the parameters and variables, we start NSFA simulation by randomly setting values for the initial population, where each Firefly represents asset price (S) and time (T). After setting the initial values, we evaluate the payoff and probability for each Firefly. Further, in line 6 and 7 of Algorithm 3, we do the non-domination sorting (Population sorted according to ascending order of non-domination) [25] of the initial population and evaluate the fitness for each Firefly using equation: $F(x_i) = n - NonDomLevel(x_i)$ in which *n* is the size of population and *NonDomLevel* is the non-domination level of *ith* Firefly. Using this equation all fireflies with better non-domination level will have higher fitness and will have more significant impact during the simulation, since in our implementation we are trying to maximise the fitness (Firefly with more fitness is considered to have higher light intensity).

Further, we also compute the crowding distance metric [25] for each Firefly. Crowding distance d_i of a solution is a measure of search space around solution i that is not occupied by any other solution in the population. In other words, crowding distance is the density estimator of all the solutions in a population. Therefore, we use this metric in-order to maintain diversity in the solutions found during the simulation of NSFA.

In our algorithm of non-dominant sorting multi-objective optimization, we use an external archive with the name *nonDomPopulation* that stores all the non-dominated solution. This is to retain all the non-dominated solution found during the simulation of algorithm. This archive is continuously updated during the simulation and after the completion of simulation, it is returned as final output for the optimization problem. This archive is first initialized in line 7 of the algorithm where all the non-dominated solution from the initial population are identified and stored in the list *nonDomPopu-*

Algorithm 2 Procedure FFAMOVE

1: **Input:** *previousPopulation*: A List with fireflies as its member.
2: **Output:** *newPopulation*: A List with updated position of fireflies
3: n= sizeof(previouspopulation)
4: $F(x_i)$ and $D(x_i)$ denotes Fitness and Crowding distance respectively for a Firefly x_i where $i = (1, 2, 3...n)$
5: **procedure** FFAMOVE
6:     *newPopulation* → *previousPopulation*
7:     **for** $i = 1 : n$ **do**
8:         **for** $j = 1 : n$ **do**
9:             **if** $F(x_i) > F(x_j)$ **then**
10:                 Move Firefly j towards Firefly i by updating it's position in all the dimensions using Equation 2.
11:             **else if** $F(x_i) = F(x_j)$ **then**
12:                 **if** $D(x_i) > D(x_j)$ **then**
13:                     Move Firefly j towards Firefly i by updating it's position in all the dimensions using Equation 2.
14:                 **end if**
15:             **end if**
16:         **end for**
17:     **end for**
18:     **return** *newPopulation*
19: **end procedure**

lation.

In the following, we explain the core working of non-dominant sorting Firefly algorithm. In line 8 of our algorithm 3, list *previousPopulation* is initialized with the contents of a list containing initial population and starts the main part of the simulation in line 9 where Firefly searches the bounded search space and continuously updates the contents of the population. This simulation runs until the stopping criteria is met. In line 10 of our algorithm, *newPopulation* is constructed using the algorithm 2 where the position of each Firefly is changed with an aim to reach a better position. Here, better position means a position close to the true Pareto front. Position of fireflies are changed using the equation 2 in which a less fit Firefly is attracted towards a more fit Firefly. We compare the crowding distance value for each Firefly when fitness of two fireflies are same (Line 10 and 11 of Algorithm 2). We do this because Firefly with higher crowding distance is considered to be in a position of comparatively less denser area. Therefore, when a Firefly moves from a more denser area towards a Firefly in less dense area, probability of finding a more diverse set of solution increases.

After the creation of new population, both the list *newPopulation* and *previousPopulation* are combined into an new population of size *2n* and named *combPopulation* (line 12 of algorithm 3). Then, this new *combPopulation* is searched for any newly found non-dominated solution and if found, is added to the archive *nonDomPopulation* (line 13 of algorithm 3). Although this step of combining population and finding non-dominated solution requires more effort, it allows for the global non-domination check and leads to an elitist strategy. Along with adding the non-dominated solution to *nonDomPopulation*, some solutions are also removed for whom the status of dominance is changed from non-dominant to dominant on addition of new solution. This process of updating *nonDomPopulation* is continued until the stopping criteria for simulation is reached. Finally, the archive *nonDomPopulation* is returned as final solution for the problem representing the Pareto optimal solution (line 16 of algorithm 3).

VI. Experimental Results

In this section we present and discuss the experimental set up that we have organized for the study and discuss the results for accuracy and efficiency of our proposed option pricing model for American options using the multi-objective Firefly algorithm.

A. Experimental Setup

For the current study we have selected 5 equities: Apple (AAPL), Google (GOOG), Goldman Sachs (GS), Amazon (AMZN) and IBM (IBM) and have gathered their data from Bloomberg [26].

We have tested the efficiency of our algorithm by validating our results against the real-world data and compared them with results from binomial lattice model and Monte-Carlo simulation.

CBOE publishes the market's expected volatility on five highly active equities (VXAPL, VXAZN, VXGOG, VXGS, VSIBM). For our experiments, to estimate the volatility, we have used volatility indexes from CBOE for Options on Individual Equities [1]. We did our experiments on 6 months data ranging from January till June 2015. Exhaustively evaluating all the contracts on each date (approx 3500 contract/day)is very difficult. Therefore, in our study, we have selected four particular days for valuation, i.e. four Mondays in each month from January till June 2015 to evaluate our algorithm. Considering only the data for Monday may produce biased results, however in our approach we are trying to capture whatever happened (biased or not) in the market that day. Our results show that we are able to capture real data very accurately, while the results themselves might show market bias.

Further, to show the efficiency of our model on a broader scale of contracts we have selected different possible types of contracts on the basis of *time to expiration* and *moneyness* ($\frac{S}{K}$), which is shown in Table VI-A.The selection of parametric values were done in order to keep track of the market and stocks in the market. That is, by focusing on limited data set, we try to establish the power of proposed algorithm using Firefly. Now that we have shown Firefly can capture real results very accurately, we might be able to included more parametric study in the near future.

Decision on parametric value for the firefly algorithm is very important for it's efficiency in a given application. Therefore,

[1] http://www.cboe.com/micro/equityvix/introduction.aspx

Algorithm 3 Non-Dominant Sorted Firefly Algorithm to evaluate Call Option

1: FA Input parameter: α (Randomness), γ (Attractiveness), β_o (Light Intensity at source), Population Size (n), Number of Iteration (t)
2: Option Input Parameters: Initial Asset Value (S), Expiration Time (T), Strike Price (K), Volatility (σ), Risk-free rate of interest (r)
3: Initialize Upper Bound and Lower bound for each variables representing a Firefly
   a: $S_{min} = S(1-(2*\sigma))$ and $S_{max} = S(1+(2*\sigma))$
   b: $T_{min} = 1$ and $T_{max} = T$
4: Initialize position for all fireflies x_i where $i = (1, 2, ...n)$ and store the population in the list *initPopulation*
   a: Each Firefly x_i is represented by two variables: Stock and Time
   b: S_i is initialised with random real number within range S_{min} and S_{max}
   c: T_i is initialised with random Integer within a range T_{min} and T_{max}
5: For each Firefly x_i evaluate their objective function values
   a: Calculate Pay-off for each Firefly x_i using equation $max(S_i - K, 0)$, where S_i represents asset value of each Firefly x_i
   b: For each Firefly x_i Call procedure PROBCAL such that $Probabililty(x_i)$= PROBCAL(S,S_i,T_i,σ,r)
6: Sort each Firefly according to their Non Domination level and evaluate their Fitness and Crowding Density
   a: Fitness is calculated as $F(x_i) = n - NonDomLevel(x_i)$
   b: Crowding Distance is calculated by calling procedure $CrowdingSort(x_i)$
7: Identify the non-dominated solution from population and store them in the list *nonDomPopulation*
8: Assign list *previousPopulation* $\rightarrow$ *initPopulation*
9: **while** $t < MaxGeneration$ **do**
10: $newPopulation \rightarrow FFAMOVE(previousPopulation)$
11: Evaluate Fitness and Crowding Distance of *newPopulation*
12: Combine *previousPopulation* and *newPopulation* into a list *combPopulation* of size 2n
13: Identify the non-dominated set of solution from the list *combPopulation* and update the list *nonDomPopulation*. Also remove any solution from the list if it gets dominated by adding new solutions.
14: Assign *previousPopulation* $\rightarrow$ *newPopulation*
15: **end while**
16: Return the final set of *nonDomPopulation* as a set of Pareto optimal solution for Option Pricing Problem

after trying different possible combinations of parameters, final values used in our experimentation is presented in Table VI-A.

After creating the experimental setup, we did different types of experiments to validate our model and algorithm for American option pricing. We did the evaluation of experiments in two parts: (i) we showed the efficiency of our algorithm to find the complete Pareto front for the American option prices and (ii) we did the analysis of the solution from our algorithm to evaluate the correctness of our model.

TABLE I
VARIATION OF CONTRACTS BASED ON TIME TO EXPIRATION AND MONEYNESS

Expiration Period	**Moneyness S/K**
1 Week	0.95,0.98,1,1.02,1.05
1 Month	0.95,0.98,1,1.02,1.05
3 Months	0.90,0.95,1,1.05,1.1
6 Months	0.90,0.95,1,1.05,1.1
1 Year	0.90,0.95,1,1.05,1.1

TABLE II
PARAMETER SETTING FOR FIREFLY ALGORITHM

Parameter	**Value**
Population Size	100
Number of Iteration	250
Randomization parameter (α)	0.2
Light absorption coefficient (γ)	1.0
Attractiveness value (β_o)	1.0

B. Pareto Front for Option Value

First, we discuss the Pareto front obtained from the implementation of our model. We have performed non-dominant sorting firefly algorithm on over 720 contracts with different combinations of "maturity" and "moneyness" on different possible dates and with different underlying assets. In all the experiments performed, our algorithm was successful in finding the complete set of Pareto optimal solutions for American style of option where trade-off solutions have to be searched for the objectives: pay-off and probability. Some of the Pareto fronts computed are shown in Figures 2 to 4 where, it is clearly observable that non-dominant sorting firefly algorithm was successful in finding the Pareto-optimal solution for multi-objective option pricing problem. For example, in Figure 2, which is the representation of the solutions found for 6 months at-the-money contract on January 5, 2015 for Apple stock, we can see that the our algorithm (NSFA) successfully found the solution for maximum and minimum of both the objectives (represented by maximum payoff and minimum payoff arrow in figure) and also successfully found the set of all other non-dominant solutions with respect to both objectives (represented by all points between the maximum payoff and minimum pay off arrow). Similar behavior is also observed with other contracts shown in Figures 3 and 4, which validates the ability of NSFA to find solution for multi-objective optimization problem.

Further, in order to analyse the quality (accuracy and efficiency) of the Pareto front found using our algorithm, we plotted the pay-off (y-axis), probability (x-axis) and error (shown using colours: Red for negative error and green for positive error, brightness resembles the magnitude of error) for all the solutions on a scatter plot using data visualization tool, called Tableau [2]. Some of the visualization are shown in Figures 2 to 4. Error is calculated using the actual worth of the contracts. As we were experimenting on historical data, it was possible to get actual worth of all the contracts which was further used to determine the quality of our Pareto front. One important insight we observed on analyzing all the 600 computed Pareto fronts is that in 98% of the experiments our algorithm was successful in capturing the actual worth of the option contract with less than 1% error. For example, let us consider Figure 4 in which solutions found using NSFA are shown (at-the-money 6 months contract for Goldman Sachs). Here, S_T=218.39, S_o=193.06 and K=193. The real option value of the contract from historical data is noted as 25.4. In Figure 4 one of the solution represents pay-off as 25.84 which is just 0.44 more than the actual value. This solution point is identified with an arrow in the Figure 4. Similarly in other contracts also solutions found using NSFA algorithm captured the solution with almost 0-1 % error. This observation shows us the ability that our model of option pricing as multi-objective optimization is capable of capturing the accurate worth of the contract in real market conditions.

Finally, we conclude using the observation from this section that 1) Non-dominant sorting Firefly algorithm efficiently finds the true Pareto-optimal solution for multi-objective option pricing problem and can be further extended to optimize any other multi-objective optimization problem (2) Our model of mapping probability and pay-off as a multi-objective optimization problem is accurate and is successful in capturing the true behavior of most of the American style of option contracts.

C. Risk-Aware Application of the Model

After validating our algorithm and our model, we analyze how investor will deploy this study to find the accurate worth of a contract. In our model each solution on the Pareto front, represents two variables: the *pay-off* and *probability of getting that pay-off* and an investor is always expected to select a solution from the Pareto front which gives maximum values for both these objectives. However, it is difficult to find such a solution as both the objective has tendency of behaving in opposite manner. Therefore, the investor has to select a point, which finds a trades-off between both these objectives. Our model and algorithm mainly presents a Pareto front of solutions to the investor and advise investor to select one particular solution from the Pareto front on the basis of their *decision variable* that estimates the accurate worth of an option with minimum error.

Financial markets are very volatile due to which it is not possible to have a single value for any option contract. Therefore, we believe that every option contract has its own value depending on the level of risk an investor may take. In

[2]http://get.tableau.com/trial/tableau-software.html

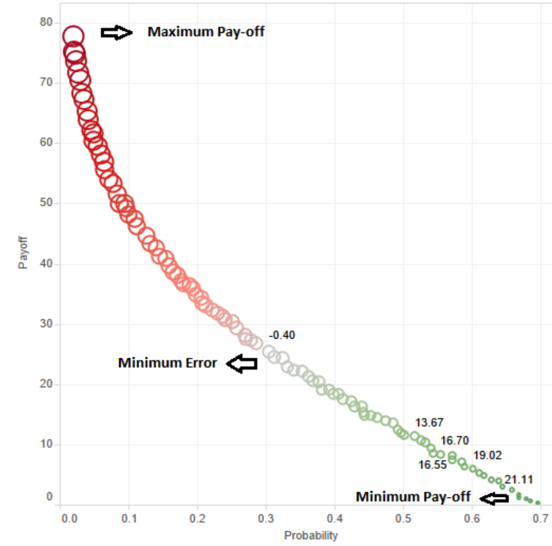

Fig. 2. Pareto Front for at-the-money 6 months contract on January 5, 2015 for Apple Stock

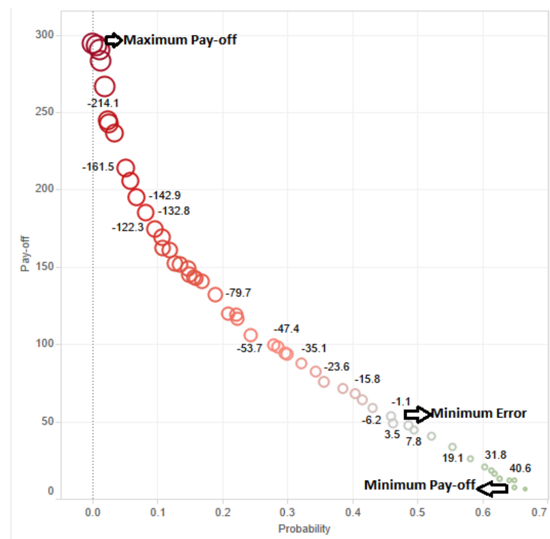

Fig. 3. Pareto Front for at-the-money 6 months contract on January 5, 2015 for Google Stock

other words, value of an option contract varies with the risk taking capacity of an investor. Therefore, in our algorithm, we may assume *investment risk* as the decision variable to compute the worth of an option. We suggest mapping the risk taking capacity of an investor with the probability of attaining a pay-off. On the basis of risk capacity of an investor as a decision variable, the investor can select a point on the Pareto front to get an option value for the contract. For example if an investor is not willing to take any risk, then he/she can select a solution with high *probability* (referred as "Minimum Payoff" in Figures 2 to 4) that may give less profit to the investor. Similarly, if an investor is willing to take high risk then he/she may select a solution with low probability and high payoff (referred to as high payoff in Figure 2 to 4) that

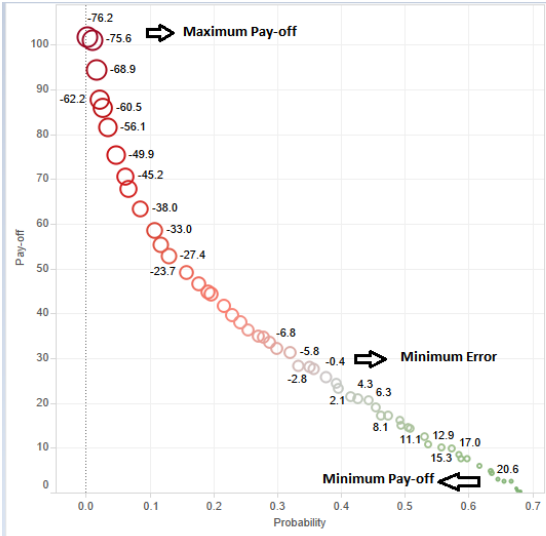

Fig. 4. Pareto Front for at-the-money 6 months contract on January 5, 2015 for Goldman Sachs Stock

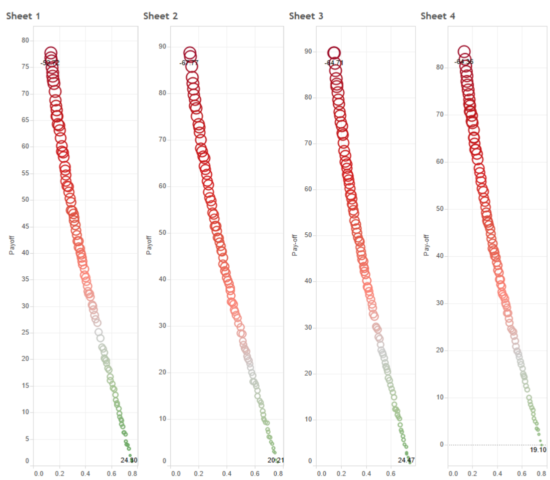

Fig. 5. Pareto Front for at-the-money 1 Year contract on 4 Mondays in January, 2015 for Apple Stock

may give high profit to the investor but chances of getting that profit is very less. Investor usually select the risk factor on the basis of volatility and current market condition. In the next section we have introduced a strategy for investor to select risk variable on basis of historical data that may help investors to approximate the accurate worth of the contract.

D. Strategy to evaluate current risk level

We have analyzed all our experiments done on the historical data, in order to identify the relation between the risk level and worth of an option contract. In Figures 5 - 7, the degree of similarity in Pareto fronts for similar type of contracts is clearly visible. Therefore, in this section we proposed a strategy of using the historical data and analyze Pareto front for similar type of contracts from previous dates in order to predict the current level of risk.

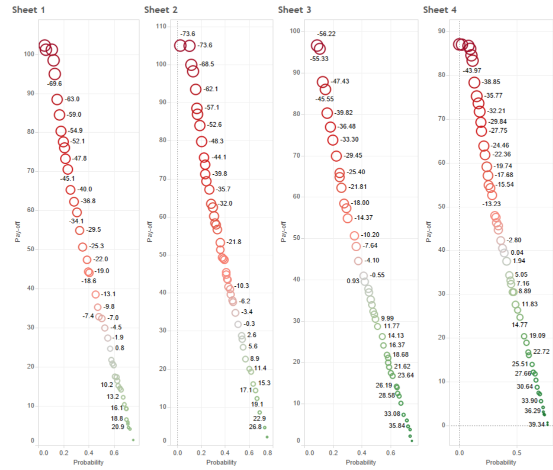

Fig. 6. Pareto Front for at-the-money 1 Year contract on 4 Mondays in January, 2015 for Goldman Sachs Stock

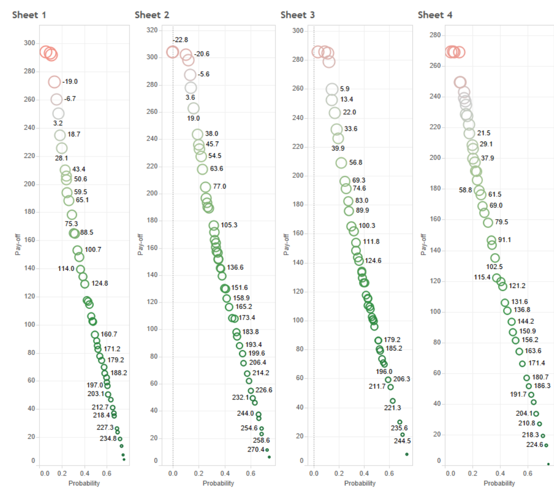

Fig. 7. Pareto Front for at-the-money 1 Year contract on 4 Mondays in January, 2015 for Google Stock

In order the analyze the accuracy of this strategy, we have done back-testing using 6 months historical data from January 2015 to June 2015. We have considered data of 4 Mondays of each month. In our strategy we analyze each and every historical contract for the considered period and note that value of probability or risk for each contract where, the error is found to be minimum. Then we compute the average value of probability or risk for each type of contract where it has maximum chances of getting minimum error. Then we use that average computed value as the average risk level for that particular type of contract and use it to compute the option price.

For example, If we want to evaluate a contract on 6th July, 2015 (Monday), we considered the probability of Mondays in previous 6 months. We observe the Pareto front of all contracts on all these 24 days and noted those solutions on their front, which gives the minimum error. Further, we compute the average probability for noted solution and consider it as the current risk level. Further, using the computed risk level we select a solution on Pareto front for the current contract and use its pay-off value as the solution.

Also, we did some experiments to evaluate option price for American style of contracts for 5 stocks and have noted the error between real worth and the worth predicted using our strategy. We also compared the results with other two popular American option pricing techniques: Monte-Carlo simulation [27] and Binomial Lattice model [2]. Results are shown in Tables III-V . It is clearly observable that result using this strategy are much better than the other conventional techniques. Similarly, we can explore different risk selecting strategies to predict the accurate worth of a contract.

From our experiments we made one more crucial observation that for similar type of contracts on different dates, option prices listed on the exchange have same probability of getting that option price. For example, for at-the-money 1 year contract on Apple on 4 Mondays in January 2015, option price asked on the exchange was 4.08, 4.9, 4.64, and 4.52 respectively. Option parameter on all four dates is different, that is, S_0, σ, r are different on all the dates. However, probability of attaining respective pay-off listed on exchange, for all different dates, turns out to be same i.e. 0.718. This type of behavior is also seen in all other contracts. This insight from the analysis depicts that the market tries to price the contract using same order of risk, which tends to be small. That is, they try to give the minimum price of contract and our model is able to capture that accurately, which further proves the efficiency of our model.

TABLE III
COMPARISON OF ERROR BETWEEN NSFA, MONTE-CARLO SIMULATION AND BINOMIAL LATTICE FOR AT-THE-MONEY OPTION PRICING

Maturity	**Error using FA algorithm**	**Error using Monte-Carlo Simulation**	**Error using Binomial Lattice**
1Week	14	50	46
1Month	35	51	51
3Month	14.3	56	54.5
6Month	29.1	37.2	37.8
1Year	12.4	31.9	30.8

TABLE IV
COMPARISON OF ERROR BETWEEN NSFA, MONTE-CARLO SIMULATION AND BINOMIAL LATTICE FOR OUT-OF-MONEY OPTION PRICING

Maturity	**Error using FA algorithm**	**Error using Monte-Carlo Simulation**	**Error using Binomial Lattice**
1Week	20	55	48
1Month	28	55	52
3Month	20.7	55.9	53.4
6Month	25.2	35.8	37.2
1Year	14.8	33.1	34.6

VII. CONCLUSIONS

The main goal of this study is to solve option pricing problem using the nature-inspired optimization techniques. Two major contributions from our study are: (i) proposal of a novel model where we mapped option pricing problem as a multi-objective optimization problem, (ii) design and development of multi-objective Firefly algorithm that was used to compute the solution for the option pricing problem.

TABLE V
COMPARISON OF ERROR BETWEEN NSFA, MONTE-CARLO SIMULATION AND BINOMIAL LATTICE FOR IN-THE-MONEY OPTION PRICING

Maturity	**Error using FA algorithm**	**Error using Monte-Carlo Simulation**	**Error using Binomial Lattice**
1Week	12	52	45
1Month	24.5	45.8	47.1
3Month	10.6	47.4	44.2
6Month	22.1	34.3	35.6
1Year	10.7	30.4	30.9

We have proposed the pay-off from the option contract and the probability of attaining that pay-off as the two major optimization objectives and used them with new firefly algorithms to find Pareto optimal solution(s). Since option pricing is modelled as a multi-objective optimization problem, we present not one but a set of solutions to the investor for the option pricing problem. To the best of our knowledge this is the first attempt of studying financial option pricing problem as a multi-objective optimization using the firefly algorithm.

With our experiments, we were able to show the competency of our model and accuracy and efficiency of our algorithm to price American style of options. In this study, we designed experiments in order to cover a spectrum of option contracts with multiple strike prices, expiration dates and stocks. In all these experiments, our algorithm was successful in finding the Pareto front for option prices and that the solution computed from all experiments captured the real market values very accurately. Analyzing all the Pareto fronts, it was observed that in 99% of the experiments, our algorithm was successful in capturing the true option price as a solution on the Pareto front with less than 2% error. That is, in almost all the experiments, our algorithm was able to find a solution that is equal to option price of a contract as available from real (historical) market data. Therefore, we can conclude that using our model and algorithm, an investor can accurately evaluate the option contract for a given level of risk to enable him/her to decide before entering the contract.

Moreover, we proposed a strategy using the historical data to help investors to approximate decision variable (risk level) in order to evaluate the accurate worth of a contract. This strategy worked best with promising results for American options. Our experiments showed significant results for this strategy. Therefore, we conclude that the strategy of using historical data to find the risk level is very efficient in terms of accuracy and can be used by investors in real financial market. This study can be extended and larger set of data (2-5 years) for each stock can be used to get a more refined historical risk level for any particular type of contract. Also, a tool can be developed, which keeps refining the risk level value continuously for multiple type of stocks on different indices. However, for this tool, data becomes too big if we consider all the listed stocks or contracts on the wall street and therefore, various analytics from the field of big data may be needed to tackle such an issues. We leave this as a important and immediate future work.

Further, various other risk selection strategies can be explored and their effects can be studied for different trading strategies. Our technique presents a set of solutions to the user and lets user to choose any one solution but the basis on which user selects the solution is something that needs to be studied further. Basically how to use the Pareto front in different ways and determining the decision variable (risk level) can be seen as a new future study. Also, Pareto front and its relation with different type of option trading strategies can be analyzed in future. Our methodology can be extended to price exotic options also such as Asian or Russian option and its performance or results can be analyzed.

Acknowledgements

The second and third authors acknowledge Natural Sciences and Engineering Research Council (NSERC) Canada for partial financial support for this research through Discovery Grants. The first author acknowledges the support by Faculty of Graduate studies, University of Manitoba for their financial support through Graduate Enhancement of Tri-Council Stipends (GETS).

References

[1] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *The journal of political economy*, pages 637–654, 1973.

[2] Robert C Merton. Theory of rational option pricing. *Bell journal of economics and Management Science*, pages 141–183, 1973.

[3] John C Cox, Stephen A Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979.

[4] J C Hull. *Options, Futures and Other Derivatives, 9th edition*. Prentice Hall, 2014.

[5] Lishang Jiang and Canguo Li. *Mathematical modeling and methods of option pricing*. World Scientific, 2005.

[6] Michael J Brennan and Eduardo S Schwartz. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and Quantitative Analysis*, 13(03):461–474, 1978.

[7] Ales Cerny. *Mathematical Techniques in Finance - Tools for Incomplete Markets*. Princeton Univ. Pres,, NJ, USA, 2004.

[8] Xin-She Yang. *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[9] Sameer Kumar, Gitika Chadha, Ruppa K Thulasiram, and Parimala Thulasiraman. Ant colony optimization to price exotic options. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation Conference*, pages 2366–2373. IEEE, 2009.

[10] Ruppa K Thulasiram, Parimala Thulasiraman, Hari Prasain, and Girish Jha. Nature- inspired soft computing for financial option pricing using high performance analytics. *Concurrency and Computation: Practice and Experience - appeared on-line First*, 2014.

[11] N K Chidambaran, Chi-Wen Jevons Lee, and Joaguin R Trigueros. An adaptive evolutionary approach to option pricing via genetic programming. 1998.

[12] Guang He, Nanjing Huang, Huiqiang Ma, Jue Lu, and Meng Wu. An improved particle swarm optimization algorithm for option pricing. In *Proceedings of the Eighth International Conference on Management Science and Engineering Management*, pages 861–869. Springer, 2014.

[13] Paul Wilmott. *Paul Wilmott introduces quantitative finance*. John Wiley & Sons, 2007.

[14] Christian Keber and Matthias G Schuster. Generalized ant programming in option pricing: Determining implied volatilities based on american put options. In *Computational Intelligence for Financial Engineering, 2003. Proceedings. 2003 IEEE International Conference on*, pages 123–130. IEEE, 2003.

[15] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[16] Karl Doerner, Walter J Gutjahr, Richard F Hartl, Christine Strauss, and Christian Stummer. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of operations research*, 131(1-4):79–99, 2004.

[17] James M Hutchinson, Andrew W Lo, and Tomaso Poggio. A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, 49(3):851–889, 1994.

[18] Zheng Yin, Anthony Brabazon, and Conall O'Sullivan. Adaptive genetic programming for option pricing. In *Proceedings of the 9th annual conference companion on Genetic and evolutionary computation*, pages 2588–2594. ACM, 2007.

[19] Christian Keber and Matthias G Schuster. Option valuation with generalized ant programming. In *GECCO*, pages 74–81, 2002.

[20] Marco Dorigo and Mauro Birattari. Ant colony optimization. In *Encyclopedia of machine learning*, pages 36–39. Springer, 2010.

[21] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

[22] Gobind Preet Singh, Ruppa K Thulasiram, and Parimala Thulasiraman. Muti-objective firefly algorithm for pricing european option. *IEEE SAI Intelligent Systems Conference, London*, September, 2016.

[23] Eugene F Fama. The behavior of stock-market prices. *Journal of business*, pages 34–105, 1965.

[24] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[25] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

[26] L. P. Bloomberg. Option data for s&p 500 index, 2015.

[27] Francis A Longstaff and Eduardo S Schwartz. Valuing american options by simulation: a simple least-squares approach. *Review of Financial studies*, 14(1):113–147, 2001.