**Social Class multi objective Particle Swarm Optimization SC-MOPSO for Variable Length Problems**

**Abstract**

Meta-Heuristic searching optimization (MHSO) is set of random searching algorithms with heuristic nature. It boosts the performance of random searching and exploits the hardware computation for solving NP-hard optimization problems. Two aspects when added together to optimization problems require special types of MHSO algorithms. Firstly, the multi-objective nature which is defined by having more than one objective for optimization with implicit conflicting nature. Secondly, the variable length aspect which is defined by having a solution space with different lengths of solutions. Consequently, direct solution interaction and ranking for random heuristic searching becomes problematic. In this article, social class multi-objective particle swarm optimization (SC-MOPSO) was developed for solving difficult optimization problems with multi-objective and variable length nature. The algorithm extends the concept of social interaction of particle swarm optimization (PSO) by decomposing the solution space into classes based on their dimension. Next, it enables two modes of interaction: the first one is particles interaction within one class based on the selected exemplar and the second one is classes interaction by moving solutions from one class to another. The article uses novel concepts in this domain, mainly, adjacency matrix, reduced adjacency matrix and probability density function. In addition, it adds histogram of Pareto front solutions dimensions as a novel performance metrics for evaluation. Furthermore, a special variant of the algorithm for solving mobile sink deployment problem in the area of wireless sensor network WSN was developed under the name of mobile sink SC-MOPSO (MS-SC-MOPSO) by incorporating an operator for variable length solution interaction and a technique for progressive evaluation of solutions while generation. The article has used for evaluation novel set of mathematical optimization problems with two objectives and three objectives based on different dimensions of mathematical functions. Besides, SC-MOPSO and the benchmarks were evaluated for accomplishing WSN deployment with two objectives. Furthermore, MS-SC-MOPSO was evaluated and compared with heuristically modified (MOPSO) and (NSGA-II) based on synthetically generated WSN environment. In all evaluations SC-MOPSO and MS-SC-MOPSO have shown superiority over (MOPSO) and (NSGA-II) with respect to almost all MOO evaluation metrics. (add sentences)

# Introduction

Meta-heuristic based optimization has proven its effectiveness in solving many complex and NP-hard problems. It has been applied to various real world applications in many fields such as computer science[1], robotics[2], communication[3] , networking [4], manufacturing [5]…etc. Furthermore, it has been recognized as an effective solution for many artificial intelligence AI problems such as optimizing neural networks[6], selecting more discriminative features[7] …etc.

Despite the fact of the superiority of models based on meta-heuristic searching for optimization over other approaches, there is still many challenges and unsolved problems. One of them is the variable length of decision space in the meta-heuristic optimization[8]. It is meant by variable length of decision space, when the candidate solutions differ in the dimensions. Such phenomenon is observed in many practical fields. In the wireless communication, when we are required to deploy set of sensors in an environment we aims at minimizing the non-covered areas, the cost and the energy consumption. This causes set of non-inferior solutions due to the conflicting nature in the objective. This gives the multi-objective aspect of such application. On the other hand, the variable length aspect is witness due to the relation between changing the values of the objectives and the number of deployed sensors which leads to solutions different in lengths (citation). Similarly, in the application of mobile sink path planning, we aim at selecting the best locations of rendezvous points that accomplish less energy consumption, higher covered area and faster data collection time. Hence, the problem has a multi-objective nature. On the other hand, changing the number of rendezvous points causes changes in the objective values which leads to length variability in the solution space. Consequently, the problem is described as variable length problem (citation). Another example is the community detection problem in the social networking. Such problem is an optimization problem with multi-objective nature because it aims at maximizing the intra-similarity and minimizing the inter-similarity of the communities simultaneously. On the other hand, the problem is regarded as variable length problem because changing the number of detected communities implies changing the length of the solution (citation). Similarly, the problem of length variability in the optimization is witnessed in the optimization of the topology and weights of neural network. Such problem aims at modifying the number of neurons and the links weights in order to maximize the accuracy of the neural network and minimize the training and prediction time. Such optimization is described as multi-objective because it has two objectives. In addition, it is described as variable length because different number of neurons implies different solutions lengths (citation).

The typical way that the researchers use for the searching in the variable length optimization problems is based on some simplification assumptions to convert the problems to fixed length [9]. However, this is not considered as a good or effective way of exploring the solution space. One reason is of the dependency of the performance on the length of the solution [8]. We add to that the ignorance of the positions of the decision variables in the solution while performing in the interaction. Furthermore, the literature of meta-heuristic optimization is still poor in tackling dealing with of the length variability and the multi-objective aspect together although they appear in many problems [10]. Some of the real world problem is the wireless sensor network deployment where the system builder has to decide the number of nodes, their positions and types in order to maintain maximum coverage with minimum cost in a given environment[11]. Another example related to the this field is also the mobile sink planning where the optimization algorithm has to define the best rendezvous points for mobile sink to accomplish minimum traveling time for the sink and minimum energy consumption for the sensors[12]. This problem is not only faced in wireless sensor network field. We encounter it in many real world applications such as data clustering where the variable length optimization algorithm has to decide the best number of clustering with the minimum intra and maximum inter distance of clusters’ points[13]. Fourth example is community detection where the number of community and their definition which is variable length has to maintain various objectives at the same time, e.g., maximum modularity and normalized mutual information metric[14].

The lacking of focus on variable length problems is observed in the evolutionary based approaches such as genetic algorithms and in the swarm-based approaches such as particle swarm optimization except one recent articles [15]. Furthermore, to the best of our knowledge, there is no single research work that has tackled multi-objective optimization with variable length of decision space in the particle swarm framework. Hence, the goal of this article is to propose a novel multi-objective variable length particle swarm optimization based on the concept of class decomposition. The approach is named as Social Class multi objective Particle Swarm Optimization for Variable Length Problem (SC-MOPSO). The remaining of the article is organized as follows. In section -2-, we present the literature survey. Next, a background of the variable length multi-objective optimization problem is presented in section -3-. Next, the methodology is provided in section -4-. Afterwards, the conclusion and future work is provided in section 5.

# Literature Survey

The literature of variable length optimization has been developed more in the evolutionary optimization field. [10]have pointed out that the challenge in variable length multi-objective evolutionary algorithm MOEA is the design of the mating operators such as off-spring and the mutation operators. This is because such operators have to destruct the solutions and recombine them, however, due to the variability in the length of the solutions, such process is not straightforward nor systematic. [16]have stated that using NSGA-II does not lead to the entire optimal Pareto front due to non-wanted interaction between the variable solutions and the crowding distance operator. As an example, the solutions that are smaller in length causes coarse granularity in the objective space and the crowding distance comparing with longer ones. This causes difficulties in the exploration of the solution space. The term metameric has been coined by [17] to indicate to decision vectors that are combined of individual segments with varying length. They also discussed how operators can be modified in order to handle the challenge of solutions variability in length and the existence of metameric variables. Also, they put two conditions on the recombination in the case of metameric: firstly, it has to perform respectful operators. The concept of respectful is introduced by them to the first time and they defined as “respectful recombination operator is one that produces children containing any schemata shared by both parents”, secondly, it has to minimize the risk of disruption which was defined as the case of lacking full inheritance of building blocks from the parents. On the other hand, they put a condition of meeting a level of length-based diversity in the searching. Another concept that was introduced by them is the helper objective which aims at minimizing or maximizing the number of meta-variables. . Some researchers have proposed using the helper function as a guide for favouring solutions according to the length of the solution. In their work,[17] they have proposed three variable length benchmarking problems, namely, sensor coverage problem, wind farm layout and composite laminate stacking problem. They also compared various genetic variants with different combination and mutation operators on them. They found out that metameric genetic algorithm is superior over fixed size genetic for sensor coverage problem except for cut-and slice operator and they found out that the spatial recombination problem is the most superior for sensor coverage problem. Similarly, in the work of [18], several niches based selection operator was proposed for metameric problems. In addition, a window function determines at which lengths a niche is formed. Next, Local selection is then applied within each niche independently, resulting in a new parent population formed by a diverse set of solution length. However, this approach is still regarded as multi-objective with weighted average of objectives which does work well in the case of non-convex optimization surface. In the work of [8] a new evolutionary variable length multi objective optimization problem was proposed where two level decomposition based on penalty boundary intersection were proposed. It decomposes the problem into set of single objective optimizations in the global region in the first level and into multi-objective optimization in the local region in the second level. Another criterion that was used is allowing longer life time for solutions with higher length even if their objective values was not great enough. This consideration is to enable the best of them in the evolutionary process of the optimization.

One of the recent and rare works of variable length particle swarm optimization is the work of [15] where the authors have proposed variable length particle swarm optimization(VL-PSO) to apply it on the problem of feature selection. The issue when using PSO for variable length searching is the embedded constraint the in the velocity calculation of the particle that requires an exemplar with the same dimension of the particle. This was considered, with adding a new concept named probability of selecting the exemplar where the exemplar of the particle is selected according to its fitness value as its own best or another particle. The probability of selecting exemplar is based on the fitness value, the particles with better fitness value are more likely to select their own best as exemplar. On the other side, particles with lower fitness values are more likely to select exemplar from other particles best values. However, this selection is based on 2 tournaments to select the best exemplar out of the two candidates which leads to less exploitation. In the work of [19], another variant of variable length PSO was proposed for the purpose of clustering. The specific application was Identifying Non-redundant Gene Markers from Microarray Data. However, the algorithm uses the number of clusters as input variable. In the work of [20], another variant of variable length PSO was used to optimize extreme learning machine. The optimization solution is variable length due to the change in the number of neurons in the hidden layer of the neural network. Hence, the usage of variable length PSO is needed. The constraint of equality between the solution size and its exemplar is solved by selecting a subset from the biggest size and use it to move its corresponding components from the solution. A similar work for using variable length in the context of optimizing ELM is the work of [21] where the authors have developed class specific cost regulation extreme learning machine using optimization with one type of meta-heuristic named brain storm optimization algorithm BSO. Regardless of the concept of brain storm and its inspiring to conduct random searching under meta-heuristic, the variable length aspect was only created by merging only genes in the matching positions. This approach has ignored all the aspects that were discussed in the work of[17]. Another work that has claimed to use variable length particle swarm optimization for solving a multi-objective optimization is the work of [22] where the optimization was used for planning three-dimensional based network. However, in their work, the variability aspect of the length of solution was resolved by changing the number of deployed base stations by decreasing one every time and repeating the optimization algorithm. This means there was not change of the internal optimization algorithm which is what needed for variable length.

Overall, non-of the developed approaches for handing variable length optimization problem based on particle swarm optimization was multi-objective. Furthermore, majority of the variable length approaches were focusing on the evolutionary type of optimization methods MOEA. In addition, all the particle swarm based optimization approaches in particular have ignored developing mobility operators that has an awareness of variable dimension in the space with oriented operators to handle that in an optimum way.

This article proposes a novel multi-objective particle swarm optimization for variable length space with variable dimension awareness. It inspires the social metaphor of PSO from the aspect of social class existence. Hence, set of solutions that belong to the same dimension are considered one class members.

# Background

This section provides the background of the variable length multi-objective optimization. Firstly, we present the decision variable in sub-section 3.1. Secondly, we present the metameric variable in sub-section 3.2. Thirdly, we provide the metameric optimization problem in sub-section 3.3. Next, we provide a review about particle swarm optimization and multi-objective particle swarm optimization in subsection 3.4.

## Problem Formulation and Decision Variable

We assume that we have a set of mathematical functions that we call objectives defined as

defined over a searching space . The goal is to determine the best set of non-dominated solution Pareto Front (PF) , where , . The goal is to determine that achieves other close values to the true , with higher spread and higher number of non-dominated solutions. The performance is measured by achieving a higher set coverage, hyper volume, lower relative generational distance and delta metric, and flatter histogram of dimensions of solutions and number of non-dominated solutions.We call each variable decision variable and it represents the atomic variable in the decision space for changing the objective functions.

## Metameric variable

Each set of decision variable that belong to the same realistic meaning or describes one particular variable (scalar or vectorized) is named as metameric variable. As an example, the location of certain sensor in the 2D deployment in WSN network is one metameric variable and it is combined of two decision variables x and y of the sensor. Mathematically, this is expressed by the equation )1(

(1)

It includes metameric variable and each metameric variable has decision variable, and the length of the vector is fixed

## Fixed Length Metameric Optimization Problem

It is an optimization problem with a decision vector combined of more than one metameric variable and with different or same sizes of them and fixed overall size. Considering the representation of -1-, when  is fixed for all solutions then the problem is fixed length metameric optimization length.

## Variable Length Metameric Optimization Problem

It is an optimization problem with a decision vector combined of more than one metameric variable and with different sizes of them with the dependency of the size of the vector on the content of its elements. Considering the representation of -1-, the problem is considered to be variable length when the solution has a length .

## Particle Swarm Optimization

The particle swarm optimization is one meta-heuristic algorithm invented by[23] based on the concept of social swarm evolving. Basic idea of particle swarm is inspired by social interaction that provide an aggregated smart behaviour. A simulation done to swarm of bird has been used to prove the idea. The swarm of birds is combined of set of entities each is moving with two components of velocities: the first one is to pull them toward a best global entity while the second is to pull them toward best local entity. This is depicted in the equations )2( and )3(.

(2)

(3)

The pseudo code of the original particle swarm optimization is given in table-1-. The moving of the particles is done particularly in lines 8-9. The approach considers two data entities: the first one is the repository that is responsible on storing the set of non-dominated solutions. The second one is the memory that is responsible of storing the best position of the swarm of the particles. Assuming that the repository is with the size of REPS, and the memory is with the size of NP, the two data structures are REP[h] and swarm[i]. In addition, the approach quantizes the solutions in the solution space using hypercube of resolution GridRes. Thus, it tracks the number of solutions in each hypercube and performs a fitness sharing using constant z. This enables the approach to favour hypercubes with less number of solutions than hypercubes with higher number of solutions. For selecting the leader, random number is generated using roulette wheel model built using the fitness values of the objective functions and the fitness sharing. The approach originally does not include mutation; however, we add it to increase the diversity of the space.

Table 1 pseudocode for classical multi objective particle swarm optimization

|  |
| --- |
| **Input**  MAX //maximum number of iterations  W // inertia  C1 //constant for adding the influence of memory  C2 //constant for adding the influence of repository  REPS //size of repository  GridRes //resolution of hypercube  OF //objective functions OF= (f1, f2,...fm)  MR //mutation rate  NP //number of particles  X //boundaries of solution space  z //constant z > 1  **Output**  PF //Pareto front  Start  1-Swarm=initialize(NP,X)  2-PBest=Swarm  3-[SwarmObjectives,Swarm]=Evaluate(Swarm,OF)  4-[SwarmObjectives,Swarm]=sort(SwarmObjectives,Swarm)  5-Rep=add(Swarm,GridRes)  5-for t=1 until MAX    6- for i=1 until NP  7- h=RouletteWheel(Rep,z)  8- VEL[i]= W x VEL[i]+ R1 x (PBESTS[i]- Swarm[i]) + R2 x (REP[h]- Swarm[i])  9- SwarmNew= Swarm[i]+VEL[i]  10- R=random(0,1)  11- If(R< MR)  12- (SwarmNew=mutation(SwarmNew))  13- end  14- [SwarmObjectivesNew[i],SwarmNew]=Evaluate(Swarm[i],OF)  15- if(SwarmObjectivesNew[i] is dominant over SwarmObjectives[i])  16- Swarm[i]=SwarmNew;  17- PBest=SwarmNew;  18- end  19- end  20- [SwarmObjectives,Swarm]=sort(SwarmObjectives,Swarm)  21- Rep=add(Swarm,GridRes)  22- end  23-PF=Rep;  24-End |

# Methodology

This section provides the developed methodology for building CVL-MOPSO. It starts with the definition of the class concept in PSO in sub-section 3.1. Next, we present the class matching condition in sub-section 3.2. Afterwards, we provide class diversity criterion in sub-section 3.3. The class diversity is provided in sub-section 3.4. Next, we present the class distribution and killing in sub-section 3.5. Afterwards, we provide the leaders selection in sub-section 3.6. Next, we present the mutation process in subsection 3.7. Next, the prioritizing in selecting the non-dominated solution is given in sub-section 3.8. Lastly, we provide the general algorithm in sub-section 3.9.

## Definition

The pool of particles will be partitioned into subset of classes, where each class includes only particles from the same length. Any two particles with different lengths belong to different classes. Hence, the intersection between any class and another is empty set or . This is depicted in equations 4 and 5.

(4)

(5)

## Class Matching Condition

It states that particles from the same class only are eligible to influence each other. I.e., any solution will select a leader from its own class. The leader is unique in the class. This is shown in equations 6 and 7.

(6)

(7)

, belong to the same class of

## Class Diversity Criterion

The solutions are generated based on class diversity criterion which states that generating the initial pool of swarm is based on covering the whole possible number of lengths of particles equally. Assuming that the solutions space can take one of possible lengths as , … , then the initialization of particles has to follow the constraint of having number of particles for each class is and it equals to . In order to meet this criterion, we use the following process when generating any particle at the first swarm. Firstly, we generate random number between 1 and the number of dimensions. Secondly, we use the generated number to decide in which dimension the particle will be placed. The pseudocode of accomplishing this is given in table -2-.

Table -2- pseudocode of generating the first swarm and distributing it on the number of classes

|  |
| --- |
| **Input**  NumOfParticles  Boundary  DimensionsRange  **Output**  Swarm  **Start**  1-for i=1 until NumOfParticles  2- Dimension=genRan(DimensionsRange)  3- particle=genRan(Dimension,Boundary)  4- add particle to swarm  5-End |

## Adjacency Matrix (AM)

To the best of our knowledge, this article is the first in introducing the concept of adjacency matrix in the context of multi-objective meta-heuristic searching optimization. The role of it is to characterize the state of the optimization at certain iteration in terms of the objective functions values and its relation with the various particles. Hence, we define the adjacency matrix with the dimension of where denotes the number of objectives and denotes the number of particles.

(8)

Where denotes the value of objective function for particle .

In addition, we extract from subset of matrices where each one is with size where denotes the number of particles inside the class .

## Reduced Adjacency Matrix (RAM)

The reduced Adjacency matrix is another type of adjacency matrix that is used to characterize the state of the optimization at certain iteration in terms of the objective functions values and its relation with the various classes.

(9)

Where denotes the value of objective function for class .

## Probability Density Function of Exemplar Selection

The probability density function (PDF) of selecting the exemplar is based on the fitness value of the considered objective for the selection. Assuming that the particle that is subject to mobility is in the class and the considered objective is then we use the row of . The pdf is calculated using the equation

(10)

This equation is to assure

## Probability Density Function of New Class Selection

The probability density function of selecting the exemplar is based on the fitness value of the considered objective for the selection. Assuming that the considered objective is then we use the row of . The pdf is calculated using the equation

(11)

This equation is to assure

## Moving Threshold

This parameter is a pre-defined parameter to decide for how long the particle will be given a chanced in certain class to improve itself, otherwise, it will be moved to a new class. The selection of this parameter is tunable based on the nature of the problem. Another name that can be used for is the enhancement timeout.

## Class Minimum Threshold

This parameter is a pre-defined parameter to decide the lower allowed number of particles in a given class. In order to keep the class alive which provides more diverse searching results in the algorithm.

## General Algorithm

This section presents the general algorithm of Social Class multi objective Particle Swarm Optimization for Variable Length Problem SC-MOPSO. The algorithm is based on distributing the particles among the classes. At first, the particles will be distributed using random number generator that places the particle in the space and another random number generator that places the particle inside one of the classes. In this context, the class is defined as sub-set of the solution space that contains only solutions from the same dimension and all other dimensions solutions exist in other class.

The process of first swarm initialization exists in the pseudocode from line 3 until 8. The command in line 8 is responsible of distributing the solutions among the classes. Next, the algorithm runs for number of iterations according to the NumOfIterations. At each iteration, the algorithm goes over the classes one by one. For each class, the algorithm goes over the particles one by one and select for each particles an exemplar using the function selectExemplar(Classes(classIndex). The exemplar of any particle is selected from the same class of the particle. Next, the algorithm moves the particle based on its exemplar and the input parameters to a new position. After moving the particle, the algorithm checks about any improvement in the fitness functions that might occur.

In case no improvement, then the algorithm will increase the non-improvement counter with one value. Otherwise, the improvement counter will be reset. The role of the improvement counter is to enable the logic of moving the particle from one class to a new class if no improvement occurred for certain number of iterations. The logic of moving particles from any class to another class is conditioned with having minimum number of particles inside given class.

Table 3 pseudocode of Social Class multi objective Particle Swarm Optimization for Variable Length Problem SC-MOPSO

|  |
| --- |
| **Input**  NumOfParticles  NumOfIterations  Boundary  DimensionsRange  ObjectiveFunctions  Inertia,C1,C2  movingThreshold  classMinThreshold  **Output**  ParetoFront  ParetoSet  Start   1. initSwarm(NumOfParticles) 2. numOfClass=DimensionsRange.Max-DimensionsRange.Min+1; 3. for i=1 until NumOfParticles 4. Dimension=genRan(DimensionsRange) 5. particle=genRan(Dimension,Boundary) 6. add particle to swarm 7. End 8. Classes=Distribute(swarm) 9. for iteration=1 until NumOfIterations 10. for classIndex=1 until length(Classes) 11. for each particle of Classes(classIndex) 12. exemplar=selectExemplar(Classes(classIndex) //-1- 13. newParticle=moveParticle(particle,exemplar,Inertia,C1,C2) 14. if(NoImprove(newParticle,particle)) 15. particle.counter=particle.counter+1; 16. else   particle.counter=0;   1. particle=newParticle; 2. end 3. end 4. end 5. for classIndex=1 until length(Classes) 6. for each particle of Classes(classIndex) 7. if(particle.counter>movingThreshold and length(particle.Class)>classMinThreshold) 8. Class=selectNewClass(particle) //2 9. particle=moveToNewClass(particle,Class) 10. end      1. end 2. end 3. end 4. End |

## Exemplar Selection Algorithm

This section presents the algorithm that is used for exemplar selection inside a given class. It uses the adjacency matrix for the class . It builds a probability density function pdf from the AMC matrix according to the fitness value of a certain objective function. The algorithm also takes as input the corresponding objective function that will be used for building the pdf. After building the probability density function, it is used for generating the index of the exemplar that indicates to the solution with the highest fitness value among the solutions with high probability. The pseudocode of the algorithm is given in table 4.

Table 4 pseudocode for exemplar selection algorithm

|  |
| --- |
| **Input**    objectiveFunctions  Class    **Output**  exemplar  **Start**   1. ObjectiveFuncIndex=generateRan(objectiveFunctions.Range); 2. AMC=claculateAMC(Class) 3. pdf=GenerPDF(AMC,objectiveFunctions,ObjectiveFuncIndex); 4. exemplar=genrateRand(pdf) 5. **End** |

## New Class Selection Algorithm

This section presents the class selection algorithm. This algorithm is responsible of selecting a new class and moving some particles that are not improving from their current class to a new class. The pseudocode of the algorithm is given in table 5. The approach is based on the reduced adjacency matrix . It considers a given objective function, and it generates a probability density function pdf based on the corresponding row and using equation 11. Next, it generates the new class according to the pdf.

Table 5 pseudocode of selecting a class for moving particle from current class to a new class because of non-improving

|  |
| --- |
| Input  objectiveFunctions  Classes    Output  Class  Start   1. ObjectiveFuncIndex=generateRan(objectiveFunctions.Range); 2. RAM=claculateRAM(Classes) 3. pdf=GenerPDF(RAM,objectiveFunctions,ObjectiveFuncIndex); 4. Class=genrateRand(pdf) 5. End |

## Evaluation Metrics

1. **Set Coverage**

This measure is also called C-metric, which compares the Pareto sets and as presented in equation12.[24]

12

Where

Indicates to the domination of over

C equals the ratio of non-dominated solutions in that are dominated by non-dominated solutions in to the number of solutions in. Thus, when evaluating a set , minimizing the value of for all Pareto sets is important.

1. **Hyper Volume**

This measure aims to provide an indicator to the spread of solutions in the objective space. It computes the volume of the dominated portion of the objective space relative to a worst solution (reference point); this region is the union of the hypercube whose diagonal is the distance between the reference point and a solution from the Pareto set [24]. Increasing the values of this measure indicator implies a desirable Pareto front. HV is given by Equation 13

13

1. **Number of Non-dominated Solutions**

The number of non-dominated solutions which express the effectiveness of the optimization algorithm can be calculated as the cardinality of .

14

Increasing the number of non-dominated solutions implies better MOO performance.

1. **Relative Generational Distance**

This measure is also called the GD-metric, which is used to evaluate the performance of an obtained Pareto set in comparison with a reference point set [a true Pareto set ][25].

This measure is based on the distance between obtained solutions and reference points, which are calculated by Equation 15 as follows:

15where is the number of solutions in the Pareto set, is the true Pareto front, and is the Euclidean distance between the solutions in and the nearest solutions in . The value of GD should be minimized, which means closer solutions to the true Pareto.

1. **Histogram of Solution Distribution**

This measure shows the diversity of the solutions in the Pareto front from the perspective of dimension. It counts the number of solutions in each dimension. The more equally distributed the better performance as it provides a flexibility of choices to the decision maker.

## Evaluation Benchmarking Functions

In order to evaluate our developed SC-MOPSO, we use set of variable length MOO mathematical functions. Each function is defined based on its objectives and decision space. The first problem that will be used is wireless sensor deployment problem which is defined as provided in sub-section 3.10.1.

### Mathematical Functions

This section presents the mathematical functions problems that are used for evaluation. It contains different combinations of 7 mathematical functions as multi-objective functions. They are taken from (citation). All the problems will deal with the functions as variable dimensions between 1 and 30. The functions are given in the table 6. For each of the function, we determine the boundary of the decision variables. Also, we determine in the table the types of the function: convex vs. non-convex, unimodal vs. multi-modal, continuous vs. non-continuous, differentiable vs. non-differentiable. As we observe in the table, we have selected various types of functions: some of them are convex, continuous and differentiable such as sphere. Others are continuous, differentiable and non-convex such as Rosenbrock’s function, Ackley’s function, Griewanks’s function, Rastrigin’s function and Noncontinuous Rastrigin’s function. Others are continuous, non-differentiable and non-convex such as Weierstrass function.Also, we have added to the table a challenging function because of non-continuity, non-differentiability, and non-convexity named as Noncontinuous Rastrigin’s function.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NU** | **Name** | **Input interval** | **Objective Function** | **Continuous non-continuous** | **Differentiable**  **Vs. differentiable** | **Convex vs. non-convex** |
| 1 | Sphere function |  |  | **Continuous** | **Differentiable** | Convex |
| 2 | Rosenbrock’s function |  |  | **Continuous** | **Differentiable** | Non convex |
| 3 | Ackley’s function |  |  | **Continuous** | **Differentiable** | Non convex |
| 4 | Griewanks’s function |  |  | **Continuous** | **Differentiable** | Non convex |
| 5 | Weierstrass function |  |  | **Continuous** | **Non-Differentiable** | Non convex |
| 6 | Rastrigin’s function |  |  | **Continuous** | **Differentiable** | Non convex |
| 7 | Noncontinuous Rastrigin’s function |  |  | Non **Continuous** | Non **Differentiable** | Non convex |

### Wireless Sensor Network

The wireless sensor network deployment is a multi-objective optimization that involves locating sensors in a pre-defined environment with minimizing two objectives: the first one is the cost of the sensors and the second one is the non-covered area. The optimizing involves fining set of non-dominated solution, i.e. Pareto front, with respect to the above mentioned objectives. The length of the solution is variable according to the number of regarded sensors in the space. The model is given in table –

|  |  |  |
| --- | --- | --- |
| Solution |  |  |
| denotes the number of sensors |  |  |
|  |  |  |
|  |  |  |

# Experimental Work Design and Evaluation

This section presents the experimental work design and the evaluation results with the analysis. The section is decomposed from two sub-sections: the first one is the experimental work design provided in sub-section 5.1. The second one is the evaluation results and the analysis which is provided in sub-section 5.2.

## Experimental Work Design

We run the algorithm SC-MOPSO and the benchmark that we selected for comparison which is MOPSO. We emphasize on the fact that the problems that we are testing have the nature of multi-objective and variable length. Considering that all existing algorithms in the literature are either fixed length with supporting multi-objective or single objective with supporting variable length, however, non-of them supports the two aspects at the same time. Hence, in order to compare, we select MOPSO which is originally fixed length and we run it on the maximum length with encoding the solutions that have lower dimension with NA for its-non-enabled dimension. Consequently, it can be used for solving variable length problems with multi-objective nature. This modification is regarded as small ad hoc solution to convert MOPSO to be supportive to variable length multi-objective problems. The parameters that are used for both SC-MOPSO and MOPSO are depicted in table 8.

Table 8 Setting parameters of SC-MOPSO

|  |  |  |
| --- | --- | --- |
| **Parameters** | SC-MOPSO | MOPSO |
| number of iterations | 100 | 100 |
| Lower Bound\_pos | -600 | -600 |
| Higher Bound\_pos | 600 | 600 |
| Pop Size | 20 | 20 |
| nGrid | 7 | 7 |
| Inertia | 0.4 | 0.4 |
| movingThreshold | 5 | - |
| classMinThreshold | 1 | - |

Afterwards, we combine of the mathematical functions that are provided in table 9. Ten variable length multi-objective optimization problems with either two or three objectives. We present them in table 9.

Table 9 variable length mathematical problems for evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Problem | Objectives functions | | |
|  |  |  |
| 1 | Griewanks | Rastrigin |  |
| 2 | Griewanks | Sphere |  |
| 3 | Rosenbrock’s | Ackley’s |  |
| 4 | Rosenbrock’s | Ackley’s | Noncontinuous Rastrigin’s |
| 5 | Rosenbrock’s | Griewanks’s |  |
| 6 | Rosenbrock’s | Griewanks’s | Rastrigin’s |
| 7 | Rosenbrock’s | Sphere |  |
| 8 | Sphere | Ackley’s | Noncontinuous Rastrigin’s |
| 9 | Sphere | Ackley’s | Rastrigin’s |
| 10 | Weierstrass | Griewanks’s |  |

## Evaluation Results and Analysis

For evaluating the performance of our developed SC-MOPSO and comparing its performance with the classical MOPSO, we generated the set coverage metrics. This metric is the most important in deciding the domination aspect. As we see in figure--, the set coverage of SC-MOPSO with respect to MOPSO and vice versa were plotted. They are denoted as and respectively. The first indicates to the domination percentage of SC-MPSO over MOPSO and the second one indicates to domination percentage MOPSO over SC-MOPSO. The results in figure – reveals the dominance over SC-MOPSO over MOPSO with high percentage comparing with the dominance of MO-PSO over SC-MOPSO. Furthermore, we observe from the figure – that for problem 1 and problem 3, the domination of MOPSO over SC- MOPSO was zero comparing with 0.7 domination of SC-MOPSO over MOPSO. In addition, MOPSO was able to dominated SC-MOPSO with only 0.1 for the problems 5, 8, 9 and 10. Looking at the remaining problems, we see that SC-MOPSO has accomplished more domination over MOPSO. The only problem that MOPSO has slightly dominated SC-MOPSO is problem 6. The superior performance of SC-MOSPO is interpreted by the fact that SC-MPSO performs the searching in sub-sets of particles with the same dimension for each particle, and moves solutions toward an exemplar from the same set or class. However, SC-MOPSO is dynamic in the aspect of moving particles from one class to another when a domination happens. On the other hand, MOPSO deals with issue of variable length by allocating solution size with the maximum expected dimension and conduct classical searching. However, the approach does not gives dimension aware mobility which makes it able to stuck in local minima.

|  |  |  |
| --- | --- | --- |
| problem 1 | problem 2 | problem 3 |
| problem 4 | problem 5 | problem 6 |
| problem 7 | problem 8 | problem 9 |
| problem 10 |

Figure –set coverage for the problem sets from problem from 1 until 10

The other aspect of performance is the hyper-volume that indicates to the spread of solution in the objective space. Hence, it is a measure of diversity. Such measure is taken into consideration when the two approaches show the same domination. In figure--, we see that SC-MOPSO has accomplished higher hyper-volume for 6 problems comparing with only 4 problems with more hyper-volume than MO-PSO. The superiority of performance is resulted from the class/dimension aware searching and dynamical moving of solutions within classes in SC-MOPSO. On the other hand, MO-PSO show less hyper-volume in most problems even with the dominance of SC-MOPSO. This is interpreted by the lacking of class/dimension aware searching.

|  |  |  |
| --- | --- | --- |
| problem 1 | problem 2 | problem 3 |
| problem 4 | problem 5 | problem 6 |
| problem 7 | problem 8 | problem 9 |
| problem 10 |

Figure –hypervolume for the problems from problem from 1 until 10

The number of non-dominated solution is also found as a third performance measure. We see from figure --, that SC-MOPSO has accomplished higher NDS for problems 6 problems comparing with only 4 where MO-PSO has outperformed from the perspective of this metric. We also interpret it with the class aware searching comparing with the lacking of it in MOPSO.

|  |  |  |
| --- | --- | --- |
| problem 1 | problem 2 | problem 3 |
| problem 4 | problem 5 | problem 6 |
| problem 7 | problem 8 | problem 9 |
| problem 10 |

Figure –NDS for the problems from problem from 1 until 10

The other metric is the relative generational distance RGD which indicates to the difference between level of certain objective and the best accomplished level of the two approaches. The lower RGD is equivalent to higher performance and more optimality. As we observe in figure--, SC-MOPSO has accomplished more optimality based on lower RGD comparing with MOPSO. This is observed for all problems except 5 and 8. However, this does not mean a superiority of MOPSO in these two problems because going back to the set coverage metric shows that SC is more dominant for these two problems.

|  |  |  |
| --- | --- | --- |
| problem 1 | problem 2 | problem 3 |
| problem 4 | problem 5 | problem 6 |
| problem 7 | problem 8 | problem 9 |
| problem 10 |

Figure –RGD for the problems from problem from 1 until 10

The last aspect of performance is the distribution of solutions among classes. This metric is the histogram that is presented in figures – for SC-MOPSO and MOPSO respectively. We see that both approaches were able to generate solutions in different classes. This gives the decision more flexibility in terms of the solutions distribution. Because the decision maker will be able to select solutions from the same rank with different dimension. In addition, we observe that some classes have not gained any solution in the Pareto front. These is interpreted by the mathematical nature of the problem itself.

|  |  |  |
| --- | --- | --- |
| problem 1 | problem 2 | problem 3 |
| problem 4 | problem 5 | problem 6 |
| problem 7 | problem 8 | problem 9 |
| problem 10 |

Figure –histogram for SC-MOPSO for the problems from problem from 1 until 10

|  |  |  |
| --- | --- | --- |
| problem 1 | problem 2 | problem 3 |
| problem 4 | problem 5 | problem 6 |
| problem 7 | problem 8 | problem 9 |
| problem 10 |

Figure –histogram for MOPSO for the problems from problem from 1 until 10

# Conclusion and Future Works

This article has proposed a novel meta-heuristic optimization algorithm based on particle swarm optimization named social class multi-objective particle swarm optimization SC-MOPSO. It is developed to solve special type of difficult optimization problems that have multi-objective aspect and variable length dimension nature. Such problems are faced in different practical applications and researchers solve it by using standard MHSO algorithms after adding some ad hoc changes. Unlike classical approaches, this novel approach extends the concept of particle swarm optimization by incorporating solutions interaction within one class that represents set of solutions with the same dimension and solutions interaction among classes by moving solutions from the lower performing class to the higher one. The article has presented novel concepts in this domain and added new evaluation metrics for such types of difficult problems. Extensive evaluation based on 10 created mathematical problems shows the superiority of SC-MOPSO from all aspects of performance like from domination, spread, errors and diversity. Future work is to apply the developed algorithm in real world applications such as wireless sensor network coverage and mobile sink. Other future work is to use it for artificial intelligence problems such as classifier optimization and clustering.

[1] M. H. Salmani and K. Eshghi, "A metaheuristic algorithm based on chemotherapy science: CSA," *Journal of Optimization,* vol. 2017, 2017.

[2] H.-C. Huang and S.-K. Lin, "A hybrid metaheuristic embedded system for intelligent vehicles using hypermutated firefly algorithm optimized radial basis function neural network," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 2, pp. 1062-1069, 2018.

[3] B. Zeng and Y. Dong, "An improved harmony search based energy-efficient routing algorithm for wireless sensor networks," *Applied Soft Computing,* vol. 41, pp. 135-147, 2016.

[4] V. Mandhare, V. Thool, and R. Manthalkar, "QoS Routing enhancement using metaheuristic approach in mobile ad-hoc network," *Computer Networks,* vol. 110, pp. 180-191, 2016.

[5] V. Babaveisi, M. M. Paydar, and A. S. Safaei, "Optimizing a multi-product closed-loop supply chain using NSGA-II, MOSA, and MOPSO meta-heuristic algorithms," *Journal of Industrial Engineering International,* vol. 14, no. 2, pp. 305-326, 2018.

[6] R. R. Swain, P. M. Khilar, and T. Dash, "Multifault diagnosis in WSN using a hybrid metaheuristic trained neural network," *Digital Communications and Networks,* 2018.

[7] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing,* vol. 43, pp. 117-130, 2016.

[8] H. Li, K. Deb, and Q. Zhang, "Variable-Length Pareto Optimization via Decomposition-Based Evolutionary Multiobjective Algorithm," *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 6, pp. 987-999, 2019.

[9] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, "A survey of evolutionary algorithms using metameric representations," *Genetic Programming and Evolvable Machines,* vol. 20, no. 4, pp. 441-478, 2019.

[10] H. Li and K. Deb, "Challenges for evolutionary multiobjective optimization algorithms in solving variable-length problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 2217-2224: IEEE.

[11] Y. Wang, S. Wu, Z. Chen, X. Gao, and G. Chen, "Coverage problem with uncertain properties in wireless sensor networks: A survey," *Computer Networks,* vol. 123, pp. 200-232, 2017.

[12] P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "ACO-based mobile sink path determination for wireless sensor networks under non-uniform data constraints," *Applied Soft Computing,* vol. 69, pp. 528-540, 2018.

[13] S. Mukhopadhyay and T. D. Chaudhuri, "Different Length Genetic Algorithm-Based Clustering of Indian Stocks for Portfolio Optimization," in *Advances in Intelligent Computing*: Springer, 2019, pp. 45-59.

[14] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *Journal of Network and Computer Applications,* vol. 108, pp. 87-111, 2018.

[15] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 3, pp. 473-487, 2018.

[16] C. Byers, B. H. Cheng, and K. Deb, "Unwanted feature interactions between the problem and search operators in evolutionary multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2015, pp. 19-33: Springer.

[17] M. L. Ryerkerk, R. C. Averill, K. Deb, and E. D. Goodman, "Solving metameric variable-length optimization problems using genetic algorithms," *Genetic Programming and Evolvable Machines,* vol. 18, no. 2, pp. 247-277, 2017.

[18] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, "A novel selection mechanism for evolutionary algorithms with metameric variable-length representations," *Soft Computing,* pp. 1-14.

[19] A. Mukhopadhyay and M. Mandal, "Identifying non-redundant gene markers from microarray data: a multiobjective variable length PSO-based approach," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 11, no. 6, pp. 1170-1183, 2014.

[20] B. Xue, X. Ma, H. Wang, J. Gu, and Y. Li, "Improved variable-length particle swarm optimization for structure-adjustable extreme learning machine," *Control and Intelligent Systems,* vol. 42, no. 4, pp. 1-9, 2014.

[21] J. Cheng, J. Chen, Y.-n. Guo, S. Cheng, L. Yang, and P. Zhang, "Adaptive CCR-ELM with variable-length brain storm optimization algorithm for class-imbalance learning," *Natural computing,* pp. 1-12, 2019.

[22] Y. Wang and X. Zhu, "A Novel Network Planning Algorithm of Three-Dimensional Dense Networks Based on Adaptive Variable-Length Particle Swarm Optimization," *IEEE Access,* vol. 7, pp. 45940-45950, 2019.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942-1948 vol.4.

[24] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE transactions on Evolutionary Computation,* vol. 3, no. 4, pp. 257-271, 1999.

[25] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Citeseer1998.

List of actions

1. Modify pseudocode of SC-MOPSO
2. Add pseudocode of MS-SC-MOPSO
3. Add pseudocode of solution interaction of MS-SC-MOPSO
4. Add pseudocode of progressive evaluation of MS-SC-MOPSO
5. Add results of mathematical function
6. Add results of MS-SC-MOPSO and SC-MOPSO for WSN deployment
7. Change table of mathematical problems