



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Bio-inspired Metaheuristics for Bin Packing Problems

R. Yesodha¹

M.Phil Research Scholar¹

Department of Computer Applications

Bharathiar University, Coimbatore, INDIA

Email: yesodhamca@gmail.com

T. Amudha²

Assistant Professor²

Department of Computer Applications

Bharathiar University, Coimbatore, INDIA

Email: amudhaswamynathan@buc.edu.in

Abstract: Bio-inspired computing is a subset of nature-inspired methods that focus on social behavior and emergence of biological species. This field is much closely related to the domain of Artificial Intelligence. Biologically-inspired methods are becoming increasingly important in face of the complexity of today's demanding applications. In this paper, One-Dimensional Bin Packing Problem (1-BPP) with fixed sized bins was analyzed. 1-BPP is to pack a set of items with different weights into a minimum number of bins, minimizing bin space wastage and execution time. A Bio-inspired algorithm called FireFly Algorithm (FFA) and a traditional algorithm called First-Fit were implemented to solve benchmark instances of Bin Packing problem. The performance of these algorithms was evaluated by testing on two classes of benchmark Data set 1 and Data set 2 for BPP-1 and the results have shown the competence of Bio-inspired metaheuristics in problem solving.

Keywords: Combinatorial Optimization, Bin Packing Problem, FireFly Algorithm, First-Fit algorithm.

I. Introduction

In the last two decades, the computational researchers are increasingly interested in natural sciences, especially biology as a source of modeling paradigms. Bio-inspired usage has been very promising in various fields ranging from computer science, electronics and mechanical engineering to chemical engineering and molecular biology [13]. The Bin Packing problem (BPP) was introduced by Garey & Johnson in 1979. It is a combinatorial NP-hard problem. Bin packing is a basic term that takes place in factory outlets and various places throughout the world. Most commonly it involves packing smaller items into defined spaces. In the real world, the critical issue is to make efficient use of time and space. Some areas where the BPP can occur are in Container Loading, Cutting Stock, Packaging Design, Resource Allocation, Scheduling, Multiple Knapsack Problem, Cargo Airplanes and Ships. The list is endless and extremely diverse [12]. Bin packing problem has wide opportunity to solve with the new era of Bio-inspired computing. Very few Bio-inspired metaheuristics were applied in recent years such as Ant Colony Optimization algorithm (ACO), Cuckoo search algorithm (CS) and Genetic algorithm (GA) have been successfully applied to Bin Packing problem and the results were proven to be highly efficient. In this research work BPP was solved by using one traditional heuristic and one metaheuristic which was inspired by biology. There are three main variants of Bin Packing problems: one dimensional, two dimensional and three dimensional Bin Packing problems and other variants are High Multiplicity Bin Packing Problems, linear packing, packing by weight, packing by cost etc.,

II. Related Work

Abdesslem Layeb and Seriel Rayene Boussalia (2012) have proposed a new stochastic method based on the Quantum Inspired Cuckoo Search algorithm (QICSA) to deal with the 1-BPP problem. A new hybrid measure operation integrates the FF heuristic in the core of the standard measure operation. The great feature of this hybrid measure is its great ability to generate different good random solutions. The proposed algorithm reduces the population size and the number of iterations to have the optimal solution. The obtained results are very encouraging and show the feasibility and effectiveness of the proposed approach [1].

Aphirak Khadwilard., et al., (2011) have presented the successful application of a recent developed metaheuristic called FireFly Algorithm for solving JSSP, explore the parameters of the proposed algorithm and investigated the performance of the FireFly Algorithm with different parameter setting and compare with best known solution. The computational experiment was designed and conducted using five benchmarking JSSP data

sets from a classical OR-Library. The analysis was aimed to study the effect of the FireFly parameter setting on its performance before comparing the FireFly Algorithm results between using and not using optimized parameter settings. The FireFly Algorithm performance with optimized parameter setting could produce the operation schedule better than the other parameter setting [2].

Sh. M. Farahani, et al., (2011) have stabilized the firefly's movement and proposed a new behavior to direct firefly's movement to a global best. In their proposed algorithm, if a firefly can't find any better firefly that is brighter in its neighborhood, it will move towards global best in that iteration, so firefly's movement will direct to better solution and algorithm can guide them to better state, so they can get near to optimum solution at the end of iteration. Moving fireflies by a Gaussian distribution as a social behavior causes a better position for each of them for next iteration and fireflies with worse cost have more chance to move to global best with a longer step length. Simulation results have shown a better performance than standard Firefly algorithm. Proposed algorithm was tested on five standard functions that were commonly used for testing the static optimization algorithms [6].

Mohamed Maiza and Mohammed Said Radjef (2011) have aimed to solve one dimensional Bin Packing problem with conflicts. The conflicts are represented by a graph whose nodes are the items, and adjacent items cannot be packed into the same bin. They have proposed an adaptation of Minimum Bin Slack heuristic with a combination of heuristics based on the uses of the classical bin-packing methods to packing items of maximal-stable-subsets (MSS). Computational results based on a benchmark test showed good performance of proposed heuristics both on quality of solution and execution time [8].

Ruslan Sadykov and Francois Vanderbeck (2012) presented a branch-and-price algorithm for the bin packing problem. Implementation was made by using the software platform BaPCod. They developed a dynamic programming algorithm for pricing when the conflict graph is an interval graph and a depth-first-search and branch-and-bound approach for pricing when the conflict graph has no special structure. Their computational experiment report sets new benchmark results for this problem. The algorithm was tested on instances from the literature and newly generated ones. Branch-and-price algorithm gave comparatively better results [9].

III. Bin Packing Problem Solutions

This paper aimed to solve the NP-Hard Bin Packing Problem by using FireFly Algorithm and First-Fit algorithm and to examine the efficiency of both the algorithms in solving benchmark instances of BPP. The Bin Packing problem can be formulated as follows:

$$\begin{aligned} \text{Minimize } \sum_{j=1}^n y_j & \longrightarrow (1) \\ \text{subject to constraints} & \\ \sum_{i=1}^m w_i x_{ij} \leq cy_j \quad \forall j \in \{1, \dots, n\} & \longrightarrow (2) \\ \sum_{j=1}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} & \longrightarrow (3) \\ x_{ij} \in \{0, 1\} \quad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, n\} & \longrightarrow (4) \\ y_j \in \{0, 1\} \quad \forall j \in \{1, \dots, n\} & \longrightarrow (5) \end{aligned}$$

With, $y_i = 1$ if the bin i is used; else 0
 $x_{ij} = 1$ if the item j is stocked in bin i .

Notations

c : Bin capacity,
 i, j : Index of items and bins respectively
 w_i : Weight or size of item i
 m : minimal number of bins

Equation (1) states the objective function which is to minimize the number of bins assigned. Equation (2) ensures the constraint that the bin capacity is not exceeded by the sum of item sizes assigned to this bin. This is often called the capacity constraint. Equation (3) ensures that every item is packed to exactly one bin. Equation (4) and (5) determine that all variables are binary [1, 4].

A. FireFly Algorithm

The FireFly Algorithm (FFA) is a metaheuristic algorithm, inspired by the flashing behavior of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. The flashing characteristics of fireflies consequently develop FireFly-inspired algorithms. There are three idealized rules. On the first rule, each firefly attracts all the other fireflies with weaker flashes. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex. Secondly, attractiveness is proportional to their brightness which is reverse proportional to their distances. For any two flashing fireflies, the less bright one will move towards the brighter one [3]. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly. Finally, no firefly can attract the brightest firefly and it moves randomly. Based on these three rules, the basic steps of the FireFly Algorithm (FFA) can be summarized as the pseudo code shown below. In the FireFly algorithm there

are two important issues of the variation of light intensity and the formulation of the attractiveness. The light intensity $I(r)$ varies with distance r monotonically and exponentially

$$I = I_0 e^{-\gamma r} \longrightarrow (6)$$

where I_0 is the original light intensity and γ is the light absorption coefficient. For simplicity, it is assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function of the optimization problems. On the attractiveness of the FFA the main form of attractiveness function or $\beta(r)$ can be any monotonically decreasing functions such as the following generalized form of

$$\beta(r) = \beta_0 e^{-\gamma r^2} \longrightarrow (7)$$

where r or r_{ij} is the distance between the i^{th} and j^{th} of two fireflies. β_0 is the attractiveness at $r = 0$ and γ is a fixed light absorption coefficient. The distance between any two fireflies i and j at x_i and x_j is the Cartesian distance as follows:

$$r_{ij} = \|x_i - x_j\|_2 = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \longrightarrow (8)$$

where $x_{i,k}$ is the k -th component of the i -th firefly (x_i). The movement of a firefly, i is attracted to another more attractive (brighter) firefly j , is determined by

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha * (\text{rand} - 0.5) \longrightarrow (9)$$

where the second term is due to the attraction while the third term is the randomization with α being the

PSEUDO CODE OF FIREFLY ALGORITHM

```

Procedure FFA Meta-heuristic()
Begin
  Initialise algorithm parameters
  MaxGen: the maximal number of generations
   $\gamma$ : the light absorption coefficient
   $r$ : the particular distance from the light source
   $d$ : the domain space
  Define the objective function of  $f(x)$ , where  $x = (x_1, \dots, x_d)^T$ 
  Generate the initial population of fireflies or  $x_i$  ( $i=1, 2, \dots, n$ )
  Determine the light intensity of  $I_i$  at  $x_i$  via  $f(x_i)$ 
  While ( $t < \text{MaxGen}$ )
    For  $i = 1$  to  $n$  (all  $n$  fireflies)
      For  $j = 1$  to  $n$  (all  $n$  fireflies)
        if ( $I_j > I_i$ ), move firefly  $i$  towards  $j$ 
      end if
      Attractiveness varies with distance  $r$  via  $\text{Exp}[-\gamma r^2]$ 
      Evaluate new solutions and update light intensity;
    End for j
  End for i
  Rank the fireflies and find the current best
  End while
  Postprocess results and visualization
End procedure

```

randomization parameter. Rand is a random number generator uniformly distributed in the range of $[0, 1]$. In this paper, the initial population of firefly is 5 and the iterations are 20. In the each iteration 5 different fireflies pack the number of items with different weights and its corresponding capacity from different positions and produce different solutions. Each firefly calculates the light intensity based on the objective function of the BPP and also based on the distance. The attractiveness of a firefly is determined by its brightness and varies according to the distance. The distance of each firefly is calculated based on the starting position which it starts packing the items. In this research work the distance between fireflies is calculated by using Cartesian distance. The movement of one firefly is attracted to another more attractive (brighter) firefly [2, 6].

Table 1: PARAMETER SETTINGS FOR FIREFLY ALGORITHM

Parameter	Values	Description
α	0.2	Alpha
β_0	0.3	Beta ₀
γ	0.2	Gamma
Firefly	5	Population of Fireflies
Iterations	20	Generations

B. First-Fit Algorithm

First-Fit (FF) keeps all empty bins open. It places the next item in the lowest numbered bin in which the item fits. If it does not fit in any bin, a new bin is opened. First-Fit could not achieve better execution time as it keeps all non-empty bins active and tries to pack every item in these bins before opening a new one. In this algorithm the rule followed is: First place an item in the first bin, called lowest indexed bin into which it will fit, i.e., if there is any partially filled bin then place the item in the lowest indexed bin otherwise, start a new bin. First-Fit is an $O(n \log n)$ -time algorithm [11].

IV. Implementation Results and Discussion

The results obtained from FireFly Algorithm were compared with First-Fit algorithm. Experimental results and comparative study of the algorithms are discussed in this section.

A. Bin Packing Test Problems

In this research work the instance are classified into two classes namely Easy and Medium instances. The benchmark in the easy class instance, n (number of items) ranges from 50 – 500 and c (bin capacity) ranges from 100-150. In the medium class instance n ranges from 50 – 500 and c equals 1000 [13].

B. Implementation Results of Easy Class Instances

In this research work, 15 instances of easy class were solved by using First-Fit and FireFly Algorithm. The results of two algorithms were compared and analyzed. FireFly Algorithm gave optimal solutions for 6 instances out of 15 with minimum bin space wastage. Whereas, out of 15 instances First-Fit algorithm was found to be capable of producing 8 near optimal solutions and incase of 7 instances, it could not achieve optimal solutions. It was found from the results that FireFly Algorithm are efficient in terms of execution time when compare to First-Fit algorithm. The computational results are given in Table 2 and the Figure 1 and Figure 2 shows the graphical representation of Table 2.

TABLE 2: COMPARATIVE RESULTS OF EASY CLASS

Ins Name	N	C	Opt	FIRST- FIT TECHNIQUE			FIREFLY TECHNIQUE		
				No. of bins used	Free Space	Exe Time (msec)	No. of bins used	Free Space	Exe Time (msec)
N1C1W1_B	50	100	31	35	718	46	31	318	12
N1C1W1_C	50	100	20	26	616	31	21	116	10
N1C1W1_N	50	100	25	30	579	31	28	269	13
N1C2W1_H	50	120	23	30	977	31	23	137	10
N1C2W1_I	50	120	27	35	1280	31	27	760	13
N1C2W1_K	50	120	24	32	1144	31	24	184	10
N1C2W1_L	50	120	25	32	1093	32	25	253	10
N1C3W1_F	50	150	20	29	1435	32	20	85	10
N2C1W1_A	100	100	48	61	1428	32	49	228	15
N2C2W1_J	100	120	42	58	6960	31	51	538	16
N2C3W1_A	100	150	35	50	3862	31	36	174	15
N3C1W4_S	200	100	145	169	4025	47	146	3750	20
N3C1W1_A	200	100	105	131	3012	31	106	512	16
N4C1W1_L	500	100	258	329	7557	78	262	6685	29
N4C1W1_M	500	100	246	310	6578	78	249	478	25

C. Implementation Results of medium Class Instances

In medium class, 20 instances were solved. FireFly Algorithm gave optimal solutions for 10 instances with minimum bin space wastage and for remaining 10 instances it obtained near optimal. Whereas, out of 20 instances First-Fit algorithm was found to be capable of producing 3 optimal solutions and for remaining 17 instances, it found near optimal solutions. It was found from the results that FireFly Algorithm are efficient in terms of execution time when compare to First-Fit algorithm. The computational results are given in Table 3 and the Figure 3 and Figure 4 shows the graphical representation of Table 3.

TABLE 3: COMPARATIVE RESULTS OF MEDIUM CLASS

Ins Name	N	C	Opt	FIRST- FIT TECHNIQUE			FIREFLY TECHNIQUE		
				No. of bins used	Free Space	Exe Time (msec)	No. of bins used	Free Space	Exe Time (msec)
N1W1B1R0	50	1000	18	21	4066	31	19	2066	13
N1W1B1R1	50	1000	18	22	5054	31	19	2054	13
N1W1B1R3	50	1000	18	21	4155	31	19	2155	13
N1W1B1R4	50	1000	17	21	4590	31	18	1590	13
N1W1B1R5	50	1000	17	21	4534	31	17	534	13
N1W1B1R6	50	1000	17	21	4239	31	18	1239	13
N1W1B1R7	50	1000	17	21	4411	31	18	1411	13
N1W1B1R8	50	1000	17	21	4184	31	18	1590	13
N1W4B1R8	50	1000	6	6	624	31	6	624	10
N1W1B3R0	50	1000	17	26	1578	31	17	866	15
N1W1B3R7	50	1000	18	26	8069	31	19	1069	15
N1W3B2R0	50	1000	8	8	842	16	8	842	10
N1W3B3R4	50	1000	8	8	715	16	8	715	10
N2W1B1R0	100	1000	34	41	8106	32	34	4212	15
N2W1B1R6	100	1000	34	43	9511	32	36	2511	15
N2W4B2R1	100	1000	11	12	1331	16	11	331	10
N3W4B3R0	200	1000	24	26	2068	31	25	1068	18
N4W4B3R4	500	1000	55	59	4543	47	55	543	21
N4W4B3R6	500	1000	58	62	4595	47	58	595	21
N4W4B3R9	500	1000	56	60	4066	42	56	696	21

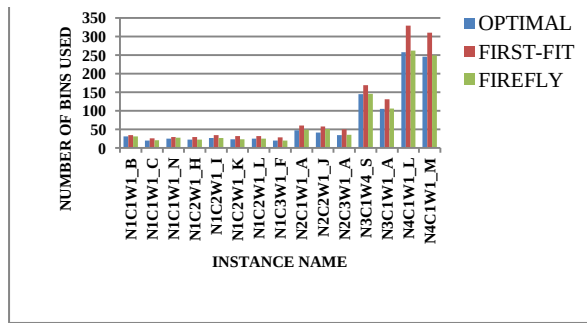


Figure 1: Number of bins used for Easy class

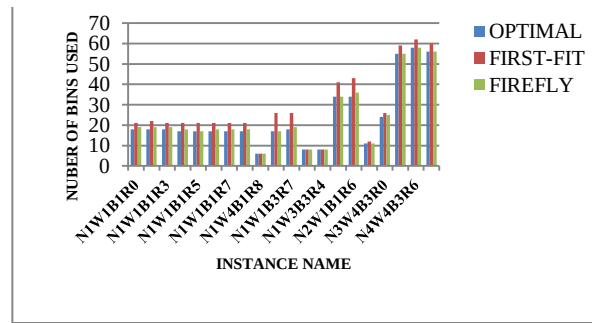


Figure 3: Number of bins used for Medium class

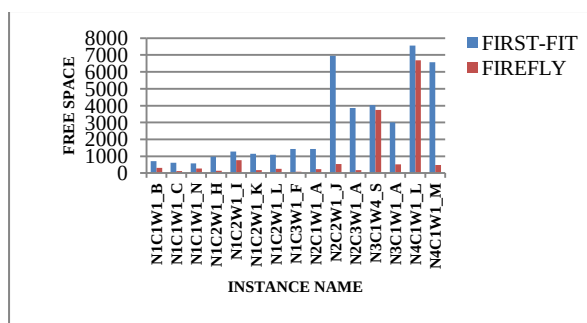


Figure 2: Bin space wastage for Easy class

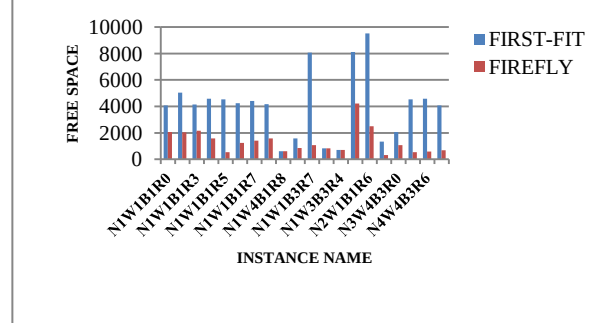


Figure 4: Bin space wastage for Medium class

V. Conclusion and Future work

In this research work, a Bio-inspired algorithm called FireFly Algorithm (FFA) and a traditional algorithm called First-Fit algorithm were implemented for One Dimensional BPP with fixed sized bins. These algorithms were tested on standard benchmark problem instances. The results obtained by the FireFly Algorithm shows good performance for most of the problem cases when compared to First-Fit algorithm. This work can be extended by modifying the parameter settings for FireFly Algorithm to improve the results. Currently, our research work is focused towards applying FFA for other benchmarks available. FireFly Algorithm can be effectively modified and be made suitable for other types of combinatorial optimization problems. FireFly Algorithm can also be applied to solve various other types of dimensions of the Bin Packing problem such as two dimensional, three dimensional etc.

VI. References

- [1] Abdesslem Layeb and Seriel Rayene Boussalia, "A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem", International Journal of Information Technology and Computer Science, 5, 58-67, 2012.
- [2] Aphirak Khadwilard, Sirikarn Chansombat, Thatchai Thepphakorn, Peeraya Thapatsuan, Warattapop Chainate and Pupong Pongcharoen, "Investigation of Firefly Algorithm Parameter Setting for Solving Job Shop Scheduling Problems", 2011.
- [3] Chai-ead N, P. Aungkulanon and P. Luangpaiboon, "Bees and Firefly Algorithms for Noisy Non-Linear Optimisation Problems", Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol II, 2011.
- [4] Doina Bein, Wolfgang Bein and Swathi Venigella, "Cloud Storage and Online Bin Packing", accepted at Intelligent Distributed Computing Conference (IDC 2011), The Delft, Netherlands, October 5-7, 2011.
- [5] Falkenauer. E and A.Delchambre, "A Genetic Algorithm for Bin Packing and Line Balancing", Proceedings of the IEEE International Conference on Robotics and Automation, 1992.
- [6] Farahani Sh. M., A. A. Abshouri, B. Nasiri, and M. R. Meybodi, "A Gaussian Firefly Algorithm", International Journal of Machine Learning and Computing, Vol. 1, No. 5, December 2011.
- [7] Mohamed Maiza and Mohammed Said Radjef "Heuristics for Solving the Bin-Packing Problem with Conflicts", Applied Mathematical Sciences, Vol. 5, no. 35, 1739 – 1752, 2011.
- [8] Prof. Dr. Karsten Weihe and Prof. Dr. Matthias Müller-Hannemann, "Solution Techniques for specific Bin Packing Problems with Applications to Assembly Line Optimization", 2008.
- [9] Ruslan Sadykov and Francois Vanderbeck, "Bin Packing with Conflicts: a Generic Branch-and-Price Algorithm", 2012.
- [10] Salim Bitam, Mohamed Batouche and El-ghazali Talbi, "A Survey on Bee Colony Algorithms", IEEE ©2010.
- [11] Ullas Mohan, "Bio Inspired Computing", Division Of computer science, SOE, CUSAT, July 2008
- [12] Zhihua Cui, Zhongzhi Shi and Janos Abonyi, "New trends on bio-inspired computation", Transactions of the Institute of Measurement and Control 34(6) 665– 667, 2011.
- [13] <http://www.wiwi.uni-jena.de/Entscheidung/binpp/>