ISSN 2229-5518

# Modified Intelligent Water Drops Algorithm with Tabu Search (MIWD-TS) for solving Multi-objective Optimization

Kavitha, Vijaya

**Abstract –** Multi-objective optimization is widely applied in a number of areas now-a-days. Unfortunately, many combinatorial multi-objective optimization problems are NP-hard. However, it is often unnecessary to have an exact solution. So, heuristic approach can obtain a near-optimal solution in some reasonable time with the smallest possible computational burden. Intelligent Water Drops algorithm (IWD) a new swarm-based optimization algorithm has attracted the interest of researchers due to its intelligent behavior, effectiveness and efficiency in solving numerous Meta heuristic problems. In this, near optimal solutions are obtained by the actions and reactions that occur among the water drops and the water drops with the riverbeds. Since, this is a constructive approach; it may trap into local optimum. In this paper, IWD algorithm is augmented with Tabu Search to find the optimal values of weighted multi-objective functions. It addresses the issues of exploration and exploitation of candidate solutions in order to provide better optimal solution. The proposed algorithm called the MIWD-TS (Modified Intelligent Water Drops with Tabu Search) algorithm is tested for the composition of Intelligent Test Sheet composition problem which is a multi-objective problem. The experimental results prove that the proposed approach performs well in comparison with other approaches as Random Search and Dynamic programming.

**Index Terms** - Intelligent Water Drops, Swarm Intelligence, Meta Heuristic, Weighted Multi-objective optimization, Intelligent Test Sheet



## 1 INTRODUCTION

Optimization forms an important part of our day-to-day life. Multi-objective optimization problems consist of several objectives that are necessary to be handled simultaneously. Such problems arise in many applications, where two or more, sometimes competing objective functions have to be minimized concurrently. Due to the multi criteria nature of such problems, optimality of a solution has to be redefined, giving rise to the concept of Pareto optimality. In contrast to the single-objective optimization case, multi-objective problems are characterized by trade-offs and, thus, there is a multitude of Pareto optimal solutions, which correspond to different settings of the investigated multi-objective problem. Thus, the necessity of finding the largest allowed number of such solutions, with adequate variety of their corresponding properties, is highly desirable. Hence, the goal may be to find a representative set of Pareto optimal solutions or finding a single solution that satisfies the subjective preferences of a human decision maker [8]. Scalarizing a multi-objective optimization problem means formulating a single-objective optimization problem such that optimal solutions to the single-objective optimization problem are Pareto optimal solutions to the multi-objective optimization problem. In addition, it is often required that every Pareto optimal solution can be reached with some parameters of the scalarization [8, 14].

New algorithms have been developed to see if they can cope with these challenging optimization problems intelligently. The natural systems that have developed for so long are one of the rich sources of inspiration for inventing new intelligent systems. In the field of Computational Intelligence, especially Evolutionary Computation and Swarm-based systems, the degree of imitation from nature is surprisingly high and we are in need of developing and proposing new algorithms, which partially or fully follow nature and the actions and reactions that happen in a specific natural system or species. Swarm intelligence is one of the scientific fields that are closely related to natural swarms existing in nature, such as ant colonies, bee colonies, brain and rivers. Among the problem solving techniques inspired from nature are evolutionary, neural networks, time adaptive self-organizing maps, ant colony optimization,


- *Kavitha is currently working as Assistant Professor, Department of MCA, AIMIT, St. Aloysius College (Autonomous), Mangalore, India. E-mail: kavitha.rajamanii@gmail.com*
- *Dr. Vijaya is currently working as Assistant Professor, Department of Computer Science, Government Arts College, Salem-7, India.*

bee colony optimization, particle swarm Optimization, DNA computing and intelligent water drops.

Among the most recent nature-inspired swarm-based optimization algorithms is the Intelligent Water Drops (IWD) algorithm. It is a population based constructive optimization algorithm which has been inspired from natural rivers and exploit the path finding strategies of rivers. A natural river often finds good paths among lots of possible paths in its ways from the source to destination. These near optimal or optimal paths follow from actions and reactions occurring among the water drops and the water drops with their riverbeds. Once an IWD finds an optimal solution, that solution becomes the iteration-best solution in the algorithm and thus the total-best solution is updated to the newly found optimal solution. The convergence in value is proven to exist if the probability of choosing any node of the problem's graph in a solution is nonzero [1-4]. To avoid getting trapped in the local optimum, steps to be taken to explore the search space and exploit the best solution from existing candidate solutions [6]. Tabu Search is a Meta heuristic search which takes a potential problem and checks its neighbours in the hope of finding improved solution [17]. So, this research aims in resolving the issue by using Tabu Search. The proposed approach Modified Intelligent Water Drops with Tabu Search (MIWD-TS) algorithm is tested for a weighted multi-objective optimization problem. The results show that embedding Tabu Search in IWD gives better efficiency.

## 2. Background and Relevant Research

In order to find the optimal solution automatically and effectively, a proper optimization method has to be used. According to the number of requested objectives, optimizations are divided into single-objective and multi-objective. In case of the single-objective optimization, there is usually a single optimal solution. In case of the multi-objective optimization (MOOP), some objectives may be conflicting: when approaching one criterion, other criteria are receding. Then, a single optimal solution can be hardly determined without knowledge of the optimized problem. Merging all objectives into the single-objective function [16] is one of potential solutions. However, the preferences of each objective have to be fixed by proper setting of weighting coefficients of partial criteria in the aggregating objective function. Most of the optimization problems are discrete in nature. Combinatorial Optimization operates on the domain of optimization problems in which the set of feasible solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution. The Combinatorial Optimization has two different approaches with respect to the way of finding the solution as Exact and Heuristic approaches.

Dynamic programming is an exact approach where it is used for optimization problems which can examine all possible ways to solve the problem and will pick the best solution. Therefore, dynamic programming is an intelligent, brute-force method that enables to go through all possible solutions to pick the best one. The critical issue arising here is the exceedingly long execution time required for producing optimal solutions. As the time-complexity of this algorithm is exponential in terms of input data, the execution time will become unacceptably long if the number of candidate items is large. By searching over a large set of feasible solutions, metaheuristics can often find good solutions.

Metaheuristics are strategies that guide the search process and the goal is to efficiently explore the search space in order to find near–optimal solutions. A constructive metaheuristic algorithm builds solutions from scratch by gradually adding solutions' components to the initially empty solutions whereas a local search algorithm starts from a complete solution and then tries to improve it over time [7]. Evolutionary-based algorithms are local search algorithms whereas most of the swarm-based algorithms are constructive algorithms. Metaheuristic approach may either be single solution based or population based. Population-based approaches maintain and improve multiple candidate solutions, often using population characteristics to guide the search; population based metaheuristics include evolutionary computation, genetic algorithms and Swarm intelligence also. Evolutionary Computation, which has been inspired from observing natural selection and reproduction systems in nature, is often used for optimization. Genetic algorithms (GA) are among the most famous algorithms in this regard. Evolution Strategy, Evolutionary Programming and Genetic Programming are other Evolutionary-based intelligent algorithms that are often used for optimization [14]. Genetic Algorithm is the population based evolutionary approach. The response time of the GA algorithm becomes higher if the size of search space grows gradually. The reason is that if there are more candidate test items, much longer chromosomes will be used and the computing time dealing with all bits in chromosomes becomes much longer as well. Hence, other approach need to be selected which can produce near optimal items coping with complexities of time and search space effectiveness.

Swarm-based algorithms have recently emerged as a family of nature-inspired, population-based algorithms that are capable of producing low cost, fast, and robust solutions to several complex problems. Although these are relatively unsophisticated with limited capabilities on their own, they are interacting together with certain behavioural patterns to cooperatively achieve tasks necessary for their survival. One of the famous swarm-based optimization algorithms has been invented by simulating the behaviour of social ants in a colony. They can find the shortest path from their nest to a food source or vice versa that show a high-level of intelligence in a colony of ants. Ant colony optimization (ACO) algorithm and Bee Colony Optimization (BCO) algorithm are among the swarm-based algorithms imitating social insects for optimization. Artificial Immune System (AIS) is another system which follows the processes and actions that happen in the immune systems of vertebrates. Another swarm-based optimization algorithm is the Particle Swarm Optimization (PSO). PSO uses a swarm of particles, which each one has position and velocity vectors and they move near together to find the optimal solution for a given problem. In fact, PSO imitates the processes that exist in the flocks of birds or a school of fish to find the optimal solution [9]. Another swarm-based optimization algorithm is the Electromagnetism-like mechanism (EM) that uses an attraction-repulsion mechanism based on the Coulomb's law to move some points towards the optimal positions [10-12].

Recently, the new metaheuristic algorithm "Intelligent Water Drops," has been introduced in the literature and used for solving various problems like the traveling salesman problem (TSP), N-Queens problem, Multiple Knapsack problem, continuous optimization, etc., It is a population based constructive optimization algorithm which has been inspired from natural rivers and exploit the path finding strategies of rivers [1-5]. In this study, some of the challenges faced by the IWD are overcome by improving it in terms of exploration and exploitation using Tabu search.

## 3. Problem Description

### 3.1 Weighted Multi-objective optimization

The process of optimizing systematically and simultaneously a collection of objective functions are called multi-objective optimization or vector optimization. The general multi-objective optimization problem is posed as follows:

$$\text{Minimize } F(X) = [F_1(X), F_2(X), \dots, F_k(X)]^T \quad (1)$$

$$\text{Subject to,} \quad g_i(X) \leq 0, i = 1,2,\dots,m,$$
$$h_j(X) = 0, j = 1,2,\dots,e$$

Where,

k is the number of objective functions,
m is the number of inequality constraints and
e is the number of equality constraints.

$x \in E^n$ is a vector of decision variables, where n is the number of independent variables $x_i$. $F(x) \in E^k$ is a vector of objective functions $F_i(x) : E^n \rightarrow E^1$. $F_i(x)$ are also called objectives, criteria, cost functions, or value functions.

### 3.2 Weighted Global Criterion Method in L2 Norm

One of the most common general scalarization methods for multi-objective optimization is the global criterion method in which all objective functions are combined to form a single function. Although a global criterion may be a mathematical function with no correlation to preferences, a weighted global criterion is a type of utility function in which method parameters are used to model preferences. One of the most general utility functions is expressed in its simplest form as the weighted exponential sum:

$$U = \sum_{i=1}^{k} W_i [F_i(X)]^p, \; F_i(X) > 0 \; \forall_i \quad (2)$$

The scalar function to be minimized can be represented as the weighted sum of squared relative distances of individual objectives from their goals as expressed below. This is the most common extension of Equation 2.

$$U = \left\{ \sum_{i=1}^{k} W_i \, [F_i(X) - Y_i]^2 \right\}^{1/2} \quad (3)$$

Here, w is a vector of weights typically set by the decision maker such that $\sum_{i=1}^{k} W_i = 1$ and $W_i > 0$. As with most methods that involve objective function weights, setting one or more of the weights to zero can result in weak Pareto optimality where Pareto optimality may be achievable. Generally, the relative value of the weights reflects the relative importance of the objectives.

### 3.3 Basic principles of IWD

In nature, flowing water drops are observed mostly in rivers, which form huge moving swarms. The paths that a natural river follows have been created by a swarm of water drops. For a swarm of water drops, the river in which they flow is the part of the environment that has been dramatically changed by the swarm and will also be changed in the future. One feature of a water drop flowing in a river is its velocity. It is assumed that each water drop of a river can also carry an amount of soil. Therefore, the water drop is able to transfer an amount of soil from one place to another place in the front. Assume an imaginary natural water drop is going to flow from one point of a river to the next point in the front. Three obvious changes happen during this transition:

- Velocity of the water drop is increased.
- Soil of the water drop is increased.
- Between these two points, soil of the river's bed is decreased.

Based on the aforementioned statements, an Intelligent Water Drop (IWD) has been suggested by Shah-Hosseini in the year 2007, which possesses a few remarkable properties of a natural water drop. This Intelligent Water Drop, IWD for short, has two important properties:

- The soil it carries.
- The velocity that it possess.

An IWD moves in discrete finite-length steps in its environment. From its current location i to its next location j, the IWD velocity is increased by an amount which is nonlinearly proportional to the inverse of the soil between the two locations i and j. Moreover, the IWD's soil is increased by removing some soil of the path joining the two locations i and j. The amount of soil added to the IWD is inversely and nonlinearly proportional to the time needed for the IWD to pass from its current location to the next location. The time taken for the IWD to move from location i to j is proportional to the velocity of the IWD, and inversely proportional to the distance between the two locations i and j. Some soil is removed from the visited path between locations i and j. The updated soil of the path is proportional to the amount of soil removed by the IWD flowing on the path joining i to j.

An IWD prefers the paths with low soils on its beds than with higher soils. A uniform random distribution is used among the soils of the available paths such that the probability of the IWD to move from location i to j is inversely proportional to the amount of soils on the available paths. The lower the soil of the path between locations i and j, the more chance this path has for being selected by the IWD located on i. The IWDs work together to find the optimal solution to a given problem. The problem is encoded in the environment of the IWDs, and the solution is represented by the path that the IWDs have converged to.

## 4. Proposed MIWD-TS

### 4.1 Modified IWD Algorithm with Tabu Search

By now IWD algorithm has many features and able to give better heuristic solution for NP hard problems. Sometimes, because of the convergence property, it may lead to trap into a local optimum. After each iteration, the path of the iteration best solution is updated and it may lead to getting trap in the local optimum by paving the way for the next iterations to choose the same path again. This issue can be overcome by embedding Tabu search.

In order to explore the search space efficiently and to exploit the best solution among candidate solutions, IWD algorithm is augmented with a local search operator called Tabu Search. Each iteration of IWD algorithm suggests an iteration best solution $T^{IB}$. Randomly some elements are picked up from the current solution and changed to other elements in the problem space. So, by using Tabu search, while identifying the neighbour of $T^{IB}$, exploration of search space is attained. Also, moves are determined based on the members in the Tabu list. Hence, exploitation of the best solution is attained.

The Swarm-based optimization technique IWD algorithm is modified as Modified IWD with Tabu Seach (MIWD-TS) algorithm by embedding Tabu Search in the step of finding the iteration best solution among candidate solutions.

The proposed MIWD-TS approach in relation with weighted multi objective optimization is specified in the following steps:

1. Scalarize the multi objective problem into single objective by Weighted Global Criterion method in L2 Norm.

$$U = \left\{ \sum_{i=1}^{k} W_i \,[F_i(X) - Y_i]^2 \right\}^{1/2} \quad (4)$$

2. Initialization of static parameters.

The problem is given in the form of graph to the algorithm. The quality of the total-best solution is initially set to the worst value. The maximum number of iterations, number of water drops, and initial soil on each edge, velocity and soil updating parameters are specified.

3. Initialization of dynamic parameters.

Every IWD has a visited node list which is initially empty. Each IWD's velocity is set to the initial value.

4. Spread the IWDs randomly on the edges of the graph as their first visited nodes

5. Update the visited node list of each IWD to include the nodes just visited.

6. Repeat Steps 6.1 to 6.4 for those IWDs with partial solutions.

6.1 For the IWD residing in node i, choose the next node j, which does not violate any constraints of the problem and is not in the visited node list of the IWD, using the probability of node selection. Then, add the newly visited node j to the list.

6.2 Update the velocity of each IWD moving from node i to j

6.3 Compute the soil that the IWD loads from the path based on the heuristic undesirability which is defined appropriately for the given problem.

6.4 Update the soil of the path from node i to j traversed by that IWD and also update the soil that the IWD carries.

7. Find the iteration-best solution $T^{IB}$ from all the solutions $T^{IWD}$ found by the IWDs

7.1 Initialization: Consider the initial solution S as the iteration-best solution $T^{IB}$ and compute the objective function F(S).

7.2 Neighbour Generation: Select neighbour S′ of the current solution S and compute F(S′)

Given the current configuration x=[$x_1$, $x_2$, …, $x_n$], one may randomly pick one item and change its value (from 0 to 1, or vice versa). The neighborhood of the current configuration is identified by executing a fixed number of candidate moves.

7.3 Sort candidate moves according to objective values: All the candidate moves identified in Step (7.2) are sorted in decreasing order according to the values of objective function using their resulting solutions.

7.4 Select the next move: The algorithm selects the best non-Tabu move (in terms of objective value) or the best Tabu move that meets the aspiration level and considers the resulting solution as the new current configuration. The aspiration level adopted is, the Tabu status of a candidate move can be overruled if it leads to a trial solution whose objective value is greater than the best objective value of all previous configurations visited.

7.5 Update the Tabu List:

Determine Δ= F(S′ ) - F(S).

If Δ <0 and S′ is "non-tabu", then a move to S′ is always accepted.

If Δ <0 and S′ is "tabu", then a move to S′ may be accepted for a promising schedule S′

(if F(S′ ) is less than the objective function value for any other solution obtained before).

If Δ >=0 and S′ is "tabu", then a move to S′ is always rejected.

If Δ>=0 and S′ is "non-tabu", then a "wait and see" approach is adopted: Move S′ into Tabu and take the least good solution out from the Tabu.

7.6 Repeat from 7.1 to 7.5 till maximum iterations and select the iteration-best solution $T^{IB}$ in terms of solution quality.

8. Update the soils on the paths that form the current iteration-best solution $T^{IB}$

9. Update the total best solution $T^{TB}$ by the current iteration-best solution $T^{IB}$

10. Increment the iteration and finally provide the total-best solution $T^{TB}$.

## 5. Performance Evaluation

Evaluation of the performance of the MIWD-TS algorithm has been accomplished through an experiment. The Intelligent Test Sheet composition in E-Learning has been taken as the multi objective optimization problem to solve using the proposed method. The Test Sheet is to be composed with personalization ie., based on the ability of the learner. Also, the test items should be able to discriminate the ability of a learner from others. Other multiple criteria like, the concepts to be covered in the test, association between the test items and the concepts, estimated time to complete, etc.,[13, 15]. The Test Sheet composition problem can be formulated as Weighted Multi-Objective optimization problem as follows:

$$\text{Minimize } Z(X) = \left\{ \sum_{i=1}^{k} W_i \left[F_i(X) - Y_i\right]^2 \right\}^{1/2} \qquad (5)$$

With objective Functions,

$F1\,(x) = \sum_{j=1}^{n} df_x$

$F2\,(x) = \sum_{j=1}^{n} ds_x$

Y= [DF,DS]

W=[0.6 0.4] - Weights assigned to the objective functions

Subject to,

$\sum_{i=1}^{n} r_{ij} \geq h_j\,,\ j = 1,\dots, m$

ISSN 2229-5518

Where,
DF = Expected Degree of Difficulty
DS = Expected Degree of Discrimination
$df_x$ = Degree of Difficulty of Test Item x
$ds_x$ = Degree of Discrimination of Test Item x
M= No. of Concepts
$r_{ij}$ = Association of the Test Item i with Concept j
$h_j$ = Lower Bound of the specific Concept j

Therefore, the objective of this model is to select a subset of test items so that the deviation of the expected and actual item parameters to be minimized.

To analyze the comparative performances of several experiments, six item banks with number of test items ranging from 25 to 2000 were constructed in the database. The test organized was relevant to three assigned topics and the corresponding weightages of them were set to w1 = 0.4, w2 = 0.4, and w3 = 0.2, respectively. Meanwhile, the target difficulty level DF was set to 75 and target discrimination level DS to 60. The intelligent test sheet was composed with the number of IWD as 10. The number of iterations made was 10, 25, 50 and 100. The test sheet generation was run for different cases with varied item bank size, iterations as mentioned above. The degrees of difficulty and discrimination of each generated test sheet are compared with the objective requirements. Four test-sheet-generating methods (i.e., random selection, dynamic programming approach and the MIWD-TS algorithm) were employed to compare the solution quality. Each approach was run for 10 times on each item bank and the average was taken.

The average measures of difficulty and discrimination are defined as

$$\text{Difficulty DIF} = \frac{\sum_{i=1}^{N}(|df_i - DF|)}{N} \qquad (6)$$

$$\text{Discrimination DIS} = \frac{\sum_{i=1}^{N}(|ds_i - DS|)}{N} \qquad (7)$$

where DF and DS are the target expected degree of difficulty and discrimination level.

The solution quality is found as the sum of absolute deviation of expected and obtained test sheet parameters. The Table 1 shows the performance of MIWD-TS with different item bank size and iterations.

TABLE 1.
PERFORMANCE OF MIWD-TS

| Item Bank Size | MIWD-TS | | | |
|---|---|---|---|---|
| | 10 Iterations | 25 Iterations | 50 Iterations | 100 Iterations |
| 25 | 6.02 | 5.65 | 5.87 | 5.66 |
| 50 | 4.65 | 4.21 | 4.04 | 3.52 |
| 100 | 3.17 | 3.46 | 3.15 | 3.03 |
| 500 | 2.20 | 3.17 | 2.26 | 2.13 |
| 1000 | 1.20 | 1.32 | 1.25 | 0.90 |
| 2000 | 1.08 | 0.92 | 0.41 | 0.35 |

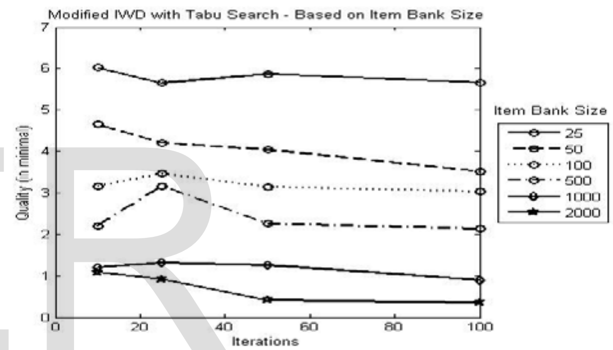

Fig. 1 Performance of Modified IWD-TS

The figure 1 shows that there is a gradual increase in the solution quality when there is increase in the number of iterations and item bank size. Regardless of item bank size, the quality is worsening if the number of iterations is 10. Also, the quality is fluctuating when there is a small item bank size like 10 and 25.



TABLE 2
COMPARISON OF PERFORMANCE OF IWD AND MIWD-TS.

| Item Bank Size | IWD | MIWD-TS | IWD | MIWD-TS | IWD | MIWD-TS | IWD | MIWD-TS |
|---|---|---|---|---|---|---|---|---|
| | 10 Iterations | | 25 Iterations | | 50 Iterations | | 100 Iterations | |
| 25 | 6.78 | 6.02 | 5.70 | 5.65 | 6.78 | 5.87 | 6.70 | 5.66 |
| 50 | 4.60 | 4.65 | 3.26 | 4.21 | 4.14 | 4.04 | 3.72 | 3.52 |
| 100 | 3.20 | 3.17 | 4.48 | 3.46 | 3.20 | 3.15 | 3.38 | 3.03 |
| 500 | 2.28 | 2.20 | 3.82 | 3.17 | 2.28 | 2.26 | 3.12 | 2.13 |
| 1000 | 1.20 | 1.20 | 1.14 | 1.32 | 1.70 | 1.25 | 0.90 | 0.90 |
| 2000 | 0.42 | 1.08 | 1.70 | 0.92 | 0.42 | 0.41 | 0.40 | 0.35 |

ISSN 2229-5518

The above table 2, compares the performance of the existing IWD and the modified IWD embedded with Tabu Search. Apart from smaller item bank size and minimum number of iterations, the proposed approach performs well. In comparison with the existing approach, the proposed approach is acceptable in terms of effective solutions.

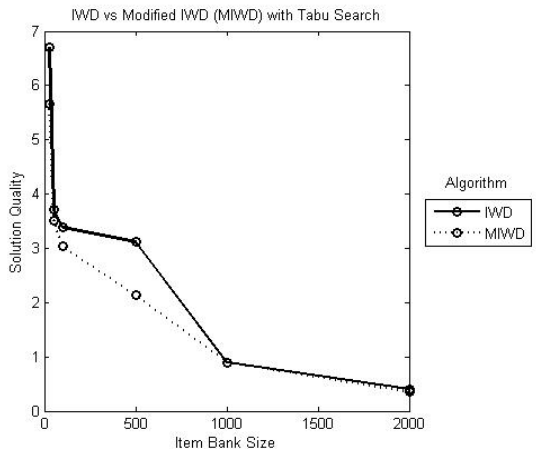


Fig. 2 Performance Comparison of IWD and MIWD-TS

The above Figure 2 shows that there is a gradual improvement in the quality by the proposed Modified IWD than the existing IWD. Table 3 consolidates and compares the efficient performance of MIWD-TS with other approaches like Random Search and Dynamic Programming.

It is obvious from the Table 3, irrespective of item bank size there is no better quality with the Random Search. Dynamic Programming is infeasible for large item bank having size more than 5000. In all the ways, the proposed approach performs well than any other approaches.



TABLE 3
COMPARISON OF MIWD WITH RANDOM AND DYNAMIC PROGRAMMING

| Item Bank Size | Random Search | Dynamic Programming | MIWD | | | |
|---|---|---|---|---|---|---|
| | | | 10 Iterations | 25 Iterations | 50 Iterations | 100 Iterations |
| 25 | 16.72 | 6.10 | 6.02 | 5.65 | 5.87 | 5.66 |
| 50 | 11.42 | 5.90 | 4.65 | 4.21 | 4.04 | 3.52 |
| 100 | 12.34 | 5.21 | 3.17 | 3.46 | 3.15 | 3.03 |
| 500 | 09.26 | 3.64 | 2.20 | 3.17 | 2.26 | 2.13 |
| 1000 | 16.62 | 2.94 | 1.20 | 1.32 | 1.25 | 0.90 |
| 2000 | 11.96 | 2.10 | 1.08 | 0.92 | 0.41 | 0.35 |

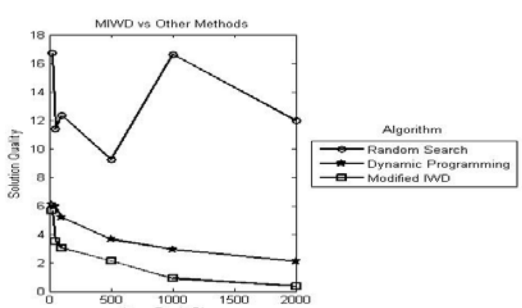


Fig. 3 Performance comparison of MIWD-TS with Random and Dynamic Programming

The plot in Figure 3 compares the solution quality in terms of minimized objective function of three approaches namely Random Search, Dynamic Programming approach and the proposed MIWD-TS algorithm. It shows that the average solution quality of the MIWD-TS search were very close to the optimal solutions as expected, while the quality of average solution value obtained by the random search was significantly worse.

## 6. Conclusion

In this research study, the existing IWD algorithm which is a swarm based optimization technique is modified as Modified Intelligent Water Drops algorithm with Tabu Search to solve the weighted multi-objective optimization problems. The multi-objective functions are scalarized using weighted global criterion of L2 Norm. The IWD algorithm is modified to exploit the good solution among the candidate solutions. The search space is efficiently explored without trapping into local optimum. The modification has been done as embedding Tabu Search to exploit the better solution and explore the search space efficiently. Intelligent Test Sheet composition problem has been taken as the study having two weighted objective functions satisfying multiple criteria together. The experimental studies shows that the proposed MIWD-TS algorithm gives better performance in producing near optimal quality Test Sheets than the IWD algorithm. Also, the proposed approach is compared with other techniques like Random Search and Dynamic Programming and the performance is proved to be better. Other mechanisms that exist in natural rivers or devising local heuristics that fit better with the IWD algorithm can be considered.

## Reference

[1] Hamed Shah-Hosseini. “Intelligent Water Drops Algorithm – A new optimization method for solving the Multiple Knapsack Problem”, International Journal of Intelligent Computing and Cybernetics. Vol 1. No 2. 2008. p. 193-212

[2] Hamed Shah-Hosseini. “The Intelligent Water Drops Algorithm: A nature inspired swarm-based optimization algorithm”, International Journal of Bio-inspired computation. Vol 1. No 1/2. 2009. P. 71-80

[3] Kesavamoorthy R., Arun Shanmugam D., Thanga Mariappan L. “Solving Traveling Salesman Problem by Modified Intelligent Water Drops Algorithm”, International Conference on Emerging Technology Trends. 2011.

[4] Hamed Shah-Hosseini. “Optimization with Nature inspired Intelligent Water Drops Algorithm”, Evolutionary Computation. Edited By, Wellington Pinheiro dos Santos. 2009

[5] Hamed Shah-Hosseini. “Problem solving by Intelligent Water Drops algorithm”, IEEE Congress on Evolutionary Computation. 2007

[6] Yenny Noa Vargas., Stephan Chen. “Particle Swarm Optimization with Resets – Improving the Balance between Exploration and Exploitation”, MICAI. Part II. Edited By, G. Sidorov et al. Springer-Verlag Berlin Heidelberg 2010

[7] Zahra Beheshti, Siti Mariyam Hj Samsuddin. “A Review of Population-based Meta-Heuristic Algorithms”, International Journal of Soft Computing and Applications. Vol 5. No 1. 2013

[8] Ivan P. Stanimirovic., Milan Lj Zlatanovic., Marko D Petkovic. “On the Linear Weighted Sum Method for Multi-Objective Optimization”, Series of Math Inform. Facta Universitatis. 2011

[9] Frans Van den Bergh. “An Analysis of Particle Swarm Optimizers”, Thesis in Unversity of Pretoria etd. 2006

[10] Ibrahim M Hezam., Osama Abdel Raouf., Mohey M Hadhoud. “A new compound Swam Intelligence algorithms for solving Global Optimization problems”, International Journal of Computers and Technology. Vol 10. No 9. 2013

[11] Iztok Fister Jr., Xin-She Yang., Iztok Fister., Janez Brest., Dusan Fister. “A Brief Review of Nature-inspired Algorithms for Optimization”, Electotehiniski Vestnik English Edition. Vol 80. No 3. 2013

[12] Hifza Afaq., Sanjay Saini. “On the solutions to the Travelling Salesman Problem using Nature Inspired Computing Techniques”, International Journal of Computer Science Issues. Vol 8. Issue 4. No 2. 2011

[13] Tsu-Feng Ho., Peng-Yeng Yin., Gwo-Jen Hwang., Shyong Jian Shyu., Yan-Nan Yean. “Multi-Objective Parallel Test Sheet Composition using Enhanced Particle Swarm Optimization”, Journal of Educational Technology and Society. Vol 12. No 4. 2009

[14] Deb K. “Multi-objective optimization using Evolutionary algorithms”, West Sussex, England. John Wiley and Sons. 2001

[15] Shu-Chen Cheng., Yen-Ting Lin., Yueh-Min Huang. “Dynamic question generation system for web-based testing using particle swarm optimization”, Journal of Expert Sysytems with Applications. Vol 36. Issue 1. 2009

[16] Kalyanmoy Deb, “Multi objective optimization using Evolutionary Algorithms: An Introduction”, Kanpur Genetic Algorithms Laboratory, Report Number 2011003.

[17] Gwo-Jen Huang, Peng-Yeng Yin, Shu-Heng Yeh, “A Tabu Search approach to generating Test Sheets for Multiple Assessment Criteria”, IEEE Transactions on Education, Vol. 49, No. 1, 2006