S-metric Based Multi-objective Fireworks Algorithm

Lang Liu, Shaoqiu Zheng and Ying Tan
† Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University
Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing,100871, P.R. China
Email: {liulang, zhengshaoqiu, ytan}@pku.edu.cn

***Abstract*—Fireworks Algorithm(FWA) is a recently developed swarm intelligence algorithm for single objective optimization problems which gains very promising performances in many areas. In this paper, we extend the original FWA to solve multi-objective optimization problems with the help of S-metric. The S-metric is a frequently used quality measure for solution sets comparison in evolutionary multi-objective optimization algorithms (EMOAs). Besides, S-metric can also be used to evaluate the contribution of a single solution among the solution set. Traditional multi-objective optimization algorithms usually perform a $(\mu + 1)$ strategy and update the external archive one by one, while the proposed S-metric based multi-objective fireworks algorithm(S-MOFWA) performs a $(\mu + \mu)$ strategy, thus converging faster to a set of pareto solutions by three steps: 1)Exploring the solution space by mimicking the explosion of fireworks; 2)Performing a simple selection strategy for choosing the next generation of fireworks according to their S-metric; 3)Utilizing an external archive to maintain the best solution set ever found, with a new archive definition and a novel updating strategy, which can update the archive with μ solutions in a single process. The experimental results on benchmark functions suggest that the proposed S-MOFWA outperforms three other well-known algorithms, i.e. NSGA-II, SPEA2 and PESA2 in terms of the convergence measure and covered space measure.**

I. INTRODUCTION

FIREWORKS algorithm (FWA) [1] is one recently proposed swarm intelligence algorithm. The initial idea of FWA is from the observation of the fireworks explosion in the night sky. The fireworks are set off to the sky and each of them explodes and generates a number of explosion sparks, illustrating the nearby space. In fact, this nature phenomenon is similar to the optimization process. If one local region of the optimization problem is thought promising, the fireworks could be set off to this region and the explosion sparks will be generated to perform the local search.

Since the proposal of FWA in 2010, the interesting idea has attracted many researchers to devote themselves working on this topic. So far, the developments of fireworks algorithm can be grouped into two parts, the algorithms developments and applications. For algorithms developments, much work has been done, for example, the algorithm improvements based on operators analysis [2]–[5], based on hybrid with other algorithms [6]–[11]. Moreover, the parallel FWA on Graphic Processor Unit has also been proposed [12]. For the applications, FWA has been used for digital filters design [6], non-negative matrix factorisation calculation [13], pattern recognition [14], spam detection [15], network reconfiguration [16], [17], truss mass minimisation [18], clustering [19] and so on.

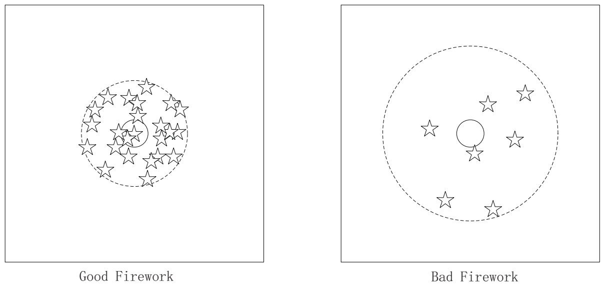


Fig. 1. Good/Bad fireworks comparison in explosion amplitude and explosion sparks number

The basic framework of FWA works as follows: At first, a number of fireworks are initialized within the feasible search range, each firework will be evaluated for their fitness. In general, the fitness is one character to denote whether the local region is promising or not. The fireworks with better fitness will have larger numbers of explosion sparks and smaller explosion amplitudes, while the fireworks with worse fitness will have smaller numbers of sparks and bigger explosion amplitudes. Each firework performs an explosive mutation process for the local search. Fig. 1 gives a brief comparison of good and bad fireworks in terms of the explosion sparks numbers and explosion amplitudes.

After the explosion process, a certain number of mutation sparks are also generated to increase the diversity of the swarm. Then the fireworks for next generation will be selected from the candidates set which includes the fireworks, explosion sparks and mutation sparks, according to their fitness and diversity. This procedure repeats iteratively until the max iteration time is reached or a solution good enough is found.

Multi-objective problem has two or more conflicting objectives to optimize simultaneously. Lack of prior knowledge about the objectives, we usually investigate into a vast number of solutions and reserve the non-dominated ones, i.e. a Pareto solution set, as the approximation of the true Pareto-optimal set. This idea contributes directly to the rapid development of evolutionary multi-objective optimization algorithm(EMOA) [20].

In MSOPS(Multiple Single Objective Pareto Sampling) [21] [22], the relationship between conflicting objectives was quantified as weights added to each objective. Despite unknown, when enough trials of ways to combine and adjust the weights were explored, enough non-dominated solutions would be found to make a good spread of the Pareto front.

During the evolution of EMOA, there are two crucial points to consider: convergence and diversity. Most EMOAs consider them apart. For example, in NSGA-II [23], the non-dominated sorting is performed first to classify all the solutions into hierarchical dominated levels as their measure of convergence, then a crowding distance will be calculated as their measure of diversity. In SPEA2 [24], the fitness of each solution is composed by two parts: dominated strength which represents the convergence, and density metric which represents the diversity.

Different from the above methods, indicator based methods choose a single metric to represent both convergence and diversity. IBEA [25] defines a binary quality indicator and each solution needs to be compared with every other solution in order to get a single fitness. The comparison times required is as much as that for non-dominated sorting. S-metric [26] is another indicator that represents both convergence and diversity, by calculating the hyper-volume of dominated space. A great amount of attention was paid to this metric since its proposal. Nicola Beume and Carlos did a detailed analysis on the complexity of computing the hyper-volume indicator [27]. Knowles and Corne [28] gives an example for fast calculating the S-metric.

In this paper, a multi-objective fireworks algorithm base on S-metric is proposed. The explosion amplitudes of fireworks and the numbers of sparks are calculated according to each firework's S-metric, the next generation of fireworks are selected by their S-metric and an external archive with fixed size is used to maintain the best solution set found ever, by a novel updating strategy.

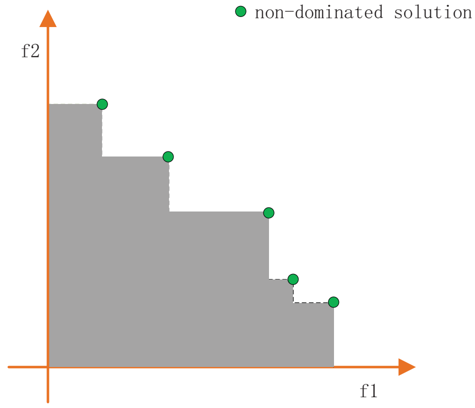
The rest of this article is organised as follows: The S-metric is introduced in section 2. The proposed S-metric based multi-objective fireworks algorithm is demonstrated in section 3. Section 4 deals with the performance on several test problems compared to three other algorithms. Section 5 draws a brief summary and introduces the future work.

II. S-METRIC

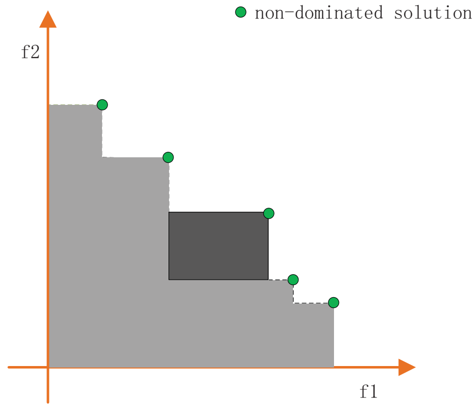
The S-metric or hyper-volume indicator was first proposed by Zitzler and Thiele [26]. It can be viewed as the *size of the space covered* or *size of dominated space*. Let Λ denote the Lebesgue measure, then the S-metric for a solution set $M = \{m_1, m_2, ..., m_i, ...m_n\}$ is defined as [29]:

$$S(M) := \Lambda\left(\bigcup_{m \in M} \{x | m \prec x \prec x_{ref}\}\right) \quad (1)$$

where, $\prec$ denotes the dominance relationship, i.e. $a \prec b$ means b is dominated by a. The x_{ref} is a reference point dominated by all valid solutions in the solution set.



(a) S-metric for the set



(b) S-metric for a solution

Fig. 2. S-metric

Each solution in set M contributes a part to this total S-metric, so the S-metric for each solution is defined as:

$$S(m_i) = \Delta S(M, m_i) := S(M) - S(M \backslash \{m_i\}), \quad (2)$$

The S-metric for a solution m_i can be seen as the size of space dominated by m_i but not dominated by any other solutions in the set. A large value means m_i is important in set M, while zero indicates that m_i must be dominated by some other solutions in set M.

Fig. 2 shows the meaning of S-metric and the difference between S-metric for set and S-metric for a single solution. It can be seen that S-metric for a solution has no relation with the choice of a reference point. And a solution with larger S-metric value lies further away from its nearest neighbors, which is coincident with SPEA2's density metric in the diversity meaning. Besides, a non-zero value guarantees that the solution is non-dominated, coincident with convergence meaning, because according to the definition in Eq.(2), the S-metric for a solution dominated by others is zero.

III. THE PROPOSED S-MOFWA

As for multi-objective problems, there are more than one objective function, f_1, f_2, etc. But for an EMOA, the fitness function is unique.

In the proposed S-MOFWA, the fitness for an individual is its S-metric in the current solution set. Noted that the S-metric of one solution is merely relevant to its neighbors and could change when the members of the current solution set change.

A brief description of S-MOFWA is as follows: At first, a number of fireworks will be initialized in the search space. In each iteration, we calculate the numbers of explosion sparks and amplitudes of explosion for each firework. Then the explosion and Gaussian mutation will be performed to generate a number of sparks. Finally fireworks for next iteration will be selected among the candidates. An external archive is used to maintain the best solution set. The details of each step can be found in the following paragraphs.

A. *Initialization*

The algorithm randomly selects N points in the search space as the first generation of fireworks. As it's the first generation, there is no need to calculate or set the objective function values. Here, we simply set their S-metric the same values, so they have the same sparks numbers and explosion amplitudes, because we don't hurry to tag them as good or bad in the initial stage.

B. *Calculation of the explosion sparks number and explosion amplitude*

As is mentioned above, to make a contrast among the fireworks, the firework with better fitness will have larger number of explosion sparks and smaller explosion amplitude, while the firework with worse fitness will have smaller number of sparks and bigger explosion amplitude. To ensure this principle, the number of sparks z_i and explosion amplitude A_i for each firework x_i are calculated by Eq.(3) and Eq.(4), which are determined by the S-metric solely.

$$z_i = M_e * \frac{S(x_i) + \varepsilon}{\sum_{i=1}^{N} (S(x_i)) + \varepsilon}, \tag{3}$$

$$A_i = \hat{A} * \frac{S_{\max} - S(x_i) + \varepsilon}{\sum_{i=1}^{N} (S_{\max} - S(x_i)) + \varepsilon}, \tag{4}$$

where $S_{\max} = max(S(x_i)), i = 1, 2, ..., N$. M_e and $\hat{A}$ are two constant parameters to control the sparks number and amplitude. To bound the values z_i to a proper range, two other constants $a, b \epsilon [0, 1]$ was also needed.

$$z_i = \begin{cases} round(aM_e) & \text{if } z_i < aM_e, \\ round(bM_e) & \text{if } z_i > bM_e, \\ round(z_i) & otherwise. \end{cases} \tag{5}$$

Noted that the best solution, namely the solution with a max S-metric gets a very small explosion amplitude which is close to zero according to Eq.(4). This may be acceptable in the final stage of the evolution, but that's unreasonable for the early stage. The non-linearly decreasing amplitude threshold $A_{\min}$ introduced in [30] is also employed here:

$$A_{\min}(t) = A_{init} - \frac{A_{init} - A_{final}}{evals_{max}} \sqrt{(2 * evals_{max} - t)t}, \tag{6}$$

where A_{init} and A_{final} are the initial and final minimum explosion amplitudes, $evals_{max}$ is the maximum evaluation times and t is the current evaluation times. And,

$$A_i = max(A_i, A_{\min}), \tag{7}$$

C. *Explosion*

For each firework x_i, z_i sparks will be generated according to Alg.1. This explosion procedure can be seen as the exploration among the search space.

Algorithm 1 Explosion

for $j = 1 \to z_i$ **do**
  initialize the location of the explosion spark: $\hat{x}_j = x_i$
  for each dimension k of x_i **do**
    if $rand(0, 1) < 0.3$ **then**
      $\hat{x}_j^k = x_i^k + A_i * rand(-1, 1)$
      if $\hat{x}_j^k$ out of bounds **then**
        $\hat{x}_j^k$ =U($x_{\min}^k$,$x_{\max}^k$)
      end if
    end if
  end for
end for

D. *Gaussian mutation*

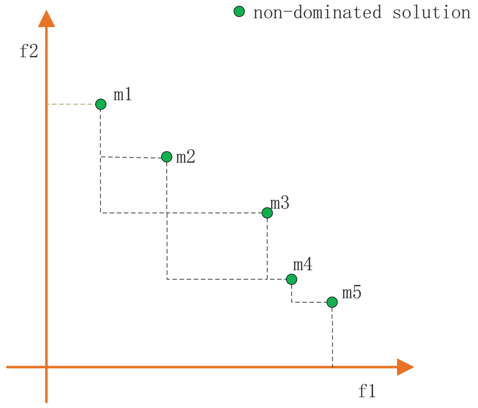
To increase the diversity of generated explosion sparks swarm, the algorithm will also generate a number of special Gaussian sparks through a process called Gaussian mutation. Alg.2 describes process of the calculation of Gaussian sparks.

Algorithm 2 Gaussion mutation

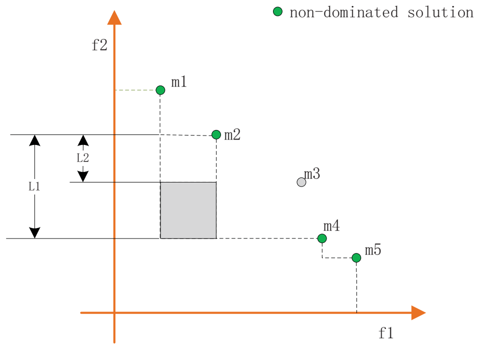
initialize the location of the Gaussian spark: $\tilde{x}_i = x_i$
for each dimension k of x_i **do**
  if $rand(0, 1) < 0.5$ **then**
    $\tilde{x}_i^k = x_i^k * normrnd(1, 1)$
    if $\tilde{x}_i^k$ out of bounds **then**
      $\tilde{x}_i^k$ =U($x_{\min}^k$,$x_{\max}^k$)
    end if
  end if
end for

E. *Fireworks selection and archive updating*

1) Calculation of S-metric: In conventional FWA, the fireworks for next generation are selected according to their fitness and diversity measure [1]. As for S-MOFWA, the S-metric is calculated according to their values of objective functions, and is also used as selection criteria. In case of two objectives, we sort the sparks in descending order according to the values of the first objective function f_1. After wiping



(a) calculation of S-metric



(b) the increase of neighbors' S-metric after wiping away a solution

Fig. 3. S-metric

away the points that are dominated, we will get a sequence sorted in ascending order according to the second objective function f_2. (Those points which lose in both two objective functions are dominated.) As showed in Fig.3(a), the S-metric for solution m_i is calculated as Eq.(8):

$$S(m_i) = (f_1(m_i) - f_1(m_{i-1})) * (f_2(m_i) - f_2(m_{i+1})), \quad (8)$$

Noted that the calculation of S-metric does not depend on the order of the objectives to be considered. All the objectives are of coordinate importance. In the process of sorting the sparks by one objective, we can easily find out and erase those who are dominated by others. Then the S-metric of a non-dominated spark is determined by its nearest neighbors in each objective. The nearest neighbors restrict the space which is only dominated by this spark, but not dominated by any other sparks. The volume of this space is the so-called hypervolume, namely the S-metric.

For two objectives problem, in each iteration, after the sorting process we can calculate all the sparks' S-metric one by one. So the complexity is $O(n\log n)$. For three or more objectives problem, the complexity is $O(n^{d/2}\log n)$ due to Overmars and Yap [31]. Here, n is the number of sparks and d is the number of objectives.

Here, we need to clarify that the calculation of S-metric can be of high efficiency and is also suitable for more objectives problems. Boris Naujoks and Nicola Beume presented an algorithm for calculating S-metric in three-objective space [32]. The LebMeasure algorithm described by Fleischer [33] and the HSO algorithm(Hypervolume by Slicing Objectives) proposed by Knowles [34] and Zitzler [35] compute the whole S-metric for all dimensional objectives. Further more, many researches have been done in order to improve the efficiency of calculating S-metric. L.While and P.Hingston presented a fast algorithm for calculating S-metric [36]. Then L.Bradstreet proposed an incremental version to calculate S-metric. On the foundation of those improvements, many new EMOAs based on S-metric have been invented like SMS-EMOA [37] and MOPSOhv [38].

2) Selection of the fireworks for the next generation: In the original FWA, we choose the best one and the rest $N-1$ fireworks according to their crowding metric as the next generation of fireworks [1]. However, as is mentioned previously, a large S-metric guarantees that the solution is far away from its nearest neighbors and it's non-dominated. In another word, a solution with large S-metric points out a good region which has been explored too little. Our job is to strengthen the exploration in the suggested region. So there is no need to introduce another unnecessary diversity measure and we simply choose the best N solutions as the fireworks for next generation, according to their S-metric, namely the fitness.

3) Updating of the external archive: In conventional archive strategy, external archive is just a place for storing non-dominated solutions. When a new individual was generated, it would be examined whether it can be put into the archive or replace someone. This was done one by one, which means the external archive needs to be updated for a thousand times if one thousand new individuals were generated.

In this paper, the external archive always keeps a fixed number of solutions (assume the fixed number is K), which differs from the conventional grow-up strategy. These K solutions are chosen from the candidates pool which includes the sparks generated by fireworks explosion and Gaussian mutation, and the old archive. The selected K solution needs to ensure that they get a maximum S-metric in all the K-sets.

We need to point out that the solutions in this archive are not necessarily non-dominated. Noted that in the early stage of evolution, not enough non-dominated solutions should have been found. But the external archive always keeps K solutions. Inevitably there are many dominated solutions existing in the archive, but this does not matter. Along with the iteration, the S-metric of the solution set in the archive will increase step by step. And Fleischer proved that a finite solution set with the theoretic maximum of S-metric comes necessarily from the true Pareto front [33].

Fig.4 demonstrates the evolvement of the external archive on function KUR of S-MOFWA. The size of archive is fixed as $K = 100$. In the first 20 iterations, most of the solutions were dominated by those points close to the true Pareto front.

After hundreds of iterations, nearly all of them became non-dominated, and the updates concentrate in the areas very close to the true Pareto front.

In each iteration, we will choose K solutions to build up the new archive. This sounds like a combinational problem. However, one solution's S-metric is solely determined by its nearest neighbors, which indicates that wiping off a solution only influences the S-metric of a few solutions. Here, we keep a proper ratio (about 1:1) between the size of external archive and the total number of sparks, wipe off the worst one solution in the candidates set and fix its neighbors' S-metric. This procedure repeats iteratively, eventually remaining K solutions as the new archive.

As is showed in Fig.3(b), after wiping off the solution m_3, its nearest neighbors m_2 and m_4 have a proportional increase in their S-metric, while the others' remain unchanged. Take m_2 for instance, the ratio of new $S(m_2)$ to the old S-metric is L_2/L_1. So a simple update will be enough:

$$S(m_2) = S(m_2) * \frac{f_2(m_2) - f_2(m_4)}{f_2(m_2) - f_2(m_3)}, \tag{9}$$

Moreover, the corresponding operator also needs to be performed on m_4, another nearest neighbor:

$$S(m_4) = S(m_4) * \frac{f_1(m_4) - f_1(m_2)}{f_1(m_4) - f_1(m_3)}, \tag{10}$$

The procedure of updating the external archive is described in Alg.3

Algorithm 3 Updating strategy for the external archive

note Q as the sparks generated by fireworks explosion and Gaussian mutation
note A as the external archive
note C as the candidates pool, set $C = Q \cup A$
calculate S-metric for each candidate in C /* *a minimum heap may be useful here.* */
while $|C| > K$ /* *K is the size of the external archive, a fixed number* */ **do**
    $r \leftarrow \arg\min_{m \epsilon C}(S(m))$ /* *detect element of C with the lowest S-metric* */
    $C \leftarrow C \backslash \{r\}$ /* *eliminate detected element* */
    fix the S-metric values of nearest neighbors of r according to Eq.(9) and Eq.(10)
end while
$A \leftarrow C$

F. Framework of S-MOFWA

The framework of S-MOFWA is described in Alg.4

Algorithm 4 Framework of S-MOFWA

randomly initialize N fireworks in the solution space
initialize external archive A as the set of the initial fireworks
iteration $p \leftarrow 0$
while terminal conditions are not met **do**
    $p \leftarrow p + 1$
    update the current evaluation times t
    update the non-linearly decreasing amplitude threshold $A_{\min}$ according to Eq.(6)
    for $j = 1 \rightarrow N$ **do**
        calculate objective functions' values of each firework x_i
    end for
    calculate each fireworks' S-metric according to Eq.(8)
    calculate each fireworks' sparks number and explosion amplitude according to Eq.(3) and Eq.(4) respectively, then bound them by corresponding Eq.(5) and Eq.(7)
    perform the firework explosion as Alg.1 for each firework
    perform the Gaussian mutation as Alg.2 for each firework
    calculate each candidate's S-metric /* *candidates refers to the sparks obtained by explosion and Gaussion mutation, and the solutions in the current archive* */
    select N best candidates as the next generation of fireworks, according to their S-metric
    update the external archive A
end while
output the external archive A

IV. Experiments

To validate the performance of the proposed S-MOFWA, experiments on benchmark suite which contains 6 test functions were designed. Moreover, the performance comparison with other three well-known algorithms (NSGA-II, SPEA2, PESA2) are also conducted.

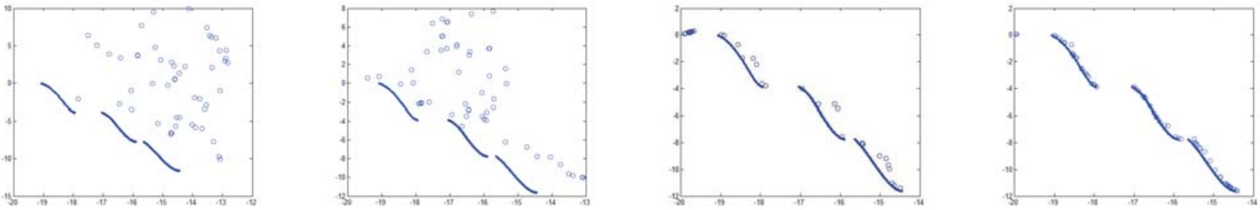
A. Experimental Setup

The test problems used to compare different algorithms are chosen from a set of significant studies in the area of multi-objective optimization. Schaffer's problem (SCH) and Kursawe's problem(KUR) are presented by Veldhuizen [39]. ZDT1, ZDT2, ZDT3, ZDT6 are selected from the six test functions suggested by Zitzler [40]. All above problems have two objective functions and none of them have constrains. The properties of these functions are listed in Tab. I.

For the parameters setting, the number of fireworks $N = 10$, size of external archive $K = 100$, total number of sparks $M_e = 100$, constants $a = 0.05$, $b = 0.4$, the mutation rate for Gaussian mutation $r_1 = 0.5$ and for explosion $r_2 = 0.3$, evaluation times $evals_{max} = 200000$, the same for the comparison algorithms. Each function runs 20 repetition times. The rest of parameters for the fireworks are identical to [30]. For the parameters setting of NSGA-II, SPEA2 and PESA2, they are set according to the source code of [41] [1].

The experimental platform: Win8; Intel Core Duo $E4500$ CPU; 2.2GHz; 2GB RAM. The proposed S-MOFWA was run on MATLAB2012b and the comparison algorithms were run on Eclipse-Java.

[1]jmetal : http://sourceforge.net/projects/jmetal/



(a) iteration =10 (b) iteration =20 (c) iteration =250 (d) iteration =300

Fig. 4. MOFWA evolution process on KUR.

TABLE I
BENCHMARK FUNCTIONS (DIM. = DIMENSION)

	Dim.	Range	Optimal locations	Property
SCH	1	$[-1000, 1000]$	$x \in [0, 2]$	convex
KUR	3	$[-5, 5]$	refer [39]	non-convex
ZDT1	30	$[0, 1]$	$x_1 \in [0, 1]$,$x_j = 0, j \neq 1$	convex
ZDT2	30	$[0, 1]$	$x_1 \in [0, 1]$,$x_j = 0, j \neq 1$	non-convex
ZDT3	30	$[0, 1]$	$x_1 \in [0, 1]$,$x_j = 0, j \neq 1$	convex, disconnected
ZDT6	10	$[0, 1]$	$x_1 \in [0, 1]$,$x_j = 0, j \neq 1$	non-convex, non-uniformly spaced

B. Evaluation Criterion

In this paper, we take the convergence measure and covered space measure to characterize the performance of one multi-objective optimization algorithm.

The **convergence measure** is the average distance to the closest point of the true Pareto front as used in [42]. The smaller the distance is, the closer the solution set lies to the true Pareto front. Usually we uniformly select hundreds of point from the true Pareto front as a approximation. So this measure is calculated as the average distance to the closest point in the selected approximate set.

The **covered space measure** is a relative ratio of the covered hypervolume, for estimation of the diversity of the spread. The denominator could be the size of space covered by the true Pareto front, or the size of a fixed feasible function space. Here, we take the latter for simple calculation. The maximum values are limited by the different shapes of test functions, so the comparative values are only of meaning within the same test function. Of course, a larger value indicates the spread is better-distributed.

C. Experimental Results

The results for average convergence measure and covered space measure of S-MOFWA, NSGA-II, SPEA2 and PESA2 are listed on Tab. II and Tab. III, repectively. And the T-test results are listed in Tab.IV and Tab.V. In addition, we also plot the final archive in the objective space on the benchmark functions of S-MOFWA.

TABLE IV
p-VALUES FOR CONVERGENCE MEASURE (THE VALUES IN BOLD INDICATE THAT MOFWA IS SIGNIFICANT BETTER COMPARED WITH THE OTHER ALGORITHM)

vs NSGA-II	**8.857E-05**	**8.900E-05**	**1.030E-04**	**5.170E-04**	**1.030E-04**	2.959E-01
vs SPEA2	**8.900E-05**	**2.509E-02**	**8.900E-05**	**1.400E-04**	**1.162E-03**	6.542E-01
vs PESA2	**8.900E-05**	**1.200E-04**	**2.930E-04**	**3.185E-03**	**8.900E-05**	**2.190E-04**

TABLE V
p-VALUES FOR COVERED SPACE MEASURE (THE VALUES IN BOLD INDICATE THAT MOFWA IS SIGNIFICANT BETTER COMPARED WITH THE OTHER ALGORITHM)

vs NSGA-II	7.314E-02	**4.550E-03**	**8.900E-05**	7.932E-02	**1.630E-04**	**8.900E-05**
vs SPEA2	1.507E-03	5.016E-01	**1.200E-04**	**7.189E-03**	**1.630E-04**	**8.900E-05**
vs PESA2	**3.380E-04**	**8.900E-05**	**1.325E-03**	**2.277E-02**	**2.190E-04**	**8.900E-05**

D. Discussion

For convergence measure, the proposed S-MOFWA performs best on all the test functions, specifically discussed by the type of functions as follows:

1) convex functions: On convex functions ZDT1 and SCH, the proposed S-MOFWA significantly outperforms the other three with smallest average convergence measure and standard deviation, both ability and stability are proven.

2) non-convex functions: On non-convex function ZDT2, the proposed S-MOFWA and SPEA2 both get the best convergence measure, and S-MOFWA shows a little superiority in the standard deviation. On function KUR, S-MOFWA performs best and SPEA2 ranks the second.

3) disconnected functions: On disconnected function ZDT3, S-MOFWA beats other three in average convergence measure but the standard deviation ranks the last.

4) non-uniformly spaced functions: On non-uniformly spaced function ZDT6, S-MOFWA, NSGA-II and SPEA2 get the best average convergence measure while PESA2 does not obtain a value of the same order of magnitude.

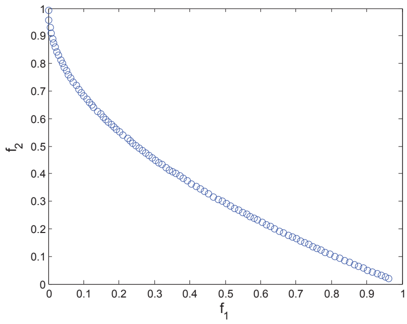
For covered space measure, the S-MOFWA wins on four functions among all the six functions. On ZDT1 and ZDT2, SPEA2 gets the best average covered space with the smallest standard deviations, which means SPEA2 distributes the solutions better along the non-dominated front. The density metric involved in the fitness assignment of SPEA2 may contributes a lot to this ability. On the other four functions, S-MOFWA finds a better spread than other algorithms.

TABLE II
CONVERGENCE MEASURE

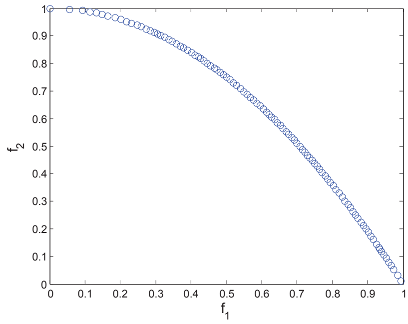
	ZDT1		ZDT2		ZDT3		SCH		KUR		ZDT6	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
S-MOFWA	**9.10E-04**	**1.00E-04**	**8.00E-04**	**0.00E+00**	**3.90E-03**	4.00E-04	**3.20E-03**	**2.00E-04**	**9.00E-03**	1.10E-03	**2.80E-03**	**1.00E-04**
NSGAII	1.40E-03	2.00E-04	1.10E-03	1.00E-04	4.60E-03	3.00E-04	9.90E-03	7.70E-03	1.31E-02	1.40E-03	2.80E-03	2.00E-04
SPEA2	1.30E-03	1.00E-04	8.00E-04	1.00E-04	4.70E-03	**2.00E-04**	1.19E-02	8.10E-03	1.01E-02	**7.00E-04**	2.80E-03	1.00E-04
PESA2	1.60E-03	3.00E-04	1.30E-03	4.00E-04	4.50E-03	4.00E-04	8.50E-03	7.40E-03	1.61E-02	2.40E-03	1.57E-02	2.05E-02

TABLE III
COVERED SPACE MEASURE

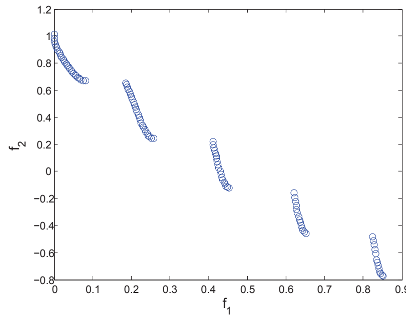
	ZDT1		ZDT2		ZDT3		SCH		KUR		ZDT6	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
S-MOFWA	0.6536	1.03E-02	0.3279	1.50E-03	**0.7795**	3.00E-04	**0.8303**	**1.70E-03**	**0.3558**	8.70E-03	**0.3227**	**0.00E+00**
NSGAII	0.6602	3.00E-04	0.3272	4.00E-04	0.7785	**1.00E-04**	0.811	5.87E-02	0.3435	**6.20E-03**	0.3205	2.00E-04
SPEA2	**0.6616**	**1.00E-04**	**0.3285**	**1.00E-04**	0.7789	1.00E-04	0.8001	5.05E-02	0.3446	6.98E-03	0.3224	2.00E-04
PESA2	0.6252	2.49E-02	0.3092	9.60E-03	0.7703	9.20E-03	0.8082	5.37E-02	0.3516	9.82E-03	0.3131	7.80E-03



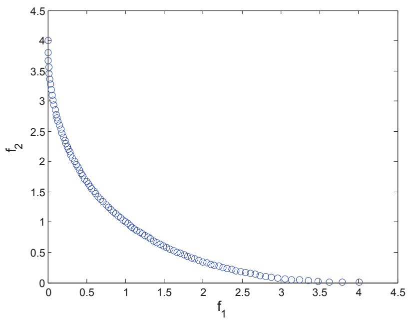
(a) ZDT1



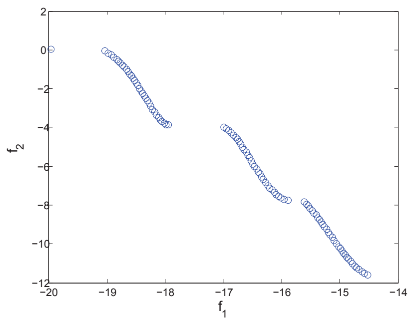
(b) ZDT2



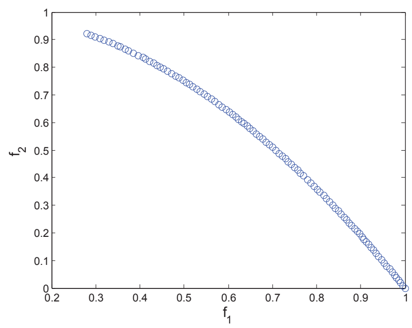
(c) ZDT3



(d) SCH



(e) KUR



(f) ZDT6

Fig. 5. Plots of the final archive of returned by S-MOFWA in the objective space

V. SUMMARY AND FUTURE WORK

In this paper, we presented the S-MOFWA, the S-metric based multi-objective fireworks algorithm. As an indicator based EMOA, the S-metric was introduced to guide the iteration process of fireworks algorithm. The explosion numbers of sparks and explosion amplitudes are calculated according to each fireworks' S-metric. And the fireworks for next iteration are selected by their S-metric. The selection for next generation and explosion to generate sparks was quite simpler compared to the conventional fireworks algorithm. And a novel archive strategy was employed to keep the best solution set, which reduces the labor in traditional archive strategy.

The comparison results with NSGA-II, SPEA2 and PESA2 demonstrated the efficiency of S-MOFWA. On most test functions, S-MOFWA finds a better spread and gets closer to the true Pareto front than the other three algorithms.

Encouraged by the promising performance on these test functions, we intent to employ the proposed S-MOFWA in real world problems, which may have more objectives or constraints in the future.

Acknowledgment

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and 61170057, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China with grant no. 2015CB352300. Prof. Ying Tan is the corresponding author.

References

[1] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," Advances in Swarm Intelligence, pp. 355–364, 2010.

[2] J. Liu, S. Zheng, and Y. Tan, "The improvement on controlling exploration and exploitation of firework algorithm," in Advances in Swarm Intelligence. Springer, 2013, pp. 11–23.

[3] B. Zhang, M. Zhang, and Y.-J. Zheng, "Improving enhanced fireworks algorithm with new gaussian explosion and population selection strategies," in Advances in Swarm Intelligence. Springer, 2014, pp. 53–63.

[4] S. Zheng, A. Janecek, J. Li, and Y. Tan, "Dynamic search in fireworks algorithm," in Evolutionary Computation (CEC), 2014 IEEE Congress on, July 2014, pp. 3222–3229.

[5] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in Evolutionary Computation (CEC), 2014 IEEE Congress on, July 2014, pp. 3214–3221.

[6] H. Gao and M. Diao, "Cultural firework algorithm and its application for digital filters design," International Journal of Modelling, Identification and Control, vol. 14, no. 4, pp. 324–331, 2011.

[7] Y. Zheng, X. Xu, and H. Ling, "A hybrid fireworks optimization method with differential evolution," Neurocomputing, 2012.

[8] B. Zhang, M.-X. Zhang, and Y.-J. Zheng, "A hybrid biogeography-based optimization and fireworks algorithm," in Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 2014, pp. 3200–3206.

[9] C. Yu, J. Li, and Y. Tan, "Improve enhanced fireworks algorithm with differential mutation," in Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, Oct 2014, pp. 264–269.

[10] C. Yu, L. Kelley, S. Zheng, and Y. Tan, "Fireworks algorithm with differential mutation for solving the cec 2014 competition problems," in Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 2014, pp. 3238–3245.

[11] Y. Tan, C. Yu, S. Zheng, and K. Ding, "Introduction to fireworks algorithm," International Journal of Swarm Intelligence Research (IJSIR), vol. 4, no. 4, pp. 39–70, 2013.

[12] K. Ding, S. Zheng, and Y. Tan, "A gpu-based parallel fireworks algorithm for optimization," in Proceedings of the 15th annual conference on Genetic and evolutionary computation. ACM, 2013, pp. 9–16.

[13] A. Janecek and Y. Tan, "Swarm intelligence for non-negative matrix factorization," International Journal of Swarm Intelligence Research (IJSIR), vol. 2, no. 4, pp. 12–34, 2011.

[14] S. Zheng and Y. Tan, "A unified distance measure scheme for orientation coding in identification." in Information Science and Technology, 2013 IEEE Congress on. IEEE, 2013, pp. 979–985.

[15] W. He, G. Mi, and Y. Tan, "Parameter optimization of local-concentration model for spam detection by using fireworks algorithm," in Advances in Swarm Intelligence. Springer, 2013, pp. 439–450.

[16] A. M. Imran, M. Kowsalya, and D. Kothari, "A novel integration technique for optimal network reconfiguration and distributed generation placement in power distribution networks," International Journal of Electrical Power & Energy Systems, vol. 63, pp. 461–472, 2014.

[17] A. Mohamed Imran and M. Kowsalya, "A new power system reconfiguration scheme for power loss minimization and voltage profile enhancement using fireworks algorithm," International Journal of Electrical Power & Energy Systems, vol. 62, pp. 312–322, 2014.

[18] N. Pholdee and S. Bureerat, "Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints," Advances in Engineering Software, vol. 75, pp. 1–13, 2014.

[19] X. Yang and Y. Tan, "Sample index based encoding for clustering using evolutionary computation," in Advances in Swarm Intelligence. Springer, 2014, pp. 489–498.

[20] K. Deb, Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, 2001, vol. 16.

[21] E. J. Hughes, "Msops-ii: A general-purpose many-objective optimiser," in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007, pp. 3944–3951.

[22] ——, "Multiple single objective pareto sampling," in Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, vol. 4. IEEE, 2003, pp. 2678–2684.

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," Evolutionary Computation, IEEE Transactions on, vol. 6, no. 2, pp. 182–197, 2002.

[24] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," 2001.

[25] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in Parallel Problem Solving from Nature-PPSN VIII. Springer, 2004, pp. 832–842.

[26] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithmsła comparative case study," in Parallel problem solving from naturełPPSN V. Springer, 1998, pp. 292–301.

[27] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," Evolutionary Computation, IEEE Transactions on, vol. 13, no. 5, pp. 1075–1082, 2009.

[28] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," Evolutionary Computation, IEEE Transactions on, vol. 7, no. 2, pp. 100–116, 2003.

[29] M. Emmerich, N. Beume, and B. Naujoks, "An emo algorithm using the hypervolume measure as selection criterion," in Evolutionary Multi-Criterion Optimization. Springer, 2005, pp. 62–76.

[30] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in Evolutionary Computation (CEC), 2013 IEEE Congress on. IEEE, 2013, pp. 2069–2077.

[31] H. Gazit, "New upper bounds in klee's measure problem," SIAM Journal on Computing, vol. 20, no. 6, pp. 1034–1045, 1991.

[32] B. Naujoks, N. Beume, and M. Emmerich, "Multi-objective optimisation using s-metric selection: Application to three-dimensional solution spaces," in Evolutionary Computation, 2005. The 2005 IEEE Congress on, vol. 2. IEEE, 2005, pp. 1282–1289.

[33] M. Fleischer, "The measure of pareto optima applications to multi-objective metaheuristics," in Evolutionary multi-criterion optimization. Springer, 2003, pp. 519–533.

[34] J. D. Knowles, "Local-search and hybrid evolutionary algorithms for pareto optimization," Ph.D. dissertation, University of Reading, 2002.

[35] E. Zitzler, "Hypervolume metric calculation," Computer Engineering and Networks Laboratory (TIK), Zürich, 2001.

[36] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," Evolutionary Computation, IEEE Transactions on, vol. 10, no. 1, pp. 29–38, 2006.

[37] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," European Journal of Operational Research, vol. 181, no. 3, pp. 1653–1669, 2007.

[38] I. C. Garcia, C. A. C. Coello, and A. Arias-Montano, "Mopsohv: A new hypervolume-based multi-objective particle swarm optimizer," in Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 2014, pp. 266–273.

[39] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," DTIC Document, Tech. Rep., 1999.

[40] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," Evolutionary computation, vol. 8, no. 2, pp. 173–195, 2000.

[41] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," Advances in Engineering Software, vol. 42, no. 10, pp. 760–771, 2011.

[42] K. Deb, M. Mohan, and S. Mishra, "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions," KanGAL report, vol. 2003002, 2003.