Pareto-Optimality Solution Recommendation Using A Multi-Objective Artificial Wolf-Pack Algorithm

Yi Chen[1,*], Zhonglai Wang[2,†], Erfu Yang[3,‡],Yun Li[1,**]
[1]Institute for Creativity and Design, Dongguan University of Technology, Dongguan 523808, China
[*]email: chenyi@dgut.edu.cn, [**]email: liy@dgut.edu.cn
[2]School of Mechatronics Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[†]email: wzhonglai@uestc.edu.cn
[3]Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK
[‡]email: erfu.yang@strath.ac.uk

***Abstract*—In practical applications, multi-objective optimisation is one of the most challenging problems that engineers face. For this, Pareto-optimality is the most widely adopted concept, which is a set of optimal trade-offs between conflicting objectives without committing to a recommendation for decision-making. In this paper, a fast approach to Pareto-optimal solution recommendation is developed. It recommends an optimal ranking for decision-makers using a Pareto reliability index. Further, a mean average precision and a mean standard deviation are utilised to gauge the trend of the evolutionary process. A multi-objective artificial wolf-pack algorithm is thus developed to handle the multi-objective problem using a non-dominated sorting method (MAWNS). This is tested in a case study, where the MAWNS is employed as an optimiser for a widely adopted standard test problem, ZDT6. The results show that the proposed method works valuably for the multi-objective optimisations.**

I. Introduction

Real-world applications, such as structural optimisations of space systems[1], parameters determination for financial market quantitative modelling, Terahertz spectroscopic analysis for drug or explosive mixture[2], and intelligent analysis for 'Big Data' management, often imply multiple objectives. Thus, people need to search for 'trade-offs', rather than a single solution, which leads to the different solution of 'optimality' under the multi-objective situations. The most widely used term is the notion of Pareto optimality [3], [4], [5].

A multi-objective optimisation problem could be written in Equ. (1),

$$\begin{aligned}&maximise:\\&F_i\left(\mathbf{x}\right)=\left[f_1\left(\mathbf{x}\right),f_2\left(\mathbf{x}\right),\cdots,f_N\left(\mathbf{x}\right)\right],\\&i=1,2,\cdots,N\end{aligned}\tag{1}$$

Subject to the equality constraints $G_i\left(\mathbf{x}\right)$, as given in equation (2),

$$G_i\left(\mathbf{x}\right)=0, i=1,2,\cdots,M\tag{2}$$

and the inequality constraints $H_i\left(\mathbf{x}\right)$, as given in equation (3),

$$H_i\left(\mathbf{x}\right)\leq 0, i=1,2,\cdots,P\tag{3}$$

where, N is the number of objective functions, the objective functions F_i $\Re^n\rightarrow\Re$; M is the number of the equality constraints; P is the number of the inequality constraints; $\mathbf{x}$ = $[x_1, x_2, \cdots, x_K]$ is the decision variables vector, K is the number of the variables.

All vectors satisfied equations (2) and (3) are named as the set $\mathfrak{F}$, in which the particular set of $\mathbf{x}^*$ = $[x_1^*, x_2^*, \cdots, x_K^*]$ yields the optimum values of all the objectives. The vector of decision variables $\mathbf{x}^* \in \mathfrak{F}$ is Pareto optimal if there is no feasible vector of decision variables $\mathbf{x} \in \mathfrak{F}$ which would increase some criterion without causing a simultaneous decrease in any other criterion. The vectors $\mathbf{x}^*$ corresponding to the solutions included in the Pareto optimal set are called non-dominated. The image of the Pareto optimal set under the objective functions is called Pareto front[3], [4], [5].

There are some widely used algorithms to solve multi-objective formulations, such as the non-dominated sorting genetic algorithm (NSGA)[6], the niched-Pareto genetic algorithm (NPGA)[7], the multi-objective genetic algorithm (MOGA)[8], the strength Pareto evolutionary algorithm (SPEA)[9], the strength Pareto evolutionary algorithm 2 (SPEA2)[10], the Pareto archived evolution strategy (PAES)[11], the nondominated sorting genetic algorithm II (NSGA-II)[12], [13], the niched Pareto genetic algorithm 2 (NPGA2)[14], the Pareto envelope-based selection algorithm (PESA)[15], the revised version of Pareto envelope-based selection algorithm (PESA-II)[16], the micro-genetic algorithm(μGA)[17], [18], the micro-genetic algorithm with variable population size(VPμGA) [19] and the bat algorithm for multi-objective optimisation [20], [21], etc.

Generally, an engineer can make trade-offs within this set under practical requirements by focusing the set of Pareto front choices, which provides a visualised demonstration of the Pareto-optimal solution, but with an unclear indication of optimal diversities for decision-making. In this paper, we propose a fast approach of Pareto-optimal solution recommendation (FPR) using the Pareto reliability index (PRI), which provides users with a recommendation list of optimal ranking and optimal trend indications with different risk tolerance.

Wolves are always regarded as one of the smartest animals on Earth, and wolves are gregarious animals who mostly live in packs - wolf-pack (WP). Inspired by the swarm intelligence of the WP's dynamic behaviours, a multi-objective artificial wolf-pack algorithm (MAWPA) has been developed, which can response quickly to the environmental changes and their neighbours in the direction and speed, etc., the information of their behaviours can be transferred to others and help them move from one swarming configuration to another almost as one unit. By borrowing this intelligence of the social behaviours, the MAWPA is parallel and independent to the initial values, and able to achieve a global optimum. In this paper, the MAWPA is proposed to handle multi-objective optimisations, which is employed for a standard test problem ZDT6.

The remainder of this paper is organised as follow. Section I introduces the background of this research work; Section II defines the technical work-flow of MAWPA; Section III describes the FPR technical roadmap; Section IV states the normalisation for the fitness functions; Section V defines the PRI factor; Section VI introduces two trend indices for the evolutionary process; Section VII gives two case studies to demonstrate the FPR method; Section VIII concludes this paper.

II. Multi-Objective Artificial Wolf-Pack Algorithm

Inspired by the swarm intelligence of the WP's dynamic behaviours, the MAWPA is an artificial intelligent algorithm that firstly simulates the behaviour of an individual artificial Wolf (AW) and constructs a WP. Each AW searches its own local optimal solution and passes information to its self-organised WP, and finally, achieves the global optimal solution. The MAWPA work-flow is given in Fig. 1, which includes 6 steps of operations: (1) initialisation; (2) behaviour selection; (3) behaviour of scouting; (4) behaviour of calling; (5) behaviour of besieging; (6) bulletin and (7) non-dominated sorting using the non-dominated sorting genetic algorithm II (NSGA-II)[13]. .

Initialisation: in this step, all the parameters will be initialised, and the programme is preparing itself for the next steps.

Behaviour Selection: the behaviour selection step takes 'Scouting' as the default behaviour or initial behaviour for each WP. According to the density of prey in this region, the number of companion and the visual conditions.

Scouting: for a certain AW individual k, $S_k = \{s_1, \ldots, s_M\}$ is its finite state set, there is M states that an AW can perform in. Within the AW's visual field, if the current state of this AW is S_i and the next state is S_j, the AW moves from S_i to S_j randomly and check the state updating conditions as stated in Equations (4) and (5). As demonstrated in Fig. 2, $r_{ij} = \|S_j - S_i\|$ is the distance between the i^{th} and j^{th} individual AW. $F = f(S)$ is the prey density for this AW, where F is the fitness function. δ is the iterate step, υ is the AW visual constant. ϵ is the random moving factor.

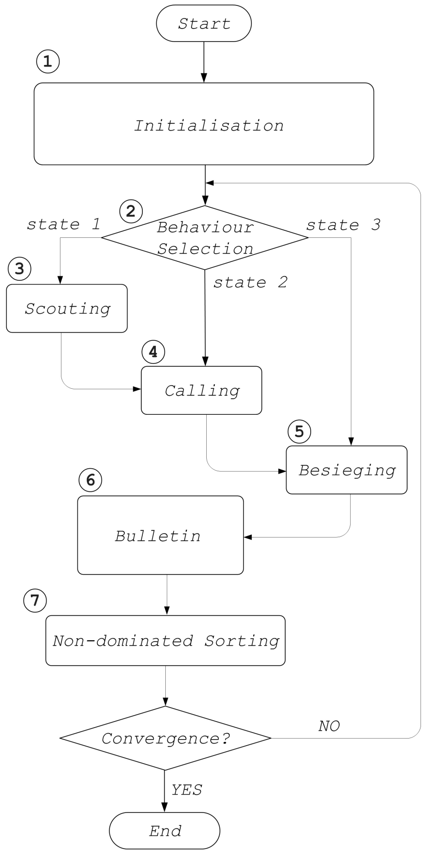

Fig. 1. The workflow of multi-objective artificial wolf pack algorithm

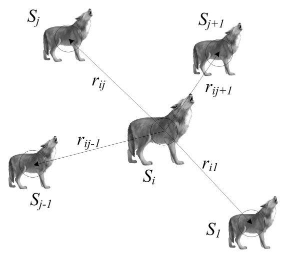

Fig. 2. The state distance between the i^{th} and j^{th} individual

$$S_{i+1} = \begin{cases} S_i + \epsilon \cdot \delta \cdot \dfrac{S_j - S_i}{\|S_j - S_i\|} & \text{if } F_j > F_i \\ S_i + \epsilon \cdot \delta & \text{otherwise} \end{cases} \tag{4}$$

$$S_j = S_i + \epsilon \cdot \upsilon \tag{5}$$

Calling: suppose the number of this AW's neighbours is γ, the central state is S_c, the prey density is $F_c = f(S_c)$ and η is the crowd factor. Within its visual field ($r_{ij} < \upsilon$), if the $F_c/\gamma > \eta F_i$ and $\eta \geq 1$, the AW implements the central state driven step; otherwise, when the $F_c/\gamma \leq \eta F_i$ or $\eta = 1$, the AW will go on with the scouting behaviour, as expressed in Equation (6).

$$S_{i+1} = \begin{cases} S_i + \epsilon \cdot \delta \cdot \dfrac{S_c - S_i}{\|S_c - S_i\|} & \text{if } \dfrac{F_c}{\gamma} > \eta F_i \quad and \quad \eta \geq 1 \\ (4) & \dfrac{F_c}{\gamma} \leq \eta F_i \quad or \quad \eta = 0 \end{cases} \tag{6}$$

Besieging: when the AW's companions reach "max" state S_{max} with the number γ within the neighbourhood, the prey density reaches F_{max} at the mean time. As stated in Equation (7), with the same conditions as Equation (6), the AW updates its state in highest prey density region; otherwise, the AW will go on with the searching behaviour, as expressed in Equation (6).

$$S_{i+1} = \begin{cases} S_i + \epsilon \cdot \delta \cdot \dfrac{S_{max} - S_i}{\|S_{max} - S_i\|} & \text{if } \dfrac{F_{max}}{\gamma} > \eta F_i \quad and \quad \eta \geq 1 \\ (4) & \dfrac{F_{max}}{\gamma} \leq \eta F_i \quad or \quad \eta = 0 \end{cases} \tag{7}$$

Bulletin: the bulletin operation is a step to compare each AW's current state S_i with the historical state data, the bulletin data will be replaced and updated only when the current state is better than the last one, as described by Equation (8).

$$S_{j+1} = \begin{cases} S_j & \text{if } F_j > F_i \\ S_i & \text{otherwise} \end{cases} \tag{8}$$

A "max-generation" of simulation is employed as the terminal condition of the MAWPA programme, which is one of the widely used criteria for optimisation.

III. Fast Approach of Pareto-optimal Solution Recommendation

As shown in Fig. 3, the flow chart of the fast approach of Pareto-optimal solution recommendation is divided into 6 steps:

- step 1, initialise parameters and start the optimisation process;
- step 2, perform optimisation using multi-objective algorithms;
- step 3, Pareto-optimal solutions generation;
- step 4, the PRI assessment block;
- step 5, check optimisation termination conditions;
- step 6, end the programme and post-calculation process.

As can be seen from Fig. 3, there are 4 sub-steps in the FPR assessment block in step 4,

- sub-step 1, normalisation for the multi-objective fitness functions of the Pareto-optimal solutions;
- sub-step 2, the PRI index β_1 calculation;
- sub-step 3, calculation of the evolutionary trend indices, the mean average precision(mAP) and the mean standard deviation(mSTD);
- sub-step 4, visualisation of Pareto front and evolutionary trend indices;

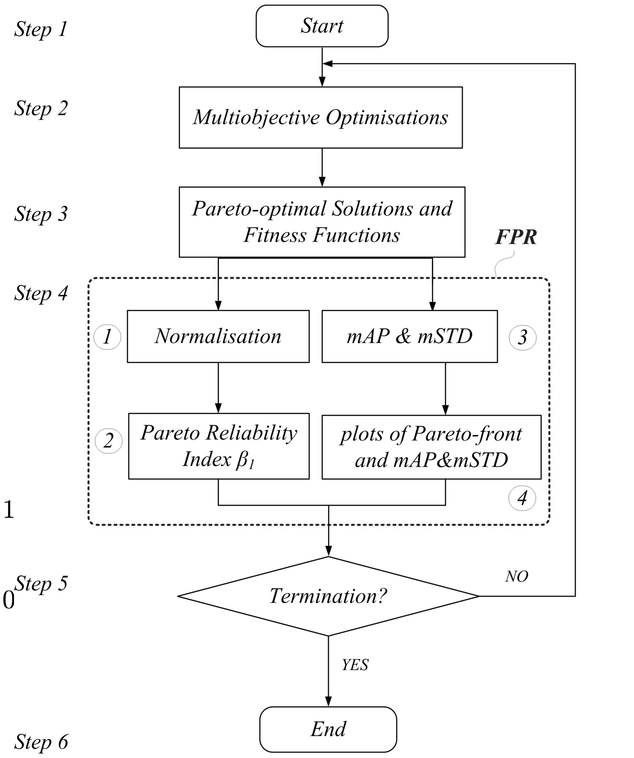

Fig. 3. Work-flow of the fast approach of Pareto-optimal solution recommendation [22]

IV. Normalisation

The normalisation process is to map variables from their original value range to a normalised value range, e.g. $[0, 1]$, by two operations of *scale* and *shift*[23]. As defined in equation (1), the vector of objective functions F_i with the values $[f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_N(\mathbf{x})]$ is the original data source to the normalisation block, in which $f_i \in [f_{min}, f_{max}]$.

Firstly, as given in equation (9), the *scale* operation calculates the scale factor according to the input range $[f_{min}, f_{max}]$ of the original data f_i, and then all the input data are scaled to the range of $[c_l, c_u]$. That is, the fitness values are mapped from the practical value range $[f_{min}, f_{max}]$ to the normalised value range $[c_l, c_u]$, which are $[0, 1]$ in this context.

$$f_{i_c} = (c_u - c_l) \times \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{9}$$

Then, in the *shift* operation, the scaled data f_{i_c} are shifted to the new range of $[c_l, c_u]$, as given in equation (10), where f_{i_h} is the normalised fitness objectives.

$$f_{i_h} = c_l + f_{i_c} \tag{10}$$

V. Pareto Reliability Index for Pareto Solutions

The Pareto reliability index (β_1) is defined in equation (11), where μ_f is the mean and σ_f is the standard deviation of the normalised objectives, as given in equations (12) and (13) respectively. W_i is the weighted normalised objectives in equation (14), w_i is the weight factor as given in equation (15), which balances the weight of all the normalised objectives, $w_i \in [0,1]$.

$$\beta_1 = \frac{\mu_f}{\sigma_f} \tag{11}$$

$$\mu_f = \frac{\sum_i^N W_i}{N} \tag{12}$$

$$\sigma_f^2 = \frac{\sum_i^N (W_i - \mu_f)^2}{N-1} \tag{13}$$

$$W_i = w_i \cdot f_{i_h} \tag{14}$$

$$\sum_i^N w_i = 1 \tag{15}$$

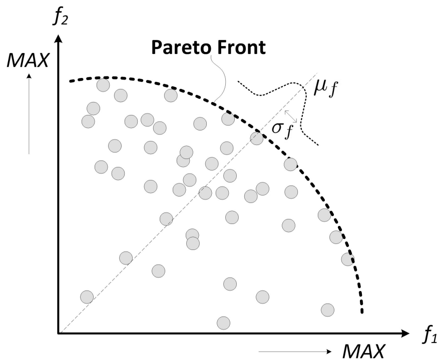

Fig. 4. Pareto Reliability Index[22]

As shown in Fig. 4, without loss of generality, a case of two objectives f_1 and f_2 is utilised to present the definition of β_1. Fig. 4 shows a geometrical illustration of the β_1 index in a dual-objective case, which indicates the distance of the mean margin of a multi-criteria range. The idea behind the β_1 is that the distance from location measure μ_f to the limit states σ_f which provides a good measure of the reliability of the Pareto solutions, that is, a larger value of β_1 leads to a better solution.

VI. Trend Indices

In this section, a factor of mAP and a factor of mSTD are introduced as the trend indices for the optimisation process, which are defined in equations (16) and (17).

As shown in Fig. 5, the solid curve is the mAP scores for each vector f_j as given in equation (16) and the dashed curves are the mAP $\pm$ mSTD for each vector f_j as given in equation (17), in which p is the population of the data set, *AVG*$(\cdot)$ is the average function and *VAR*$(\cdot)$ is the variance function.

$$\text{mAP}(f_j) = \frac{1}{p}\sum_{j=1}^{p}(AVG(f_j)) \tag{16}$$

$$\text{mSTD}(f_j) = \frac{1}{p}\sum_{j=1}^{p}\left(\sqrt{VAR(f_j)}\right) \tag{17}$$

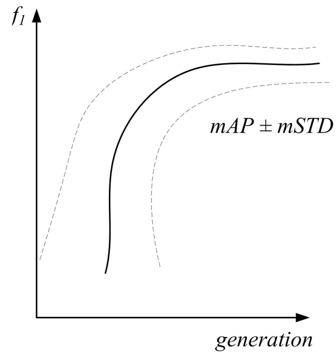

Fig. 5. The diagram of mAP $\pm$ mSTD over the full generations[22]

VII. Case Study

As given by equation (18), a standard test problem ZDT6 [24] is solved by NSGA-II implemented in the MATLAB toolboxes $SGALAB$ [19], [25], $Swarmwolf$[26] and $SECFLAB$[27], in which $x_i \in$ [0,1], n = 10 in this context.

$$\begin{cases} f_1(x) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ f_2(x) &= g(x)\left[1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right] \\ g(x) &= 1 + 9\left[\frac{\sum_{i=2}^{n} x_i}{n-1}\right]^{0.25} \end{cases} \tag{18}$$

The parameters for the case are listed in Table I, in which a max-generation 200 is the termination condition of each round test; the total test number is 10; the population is 30, tournament selection operator, binary encoding/decoding method, single point crossover and mutation operators with p_c = 0.8 and p_m = 0.01 respectively.

TABLE I
Parameters for MAWNS optimisation

	multi-objective algorithm	NSGA-II
	max-generation	200
	crossover probability (p_c)	0.8
	mutation probability (p_m)	0.01
	population	60
	test-number	10
	selection operator	tournament
	crossover operator	single point
	mutation operator	single point
	encoding/decoding method	binary
P_N	non-replaceable population	50
P_R	replaceable population	10
δ	iterate step	0.5
υ	visual	2.5
η	crowd	0.618
	try number	5

Figs. 6 and 7 are the mAP $\pm$ mSTD diagrams for f_1 and f_2 over the full simulation generations, which indicate that a better solution of f_1 can be optimised without worsening a solution of f_2, which is not dominated by any other solution in the search space. As can be seen in Fig. 6, the f_1's mAP $\pm$ mSTD curves go up quickly from generation = 1 to 8, when they reach a stable status with a minor fluctuation at generation = 10 and last to the end of simulation. Fig. 7 shows the f_2's mAP $\pm$ mSTD curves have a similar shape of generation = 1 to 8, and they have a second jump from generation = 100 to 130 and then keep stale to the end of simulation.

As shown in Table II, the recommended solutions are listed by the solution number in the column 'SOLUTION No.' with a descending rank (RANK = 1 is the most recommended) using the values of β_1.

TABLE II
SOLUTIONS RECOMMENDATION FOR THE ZDT6 TEST PROBLEM

RANK	SOLUTION No.	β_1
1	41	2.25e15
2	43	2.25e15
3	43	2.25e15
...	...	...
21	218	6.83
22	219	6.83
23	220	6.83
...	...	...
44	15	4.63
45	16	4.63
46	17	4.63
...	...	...
300	270	0.499

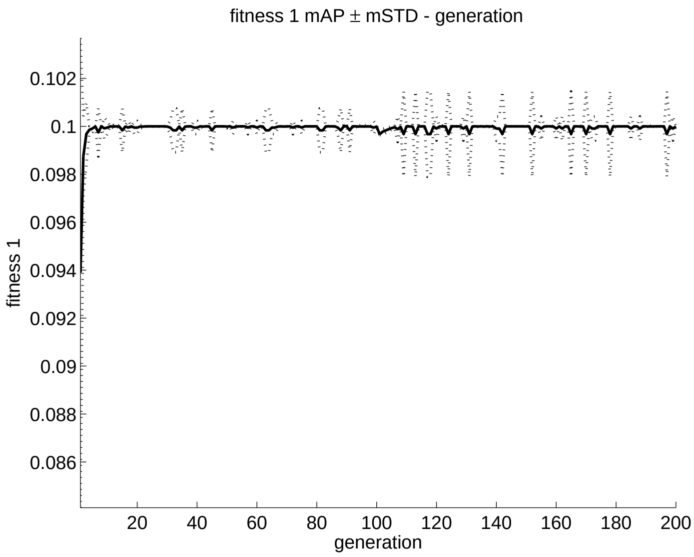

Fig. 6. f_1's mAP $\pm$ mSTD over the full generations

VIII. CONCLUSIONS AND FUTURE WORKS

Using the newly defined index of β_1, a fast approach to Pareto-optimal solutions recommendation has been developed, thereby providing a ranking list of Pareto-optimal solutions for the decision-making. The evolutionary trends are gauged via the indices of mAP $\pm$ mSTD with variable uncertainty tolerances.

The contributions of this paper includes: (1) the inclusion of the dynamic behaviours of trend indices of mAP and mSTD; (2) the development of a fast Pareto-optimality solution recommendation method, FPR; (3) the Pareto reliability index β_1 or PRI to rank the uncertainties of Pareto-optimal solutions, and a clear recommendation list for decision-making; (4) a multi-objective artificial wolf-pack swarm algorithm using non-dominated sorting method, MAWNS.

Further work aims at industrial applications,including the multi-objective optimisations for robotic systems, such as exoskeleton, robotic space tethers, humanoid robot and industrial robotics. Further, it will be applied to decision processes for computational intelligence aided design [28][29].

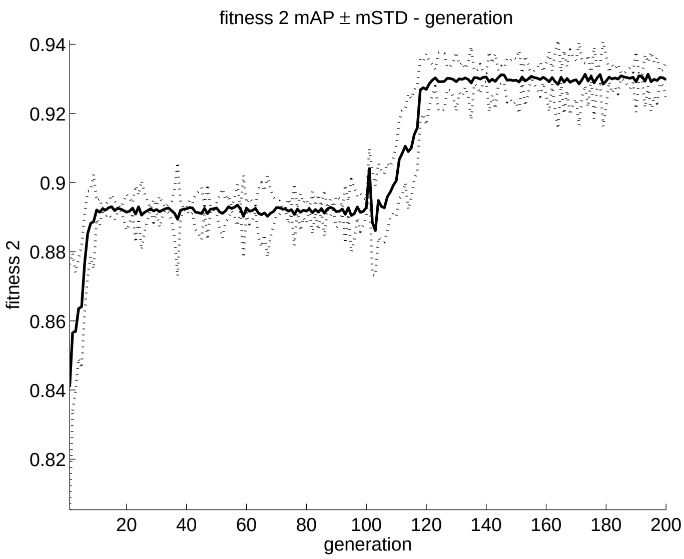

Fig. 7. f_2's mAP $\pm$ mSTD over the full generations

REFERENCES

[1] Chen, Y., Cartmell, M.P.: Multi-objective Optimisation On Motorised Momentum Exchange Tether for Payload Orbital Transfer. In Proceedings of IEEE Congress on Evolutionary Computation, pp. 987-993, (2007)

[2] Chen, Y., Ma, Y., Lu, Z., Qiu, L.X., He, J.: Terahertz Spectroscopic Uncertainty Analysis For Explosive Mixture Components Determination Using Multi-objective Micro Genetic Algorithm. Advances in Engineering Software, vol.42, no.9, pp. 649-659 (2011)

[3] Edgeworth, F.Y.: Mathematical Physics. P. Keagan, London, England, (1881)

[4] Pareto, V.: Cours d'Économie Politique, volume I and II. F. Rouge, Lausanne (1896)

[5] Stadler, W.: Fundamentals of multicriteria optimisation. In: Stadler, W.(ed.), Multicriteria Optimization in Engineering and the Sciences, pp. 1-25. Plenum Press, New York, NY (1988)

[6] Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, vol.2, no.3, pp.221-248 (1994)

[7] Horn, J., Nafpliotis, N., and Goldberg, D. E.: A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, vol. 1, pp. 82-87 (1994)

[8] Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: Forrest, S.(ed), Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423. San Mateo, CA, Morgan Kaufmann Publishers (1993)
[9] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactionson Evolutionary Computation 3(4), 257-271 (1999)
[10] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)
[11] Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation 8(2), 149-172 (2000)
[12] Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India (2000)
[13] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp.182-197 (2002)
[14] Erickson, M., Mayer, A., Horn, J.: The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D. (eds.), First International Conference on Evolutionary Multi-Criterion Optimization, pp. 681-695. Springer-Verlag. Lecture Notes in Computer Science No. 1993 (2001)
[15] Corne, D.W., Knowles, J.D., Oates, M.J.: The Pareto envelope-based selection algorithm for multiobjective optimization. In: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., Schwefel, H. P. (eds.), Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 839-848. Paris, France. Springer. Lecture Notes in Computer Science No. 1917 (2000)
[16] Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.: PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.), Proceedings of the genetic and evolutionary computation conference (GECCO-2001), pp. 283-290. San Francisco, California. Morgan Kaufmann Publishers (2001)
[17] Coello Coello, C.A., Toscano Pulido, G.: A micro-genetic algorithm for multiobjective optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D. (eds.), First International Conference on Evolutionary Multi-Criterion Optimization, pp. 126-140. Springer-Verlag. Lecture Notes in Computer Science No. 1993 (2001)
[18] Coello Coello, C.A., Toscano Pulido, G.: Multiobjective optimization using a micro-genetic algorithm. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001), pp. 274-282. San Francisco, California. Morgan Kaufmann Publishers (2001)
[19] Chen, Y., Song, Z.-J.: Spatial analysis for functional region of suburban-rural area using micro genetic algorithm with variable population size. Expert Systems with Applications, vol. 39, no. 7, pp.6469-6475 (2012)
[20] Yang, X.S.: Bat algorithm for multi-objective optimization. Int. J. Bio-Inspired Computation, vol. 3, no. 5, pp.267-274 (2011)
[21] Chen, Y.: SwarmBat-the artificial bat algorithm, http://www.mathworks.de/matlabcentral/fileexchange/39116-swarmbat-the-artificial-bat-algorithm-aba (2012)
[22] Chen, Y., Peng, B., Hao X.-H., Xie G.-N.: Fast Approach of Pareto-optimal Solution Recommendation To Multi-objective Optimal Design of Serpentine-channel Heat Sink, Applied Thermal Engineering, vol. 70, iss. 1, pp. 263-273 (2014)
[23] Chen, Y.: Dynamical Modelling of A Flexible Motorised Momentum Exchange Tether and Hybrid Fuzzy Sliding Mode Control for Spin-up. PhD Thesis, Mechanical Engineering Department, University of Glasgow, Glasgow (2010)
[24] Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Technical Report 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (1999)
[25] Chen, Y.: SGALAB - Simple Genetic Algorithm Laboratory Toolbox, http://www.mathworks.co.uk/matlabcentral/fileexchange/5882 (2009)
[26] Chen, Y.: SwarmWolf – The Artificial Wolf Pack Algorithm (AWPA), https://uk.mathworks.com/matlabcentral/fileexchange/48469 (2014)
[27] Chen, Y.: SECFLAB - Simple Econometrics And Computational Finance Laboratory Toolbox, http://www.mathworks.com/matlabcentral/fileexchange/38120 (2012)
[28] Chen, Y., Zhang, G.F., Jin, T.D., Wu, S.M., Peng, B.: Quantitative modelling of electricity consumption using computational intelligence aided design, Journal of Cleaner Production, vol. 69, pp. 143-152 (2014)
[29] Chen, Y., Peng, B.: Multi-objective optimization on multi-layer configuration of cathode electrode for polymer electrolyte fuel cells via computational-intelligence-aided design and engineering framework. Applied Soft Computing. vol. 43, pp. 357-371 (2016)