General Multi-Objective Performance Expression for Population-Based Search and Optimization

Eik Fun Khor

[1]Institute of Technical Education, 1 Choa Chu Kang Grove, Singapore 688236
Email: *khor_eik_fun[at]ite.edu.sg*

Abstract: ***This paper introduces formal syntax to construct multi-objective performance expression. The syntax is general, flexible and intuitive for decision makers to reflect variety of performance assessment for high-level decision support in the multi-objective optimization. The expression enables decision makers to elaborate their preferences using logical relations such as "AND", "OR" and "NOT", and even priority order among the objectives. Each objective allows any possible combination of specifications augmented to the objective function such as superiority threshold, satisfactory level and goal for more informative search requirements as and when is available and required. Besides, the MO performance expression can be used in various population-based optimization. In addition to numerical illustrations, the usefulness of proposed expression is demonstrated in practical application of multi-objective student internship planning to search for optimum matching of intern jobs and students in a batch where preferences can be expressed in various ways preferable by decision makers based on their needs and interests.***

Keywords: multi-objective decision making, performance evaluation, performance ranking, search and optimization, population-based optimization, computational intelligence

1. Introduction

Many real-world optimizations involve searching for a vector of optimal design variables to minimize certain decision or design cost based upon a scalar function to support decision making. For instance, in resource planning, planners often involve in the task of finding the most suitable loading of jobs to resources such as human operators, machines, transportations to achieve the best process performance in terms of completion times, running costs and so forth. In some cases, however, quality of the process performance may not be quantified as a simple objective function since the decision quality may reflect different aspects of specifications that may be competing or non-commensurable to each other. To sustain the decision quality, each objective function needs to be considered explicitly when searching for the set of optimal decision variables. This type of problem is known as multi-objective (MO) optimization problem in literature. Instead of combining the various objective functions, each objective function is treated separately in the optimization process, and the solution set is often a family of points known as the Pareto-optimal set. Each objective component of any point in the Pareto-front can only be improved by degrading at least one of the other objective components [1-5].

Evolutionary algorithms (EAs) based intelligent search techniques [6,7] that mimic the mechanics of natural selection and evolution have been found to be very effective population-based optimization for searching a set of globally optimized trade-off solutions simultaneously. The approach has several variants and can be traced back to their early developments such as the multi-objective genetic algorithm (MOGA) [8], nondominated sorting genetic algorithm (NSGA) [9], niched Pareto genetic algorithm (NPGA) [10], nongenerational EA [11], strength Pareto EA (SPEA) [12], MOEA [13] and others. Unlike traditional gradient-guided search techniques, EAs require no derivative information of the search points, and thus require no stringent conditions on the objective function, such as to be well-behaved or differentiable. Owing to the above reasons, EA-based search methods have been applied to solve optimization problems in various disciplines, such as resource management [14]; environmental sustainability [15,16], cancer treatment in medical fields [17]; controller design automation in engineering [1,18-20]; physiological processes in biology [21]; economics and finance [22] and etc.

In practical multi-objective optimization problems, it is desired to consider decision-maker (DM) preferences in order to propose the best compromise solutions. In addition to specifying a list of objective functions, it is often required by DM to include other information to better elaborate their expectations more precisely and informatively so that the optimized solutions are closer to their preferences. Having the above objectives, Fonseca and Fleming [8] modified MO ranking scheme to include goal information. Khor et al., [23] formulate the domination scheme to include goal information as well as extending the ranking algorithm to incorporate priority information among the objective functions. Deb and Kumar [24] use reference direction method from the multi-criterion decision-making literature and combine it with an evolutionary procedure to develop an algorithm for finding a single preferred solution in a multi-objective optimization scenario. Jamwal et al. [25] use Fuzzy sorting approach for multi-objective performance assessment which require prior knowledge of the range of objective function to define fuzzy sets. Lai et al. [26] study a class of mixed Pareto-Lexicographic multi-objective optimization problems where the preference among the objectives is available in different priority levels (PLs). More research works on incorporating different preference models in multi-objective evolutionary algorithms can be found in [27-29].

This paper is devoted to look into developing syntactic construct and formulation of MO performance evaluation to incorporate DM preferences in more general and intuitive format with optional prior knowledge, if available, to express their preferences. The general MO performance expression allows DM to elaborate their preferences in terms of logical relations such as "AND", "OR" and "NOT", as well as priority order among the objectives in formal syntactic construct. Each objective comprises a cost function and any possible combination of augmented specifications to better model the search requirements. They include superiority threshold (smallest difference to consider better than), satisfactory level (a level that is considered good enough) and goal (desired target to attain) information. The augmented specifications are optional for DM as and when is needed. Repeating objective(s) in the expression is allowed for DM depends on their preferences. Once constructed, DM is not required to make further decision at all apart from letting the optimizer to search for optimal solutions based on their constructed MO performance expression. The paper is organized as follow. Section 2 provides an overview of conventional MO optimization model while section 3 presents a generalized model for DM to have wider choices to express their optimization preference. Section 4 formulates the evaluation components and operations as building blocks for the proposed General MO Expression while section 5 presents the syntax of the expression with illustrative examples and explanations. Section 6 describes the evolutionary optimization process for use with the proposed performance expression in latter sections. Its implementation and usefulness in practical application is demonstrated in section 7, which includes the problem modeling, experimental results and discussion on different scenario of DM preferences in MO performance expressions. Last but not least, conclusions are drawn in final section.

2. Conventional MO Optimization Problem

Standard multi-objective minimization problem tends to minimize multiple cost functions as follows:

$$\begin{aligned}&\text{Minimize:}\\&\qquad \left(f_1(X), f_2(X), \ldots, f_C(X)\right)\\&\text{Subject to:}\\&\qquad g_j(X) \le 0 \qquad j = 1,2,\ldots,J\\&\qquad x_i^{(L)} \le x_i \le x_i^{(U)} \quad i = 1,2,\ldots,D\end{aligned} \tag{1}$$

where C, J and D are the number of cost functions, constraints and decision variables, respectively. The cost functions and constraints may be competing and non-commensurable to each other. The expected results of the minimization problem are a set of solutions that are not dominated by any other solutions in the population, which are said to be non-dominated or Pareto front.

3. Generalized MO Optimization Problem

This section takes a step further to generalize MO optimization to cover gibber scope of DM specifications and preferences. Here instead of minimizing individual cost functions subjected to constraints in the above conventional problem definition, our first step is to augment DM's practical and intuitive specifications, as listed below, into cost function:

i. Superiority threshold (α) to indicate a minimum difference in f that a function f_a is considered superior than f_b *iff* $f_a < f_b - \alpha$. This is for DM to control the sensitivity in comparison between two values in the same cost function.

ii. Satisfactory level (β) to define the satisfactory level of achievement. Once the satisfactory level is met, further improvement in the cost function is not a requirement to DM. Using this parameter enable more room for improvement on other objectives with satisfactory level either undefined or is not met.

iii. Goal (γ) to provide directional guide in improving towards desired trade-off region in multi-objective domains. It is different from the satisfactory level where there is no directional guide provided in objective spaces. In some optimization problems, especially when there are many optimization functions, the global trade-off may be too broad to cover, as though there is no clear direction in the optimization. As directional guide, the goal setting can be feasible or infeasible goal, which omit the requirement for DM to have prior knowledge on the feasible trade-off region. If the goal is feasible, the optimal solutions found will be located within the goal region confined by the goal. On the other hand for unfeasible goal, the optimal solutions will be outside but in the shortest distance possible from the unfeasible goal in objective space. Unlike satisfactory level, the optimization still continues even the goal settings are met.

For general optimization, let an objective, θ, be an ordered 4-tuple as (f, α, β, γ) where f is the cost function to minimize, α the superiority threshold, β the satisfactory level and γ the goal for the objective. An objective in this paper comprises a cost function together with augmented specifications for the cost function. Two objectives, θ_a and θ_b, are equal *iff* $f_a=f_b$, $\alpha_a=\alpha_b$, $\beta_a=\beta_b$ and $\gamma_a=\gamma_b$. It can be noted that two objectives are considered different if and only if any of the augmented specifications is different, even though their cost function is identical. Cost function is compulsory attribute for an objective. While for other attributes, such as α, β and γ, are optional attributes. Assigning "null" value to an optional attribute by DM indicates that the attribute is undefined, or not specified. For example, a partially defined objective θ = (f, null, β, γ) indicates α is null (undefined) while other attributes are defined. The same applied to β and γ when they are not defined in an objective. Thus, the least defined objective can be reduced to (f, null, null, null) when it is purely to minimize the cost function f without other augmented specifications of α, β and γ.

Having defined the objective and kept the problem in a general form, a multi-objective optimization problem can be generalized as a problem to minimize an expression Ψ of objectives as follow:

$$\text{Minimize: } \Psi(\langle\boldsymbol{\theta}\rangle,\ \Omega) \tag{2}$$

Here, Ψ denotes a syntactic construct in the general MO expression. $\langle\boldsymbol{\theta}\rangle$ denotes the ranks with respect to set of objectives $\boldsymbol{\theta}$ in the optimization problem. Ω is a set of defined logical operations in the general expression which will be explain in following sections. It serves as formal descriptive expression by DM on how MO performance among different objectives are to be evaluated according to DM's preferences.

Before we proceed, lets define some common symbols and notations used in the paper. Given a general optimization problem where M denotes number of objectives and C the number of cost functions. $\boldsymbol{\theta} = \{\theta_i: \forall\ i = 1, 2, \ldots M\}$ is a set of objectives. $\boldsymbol{F} = \{f_j: \forall\ j = 1, 2, \ldots C\}$ is a set of cost function. $\alpha_{j,k}$ is the superiority threshold k for cost function f_j. $\beta_{j,r}$ is the satisfactory level r for cost function f_j. $\gamma_{j,s}$ is the goal s for cost function f_j. S is the number of individuals in a population. $R = \{r_i: \forall\ i = 1, 2, \ldots S\}$ is a rank vector r_i is a rank value of individual i.

4. Performance Evaluation for General MO Expression

When evaluating individual performance described by DM in the general expression Ψ, the evaluation process begins with ranking operation as describe below:

4.1 Ranking Operation

The operands in Ψ is rank vectors R as defined in previous section while the ranking operation is applied to convert a given subset of objectives, $\boldsymbol{\theta}_i \in \boldsymbol{\theta}$, to rank vector R_i as below:

$$\langle \boldsymbol{\theta}_i \rangle = R_i \tag{3}$$

As the inputs to the generalized ranking operation are objectives, as defined above, rather than simply cost functions, the ranking operations must be able to incorporate augmented specifications, i.e. α, β and γ for each objective. Since they are optional, the ranking method require flexibility in only consider what is defined and accept what is not defined from DM. The rank value r_j of individual j in an objective set $\boldsymbol{\theta}_i$ is given by:

$$\begin{gathered} r_j = 1 + dom_j^{(\boldsymbol{\theta}_i)} \\ r_i \in \mathbb{Z}^+, 1 \le r_j \le S \end{gathered} \tag{4}$$

where $dom_j^{(\boldsymbol{\theta}_i)}$ denotes the number of individuals nominating j-individual. r_j takes any positive integer ranging from 1 to the number, S, of individuals in the group. Ranking operation can be nested. For example, besides objective, we can also apply the same ranking operation in (3) & (4), i.e. $\langle R_{in} \rangle$, on a rank vector R_{in} by treating it as an objective with its cost function equal R_{in} and the augmented specifications, i.e. α, β and γ, undefined. The idea is to convert any rank vector into the rank ranking format in (4).

$$\langle R_1, R_2, \ldots \rangle = R_{out} \text{ or } \langle R_{in} \rangle = \langle \theta_{in} \rangle \tag{5}$$

4.2 Domination Scheme for Flexible Handling of Superiority Threshold, Satisfactory level and Goal

In the above ranking, domination scheme is required to compare any two individuals with respective to the objectives and determine if an individual dominate the other. For conventional MO optimization without further requirement specified on each objective function, domination comparison scheme that directly utilizes the concept of Pareto-dominance [10] is sufficient. However, this may not be true when DM has more information to add into the MO optimization to instruct the search according to their desires. Thus, in general MO optimization in this paper, the coverage must be enough to handle any defined augmented specifications such as superiority threshold, satisfactory levels and goals from DM apart from cost function alone. To address the above requirement, the domination comparison scheme needs to reformulated to allow DM to provide, although not compulsory, more information for better control on the optimization process. Readers will be guided on the formulation of domination comparison scheme to accommodate defined augmented specifications in the following sections. We start with the modified domination comparison scheme with defined superiority threshold, then progressively extend the scheme to include other defined satisfactory level and goal.

4.2.1 Extended Domination Comparison Scheme for Superiority Threshold

Extending from Horn et al. [10] definition of Pareto domination scheme on cost functions and without loss of generality, an objective vector F_a in a minimisation problem is said to dominate another objective vector F_b in m-dimensional objective space with respect to superiority threshold, α, if the below sufficiency condition is satisfied.

$$\begin{gathered} F_a \underset{\alpha}{\prec} F_b, \textit{ iff} \\ f_{a,i} \le f_{b,i}\ \forall\ i \in \{1,2,\ldots,\ m\} \text{ and} \\ \exists\ i\!:\ f_{a,i} < \begin{cases} f_{b,i} - \alpha_i & ,\alpha_i \ne null \\ f_{b,i} & ,otherwise \end{cases} \end{gathered} \tag{6}$$

The formulation considers that, for a cost function with defined superiority threshold, the cost function must satisfy the threshold in order to dominate another cost function. The idea is to allow DM to have control on the sensitivity in performance comparison and only differentiate the two solutions when their difference in c.ost functions is practically significant enough.

4.2.2 Extended Domination Comparison Scheme for Satisfactory Level

Continuing from the domination comparison for superiority threshold, an objective vector F_a in a minimisation problem is said to dominate another objective vector F_b in m-dimensional objective space with respect to satisfactory level, β, if it satisfies the below sufficiency condition.

$$F_a \underset{\beta}{\prec} F_b \textit{ iff } F_a^{(\beta)} \underset{\alpha}{\prec} F_b^{(\beta)} \tag{7}$$

where,

$$\begin{gathered} F^{(\beta)} = \left\{ f_i^{(\beta_i)}\!: \forall\ i \in \{1,2,\ldots,\ m\right\} \text{ and} \\ f_i^{(\beta_i)} = \begin{cases} \beta_i & ,\beta_i \ne null \text{ and } f_i < \beta_i \\ f_i & ,otherwise \end{cases} \end{gathered} \tag{8}$$

The idea in the above formulation is to stop further differentiating the performances of two individuals where the defined satisfactory levels has been met. Once the defined satisfactory level is met, they are considered satisfied and further improvement along the cost function is meaningless. This give more room for improvement in other cost functions where the satisfactory level is undefined or met.

4.2.3 Extended Domination Comparison Scheme for Goal

Two-stage Pareto domination comparison scheme is adopted here in considering the defined goal in cost function. Adopting comparison scheme in (6) and a rank value begins from one, the first stage ranks all individuals that satisfy the goal if available, starting from the value of rank one, with reference to the infimum in the objective domain. The second stage is to rank the remaining individuals that do not meet the goal based upon the following extended domination scheme. It should be noted that the ranking method remains as range-independent ranking, which is the most appropriate type of ranking method to be used for general-purpose MO optimization problems. Let ${F_a}^{\hat{\gamma}}$ and ${F_b}^{\hat{\gamma}}$ denotes the vectors F_a and F_b that do not meet the goal γ. Then for both F_a and F_b that do not totally satisfy the goal, the vector F_a is said to dominate vector F_b denoted by

$${F_a}^{\hat{\gamma}} \underset{\gamma}{\prec} {F_b}^{\hat{\gamma}} \; iff \; |F_a - \gamma| \underset{\beta}{\preceq} |F_b - \gamma| \quad (9)$$

where $|F- \gamma| = \{|f_i - \gamma_i|: \gamma_i \neq null \;\forall\; i = 1, 2, \ldots m\}$. For the individuals outside the goal, the rank value begins from one increment of the maximum rank of individuals inside the goal. The domination comparison scheme in (9) directs the individuals outside the goal towards the goal, which is modified from Khor et al. [23] and different from Fonseca and Fleming [8] domination scheme.

4.3 Logical Operations in General MO Expression

After constructing the ranking $\langle\boldsymbol{\theta}\rangle$ of individuals with respect to objectives, $\boldsymbol{\theta}$, this section introduces the logical operations in the MO expression to describe the relations among multiple sets of specification. Without loss of generality, the MO expression can be composed of ranks $\langle\boldsymbol{\theta}\rangle$ or $\langle\boldsymbol{R}\rangle$, as operands and logical operations, $\boldsymbol{\Omega} = \{\wedge, \vee, \neg, \Rightarrow\}$, as operations. Please be noted that the input and output of a logical operation are rank vector, R, of a set of specification.

4.3.1 Conjunction Operation: ∧ (AND)

Given two rank vectors R_a and R_b of length S with $r_{a,i}$ and $r_{b,i}$ representing i-th rank value in the respective vector, a conjunction operation (AND) in MO expression is defined as:

$$R_a \wedge R_b = \langle R_{a\wedge b}\rangle \quad (10)$$

where,

$$R_{a\wedge b} = \{r_i := \max(r_{a,i}, r_{b,i}) \;\forall i = 1, 2, \ldots S\} \quad (11)$$

Basically, for each individual, it take the worst rank of the individual. Then the resulted values in the group are ranked again. The operation is commutative, meaning that $R_a \wedge R_b = R_b \wedge R_a$ and variadic, i.e. expandable to any number of rank vectors ($R_b \wedge R_a \wedge R_c \ldots$). The use of this conjunction operation is for DM to combine multiple rank vectors by find solutions at least not extremely inferior in any rank vector. In set theory, it can be interpreted as an attempt to find the intersection of different performances. If the intersection is not feasible, meaning that there is no individual superior in all the specified performances, it will at least solutions to accommodate all performances moderately, i.e. at least not too bad in any one.

4.3.2 Disjunction Operation: ∨ (OR)

In MO expression, a disjunction operation (Or) for two rank vectors, R_a and R_b of length S with $r_{a,i}$ and $r_{b,i}$ representing i-th rank value in the respective vector is defined as:

$$R_a \vee R_b = \langle R_{a\vee b}\rangle \quad (12)$$

where,

$$R_{a\vee b} = \{r_i := \min(r_{a,i}, r_{b,i}) \;\forall i = 1, 2, \ldots S\} \quad (13)$$

In contrast with conjunction operation, it takes the best rank of the individual. The idea is to unify multiple sets of performances so that individuals can maintain their superiority in their own niches. Similar to conjunction operation, it is commutative and variadic.

4.3.3 Negation Operation: ¬ (NOT)

For a given rank vector, R_a, a negation operation in MO expression is defined as:

$$\neg R_a = \langle R_{\neg a}\rangle \quad (14)$$

where,

$$R_{\neg a} = \{\, r_i := s + 1 - r_{a,i} \;\forall\; i = 1, 2, \ldots S\} \quad (15)$$

It is a logical operator that is used to invert the ranking in the input rank vector and still fulfilling the rank format in (4). The operation is reversible, i.e. $\neg(\neg R) = R$. It provides an option for DM to include in MO expression to search for certain characteristic or criterion for their desired outputs.

4.3.4 Lexicographic Order Operation: ⇒ (Higher Priority Than)

A lexicographic order operation in MO expression for two rank vectors, R_a and R_b, of length S, where R_a has higher priority than R_b, is defined as:

$$R_a \Rightarrow R_b := \langle r_i^{(a\Rightarrow b)} \forall\; i = 1, 2, \ldots S \rangle \quad (16)$$

where,

$$r_i^{(a\Rightarrow b)} = \sum_{r=0}^{r_{a,i}-1} \mu_r + rank\left(r_{b,i} \middle| r_{a,i}\right) \quad (17)$$

$rank(r_{b,i}|r_{a,i})$ represents the rank value of $r_{b,i}$ within the sub-group that $r_a = r_{a,i}$. μ_r is number of candidates having the rank r. The idea is to take the higher priority rank, i.e. R_a, as the base rank. The individuals that have equal rank in R_a, is further ranked within the sub-group according to lower priority rank R_b. $rank(r_{b,i}|r_{a,i})$ always begin the rank value 1 for each subgroup of $r_{a,i}$. It should be noted that the operator is not commutative, which means $R_a \Rightarrow R_b \neq R_b \Rightarrow R_a$. It allows DM to express the order of importance among multiple MO specifications. The arrow direction indicates the specifications are arranged in the order of reducing priority. Table 1 presents a numerical example of lexicographic order operation on a group of 10 individuals (S=10) with R_a and R_b as the input rank

Paper ID: SR24511075354 DOI: https://dx.doi.org/10.21275/SR24511075354

vectors. The output rank vector of the operation is shown in the last column as explained above. It can be noted that the operation results in the output rank is maintaining the rank format defined in (4) as far as the input rank vectors are in the similar rank format.

Table 1: Numerical Example of Lexicographic Order Operation

Individual	R_a	R_b	$\sum_{r=0}^{r_a-1} \mu_r$	$rank(r_b \mid r_a)$	$R_a \Rightarrow R_b$
1	1	8	0	1	1
2	1	10	0	3	3
3	1	8	0	1	1
4	4	5	3	2	5
5	4	3	3	1	4
6	6	6	5	2	7
7	6	6	5	2	7
8	6	2	5	1	6
9	9	1	8	1	9
10	10	3	9	1	10

5. General Expression for MO Decision Support

After introducing the operands and operations, this section presents the syntactic construct of a complete MO Expression for MO optimization. MO expression is DM's mathematical formulation of a general description of MO optimization preferences. For demonstration purpose, below are some examples on how DM can elaborate their preferences systematically and formally in the general MO expression.

Example 1:
This is an example of MO Expression with different combinations of superiority threshold, satisfactory level and goal is shown below:

Definition of objectives:
$\theta_1 = (f_1, null, null, null)$, $\theta_2 = (f_2, null, \beta_{2,1}, null)$,
$\theta_3 = (f_3, null, null, \gamma_{3,1})$, $\theta_4 = (f_4, \alpha_{4,1}, null, null)$,
$\theta_5 = (f_5, \alpha_{5,1}, \beta_{5,1}, null)$, $\theta_6 = (f_6, \alpha_{6,1}, null, \gamma_{6,1})$.

Below is a MO expression to minimize each objective separately with optionally defined attributes. It is noted that DM can flexibility define any attributes in each objective in the MO optimization.

$$\langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 \rangle$$

Any objective must be enclosed by ranking "⟨ ⟩" operator. For example, "$\langle \theta_1, \theta_2 \rangle, \theta_3$" is not proper in MO expression as θ_3 is not enclosed by "⟨ ⟩". Although the above example involves different function for each objective, the same function can be repeated in different objectives according to DM's requirements.

Example 2:
Below is an example reconstructing the conventional MO optimization problem in (1) in the MO expression. The objectives in the problem can be expressed as below:

$$\theta_i^{(f)} = (f_i, null, null, null)\ \forall i = 1,2, \dots, C$$
$$\theta_j^{(g)} = (g_j, null, 0,0)\ \forall j = 1,2, \dots, J$$

The difference between $\theta_i^{(f)}$ and $\theta_j^{(g)}$ is that the former has undefined satisfactory level and goal while the latter's ones are defined. It is noted that both satisfactory level and goal are defined for a given constraint in conventional optimization. Goal direct the search while satisfactory level indicate further improvement in a function is meaningless upon satisfied. Comparing to (1), MO expression provide more flexibilities for DM to express their preference more precisely. Below are some options for DM.

i. MO expression to treat cost functions and satisfactory levels separately:
$$\langle \theta_1^{(g)}, \theta_2^{(g)}, \dots, \theta_J^{(g)}, \theta_1^{(f)}, \theta_2^{(f)}, \dots, \theta_C^{(f)} \rangle$$

ii. MO expression to meet constraints g_j as higher priority, followed by the cost functions f_i:
$$\langle \theta_1^{(g)}, \theta_2^{(g)}, \dots, \theta_J^{(g)} \rangle \Rightarrow \langle \theta_1^{(f)}, \theta_2^{(f)}, \dots, \theta_C^{(f)} \rangle$$

iii. MO expression to find the intersection between constraints g_j and cost functions f_i:
$$\langle \theta_1^{(g)}, \theta_2^{(g)}, \dots, \theta_J^{(g)} \rangle \wedge \langle \theta_1^{(f)}, \theta_2^{(f)}, \dots, \theta_C^{(f)} \rangle$$

Different ways of MO expression will shape the search behavior differently and is based on DM preferences.

Example 3:
Since exact attainable cost regions in many real-world optimization problems may be unknown, one often requires to "guess" for an appropriate initial goal value and manually observe the evolutionary optimization process [30,31]. If any of the goals is too stringent or too generous, the goal setting will have to be adjusted accordingly until a satisfactory solution can be obtained. This approach obviously requires extensive human observation and decision making which can be tedious and inefficient in practice. To reduce human interaction and to support for higher-level decision-making, logical relations among multiple goals can be applied. To illustrate with example, lets define two sets of goal in 2-dimensional objective domain as below.

Objectives for goal 1:
$\theta_1^{(f_1)} = \left(f_1, \times, \times, \gamma_1^{(f_1)}\right)$, $\theta_1^{(f_2)} = \left(f_2, \times, \times, \gamma_1^{(f_2)}\right)$

Objectives for goal 2:
$\theta_2^{(f_1)} = \left(f_1, \times, \times, \gamma_2^{(f_1)}\right)$, $\theta_2^{(f_2)} = \left(f_2, \times, \times, \gamma_2^{(f_2)}\right)$

Below are examples of the MO expression to attain the two goals using conjunction and disjunction operations.

i. MO expression to satisfy goals 1 and 2, which tends to find the intersection of the two goals:

$$\langle \theta_1^{(f_1)}, \theta_1^{(f_2)} \rangle \wedge \langle \theta_2^{(f_1)}, \theta_2^{(f_2)} \rangle$$

ii. MO expression to satisfy goals 1 or 2 which tends to satisfy the two goals separately.

$$\langle \theta_1^{(f_1)}, \theta_1^{(f_2)} \rangle \vee \langle \theta_2^{(f_1)}, \theta_2^{(f_2)} \rangle$$

With the operations, multiple set of specifications are allowed to specify different portion of trade-offs in a single run of optimization process. Besides, multiple goals can be in a way that one is more stringent than others to direct the search through stages of stringency defined by DM until the best trade-off, where the most stringent stage may not necessary be feasible.

Example 4:
We can adopt lexicographic order operation to prioritize objectives differently. Here we characterize a lexicographic order is "weak" when the objectives at the lower priority is a subset of objectives at higher priority. That is for any lexicographic order between two objectives $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_q$ given as $\langle \boldsymbol{\theta}_p \rangle \Rightarrow \langle \boldsymbol{\theta}_q \rangle$, the lexicographic order is weak *iff* $\boldsymbol{\theta}_q \epsilon \boldsymbol{\theta}_p$. Weak lexicographic order considers bigger set of objectives first as big picture before zooming in the focus to subset of objectives within the range. Here are the examples to illustrate the difference where the former is weak lexicographic order while the latter is the normal one:

i. In, weak lexicographic order, higher priority given on the bigger picture first, e.g. $\langle \theta_1, \theta_2, \theta_3, \theta_4 \rangle$, followed by the subset of the former one, i.e. $\langle \theta_1, \theta_2 \rangle$.

$$\langle \theta_1, \theta_2, \theta_3, \theta_4 \rangle \Rightarrow \langle \theta_1, \theta_2 \rangle$$

ii. MO expression for normal lexicographic order where $\langle \theta_1, \theta_2 \rangle$ is given higher than $\langle \theta_3, \theta_4 \rangle$. It can be observed that $\{\theta_3, \theta_4\} \not\in \{\theta_3, \theta_4\}$.

$$\langle \theta_1, \theta_2 \rangle \Rightarrow \langle \theta_3, \theta_4 \rangle$$

When there are many objectives in optimization and it is hard to meet all objectives at once. Lexicographic order operator would be a useful tool to guide the search through order of priority rather than all objectives are equally focused.

Example 5:
We can also apply negation operation in MO expression to characterize the type of solutions we desire to search for. In this example, DM desires to unify solutions for two niches in a complementary way in objective domain. More specifically, the solutions are superior in the first set of objectives but not in the second set or vice versa. Let the first set of objectives represents $\langle \theta_1, \theta_2 \rangle$ while the second $\langle \theta_3, \theta_4 \rangle$. Then MO expression for the above specification would be:

$$\left(\langle \theta_1, \theta_2 \rangle \wedge (\neg \langle \theta_3, \theta_4 \rangle)\right) \vee \left((\neg \langle \theta_1, \theta_2 \rangle) \wedge \langle \theta_3, \theta_4 \rangle\right)$$

The above examples show that the General MO Expression exhibits wide range of flexibility for DM to formally and intuitively elaborate their preferences as close as possible to their needs. In real life, it may be difficult to fulfill all objectives at once and/or results in too wide range of solutions if formulated in conventional MO optimization problem and there is a need to have more elaborative and intuitive description from DM to better guide the search towards their preferred solutions.

6. Evolutionary Optimization with General MO Expression

Evolutionary-based search methods have been applied successfully to solve multi-objective optimization problems to optimize for a set of global Pareto-optimal solutions owing to their nature of population-based to evolve a group of solutions concurrently without the need of derivative information in the search space. Another reason why evolutionary MO optimization is chosen is its capabilities in incorporating cognitive specification like priority information that indicates the relative importance of the multiple tasks to provide useful guidance for the optimization. In general, the proposed MO performance expression can be applied with various population-based, including evolutionary-based, search and optimization. For experimentation and demonstration in this paper, we adopt the MO evolutionary framework in [18,32] for the search and optimization.

In the framework, a list of individual chromosomes is initialized at the initial stage of the evolution via randomization. The individuals are then decoded to parameter vectors. Then the individuals undergo performance evaluation according to the General MO Expression elaborated from DM to compute the final rank values. Subsequently, all the ranked individuals are fed to fitness sharing to avoid over degradation for individuals closely crowd together and to suite tournament selection or other evolutionary selection that are based on direct fitness comparison. If the stopping criterion is not met, the evaluated individuals are then undergone a series of evolutionary operations, consisting of deterministic tournament selection, ordered crossover as well as mutation, to reproduce the offspring for next generation. A number of elite individuals based on shared costs are preserved in next generation to encourage stability and diversity of the evolution at the Pareto-optimal frontier. This evolution cycle is repeated until the stopping criteria is met. The details of the MO evolutionary algorithms can be found in [18].

7. Application in Student Internship Planning

This section presents an application case study of the proposed General MO Expression model where MO optimization of student internship plans in higher education is investigated. Given a set of students and companies respectively, the problem is to find optimum matching of a batch of students to their intern jobs on MO specifications. Section 7.1 lists down the possible cost functions to be considered. Section 7.2 describe the implementation details search and optimization while the optimization results are presented and discussed in Section7.3 for different variety of preferences from DM.

7.1 Cost Function Formulation

The practical criteria to consider with their respective cost functions of the student internship planning are given below:

7.1.1 Job Qualification

It is important to consider how good the intern entry qualifications are able to meet company expectation. The qualification is a list of score components, which can be in terms of modules taken or areas of competency, concerned by companies. For a score component k, let $\lambda_{i,k}^{(st)}$ represents score obtained by student i and $\lambda_{j,k}^{(co)}$ represents the expected score by of company j. The cost function f_1 for a batch of students meeting the job qualification is

$$f_1 = \frac{\sum_{i=1}^{U} \sqrt{\sum_{k=1}^{Q} \left(max\left(\lambda_{co(i),k}^{(co)} - \lambda_{i,k}^{(st)}, 0 \right) \right)^2}}{U} \tag{18}$$

where U is the number of students in the batch, Q the number of score component and $co(i)$ the company where student i is attached to. The cost function takes the average gap of students in meeting the job qualifications by the attached companies. $f_1 \geq 0$ and $f_1 = 0$ indicates all students meet all their company expectations.

7.1.2 Task Preference

Meeting student expectations on the tasks involved in their attached companies in their internship is another important criterion for fruitful learning and experience aligned to their desired career paths in future. The job performed can described by a list of tasks involved with their weightages. For a task k in a task list, let $\tau_{i,k}^{(st)}$ represents the expected task weightage by student i and $\tau_{co(i),k}^{(co)}$ the task weightage performed in attached company for the student. The cost function f_2 in meeting task preferences from a batch of students is

$$f_2 = \frac{\sum_{i=1}^{U} \sqrt{\sum_{k=1}^{T} \left(\tau_{i,k}^{(st)} - \tau_{co(i),k}^{(co)} \right)^2}}{U} \tag{19}$$

where T is number of tasks in the overall task list. It is noted that $0 \leq \tau \leq 1$ and $\forall i = 1,2,\ldots U$ as shown below:

$$\sum_{k=1}^{T} \tau_{i,k}^{(st)} = 1, \sum_{k=1}^{T} \tau_{co(i),k}^{(co)} = 1 \tag{20}$$

7.1.3 Work Schedule

In terms time, the criterion to consider is work schedule. Different job may have different types of work schedule. For example, the work can be on 8-hourly shift, 12-hourly shift, overtime etc. A company may have one work schedule type that is fine or not expected by a student, depending on the job nature versus the student acceptance for the work schedule. Thus, it is crucial to be included into the planning. Let W be the possible number of schedule types in the job pool, f_3 is the cost function in matching the companies and students in terms of work schedule as given below:

$$f_3 = \frac{\sum_{i=1}^{U} \sum_{k=1}^{W} max\left(\omega_{co(i),k}^{(co)} - \omega_{i,k}^{(st)}, 0 \right)}{U} \tag{21}$$

where $\omega_{i,k}^{(st)} = 1$ if the schedule type k is fine with student i and equals 0 otherwise. $\omega_{co(i),k}^{(co)}$ is 1 if the schedule type k is required by company $co(i)$ where student i is attached and 0 otherwise.

7.1.4 Salary Expectation

In addition to the types of job, salary matching between the student expectation and company offer is also another important factor to consider. As companies offer different amount of salary while students have their own expectations, it is good to reduce the gap between them as much as possible. Given $\delta_i^{(st)}$ as expected salary of student i and $\delta_{co(i)}^{(co)}$ the salary offered by internship company for the student, the cost function f_4 to meet salary expectation for the whole batch is given as:

$$f_4 = \frac{\sum_{i=1}^{U} max\left(\delta_i^{(st)} - \delta_{co(i)}^{(co)}, 0 \right)}{U} \tag{22}$$

7.1.5 Travelling Cost

Reducing traveling helps in both environmental and economic sustainability in community as it reduces energy consumption, carbon emission, traffic loading and congestion. For students, they can also benefit from reducing traveling times, traveling expenses as well as accident risks and unforeseen disruption along the journeys. It can also be an intangible factor to their punctualities and motivations in their job. Thus, the cost function in terms of incurred traveling cost is to be considered in the optimization. Let $\eta_{i,co(i)}$ denotes the traveling cost incurred for the internship between student i and the attached company $co(i)$, the cost function of traveling cost, f_5, for the batch of students in the internship is

$$f_5 = \frac{\sum_{i=1}^{U} \eta_{i,co(i)}}{U} \tag{23}$$

Without loss of generality, the traveling cost can be quantified in terms of traveling time or traveling distance.

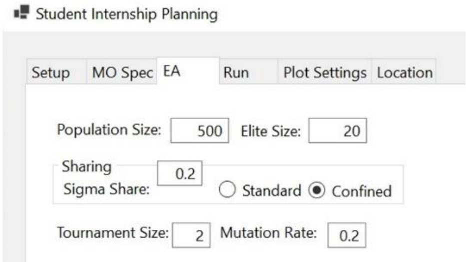
7.2 Implementations

In the case study, there are 40 students in the batch going for internship. Each student provides inputs of 6 module scores, ranging from 0 (lowest range) to 4 (highest range), for their job qualifications, weightages (0 to 1) of their expected tasks, out of 12 tasks in the list, for task preferences, their acceptable work schedule types, out of 3 types including normal shift, their expected salary and their geographical locations. As for the internship jobs, 40 jobs from various companies in the list. Similarly, for each job offered by the respective company, the inputs of job qualifications expected, weightages of the tasks performed, work schedule types, salary offered and the geographical locations of the job in the same format as student data are collected.

For the search and optimization process, the MOEA mentioned in Section 6 is configured as follow: sigma share = 0.2, tournament size = 2 and mutation rate = 0.2. The population size of 500 is initialized. The evolutionary process is run with elite size of 20 and generation 1000 stopping criteria. The parameters settings in the developed software is illustrated in Figure 1.

Paper ID: SR24511075354 DOI: https://dx.doi.org/10.21275/SR24511075354

(a) Settings in Optimization Problem



(b) Settings in Evolutionary Process

Figure 1: Implementation Setup

7.3 Demonstrations and Discussions

Among the five possible cost functions defined in previous sections, DM have choices to express their preferences in many possible ways, according to their interest and requirements, using the proposed General MO Expression. Thus, we will look into different forms of MO expression DM can use to express their preferences for the search and optimization.

7.3.1 MO Expression for Simplest Optimization Preference

Let start with the simplest DM preferences where all the five cost functions are to be optimized separately without any augmented specification. The MO expression can be described as below where each objective is purely attributed by the respective cost function alone.

$$\text{Minimize: } \langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 \rangle$$

where, (24)

$$\theta_i = (f_i, null, null, null)\ \forall i = 1,2,\ldots,5$$

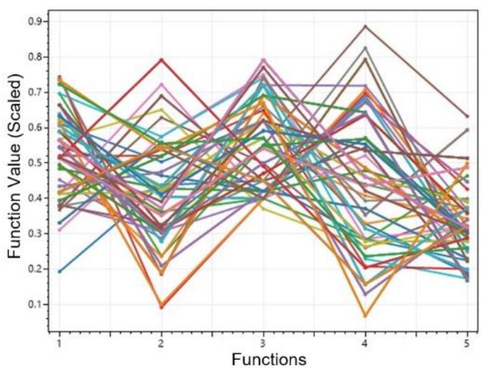


Figure 2: Non-dominated Solutions for Optimizing All Functions

Figure 2 depicts the experimental results for the above MO expression in the end of the run. Each line represents one of the best found non-dominated solutions where x-axis is the cost function labels while y-axis is the respective function values. For better visualization, the function values are scaled differently for each function. There are 137 non-dominated solutions, which form the Pareto-front, in the run.

7.3.2 MO Expression for Optimizing with Augmented Specifications

Augmented specifications can be included in the MO Expression to provide more directive guide in the optimization. Below is an example of DM's specified MO expression applying the augmented specifications.

$$\text{Minimize: } \langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 \rangle$$

where,

$$\begin{aligned} \theta_1 &= (f_1, null, null, 0.7), \\ \theta_2 &= (f_2, null, 0.65, null), \\ \theta_3 &= (f_3, null, null, null), \\ \theta_4 &= (f_4, 3,50,50), \\ \theta_5 &= (f_5, 0.2,18,18) \end{aligned} \tag{25}$$

Unlike the previous MO expression, objective 1 has goal of 0.7, objective 2 has satisfactory level of 0.65, and so forth for objectives 3 to 5. It is noted that DM is free to have any combination of the augmented specifications in the objectives based on their prior knowledge and requirements.

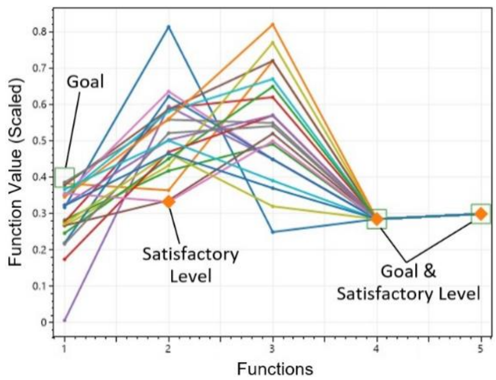


Figure 3: Non-dominated Solutions for Optimizing with Augmented Specifications

The resulted non-dominated solutions in the run is presented in Figure 3. Results shows that the goals (open squares) in functions 1,4 and 5, direct the search towards more focus area on the trade-off. The goals are feasible in this case as the function values are below the goal values. For the functions 2, 4 and 5 having satisfactory levels (solid diamonds), the function values equal the satisfactory levels once met as further minimization in the functions is not required. It allows more focus on other functions, i.e. functions 1 and 3 as shown in the figure, to achieve better results. It is noted that both satisfactory level and goal are defined in functions 4 and 5, to effect in the same requirement as constraints. That is, the search should be directed towards other objectives where the satisfactory level in undefined or not met.

7.3.3 MO Expression for Optimizing with Priority Order

DM can specify the priority order in MO expression construct if they want to distinguish different level of importance in the preference. MO expression below specify that objectives 1 to 3 are given higher priority than 4 and 5.

$$\text{Minimize: } \langle \theta_1, \theta_2, \theta_3 \rangle \Rightarrow \langle \theta_4, \theta_5 \rangle \tag{26}$$

Paper ID: SR24511075354 DOI: https://dx.doi.org/10.21275/SR24511075354

where,

$$\theta_i = (f_i, null, null, null)\ \forall i = 1,2, \ldots ,5$$

Figure 4 shows the resulted non-dominated solutions in the run. Comparing to results in Figure 2, the search is gives relatively better performances in objectives 1 to 3 since they are given higher priority. Needless to say, the improvement is on the expense of lower priority objectives (4 and 5). It is also observable that, each overall non-dominated solution is non-dominated within two separate groups of objectives, i.e. $\{\theta_1, \theta_2, \theta_3\}$ and $\{\theta_4, \theta_5\}$.

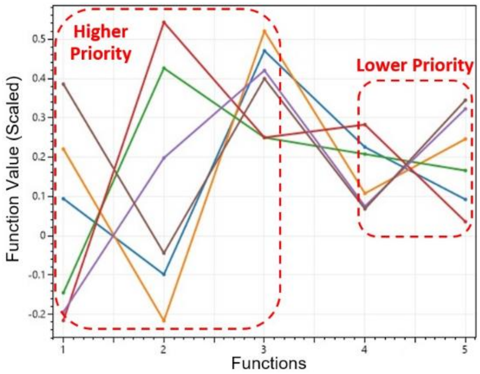


Figure 4: Non-dominated Solutions for Optimizing with Priority Order

7.3.4 MO Expression for Optimizing with Weak Priority Order

If DMs desire to optimize for the overall performances of the objectives first before further zooming into some of the objectives, they can use weak priority order in the MO expression. Below is an example to demonstrate that where the higher priority is to take care of the overall performance on the objectives and only when they are comparable, subsequent attention will be given to objectives 4 and 5.

$$\text{Minimize: } \langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 \rangle \Rightarrow \langle \theta_4, \theta_5 \rangle \quad (27)$$

where,

$$\theta_i = (f_i, null, null, null)\ \forall i = 1,2, \ldots ,5$$

The experimental results are depicted in Figure 5. It can be seen than, comparing to results in Figures 2 and 4, the run gives slightly better performances in objectives 4 and 5 as they are further improved after taking all objectives into consideration. It can also be observed that the performances of objectives 1 to 3 are not as good as (26) due to the first priority in current MO expression is more diverse, covering all objectives, than (26).

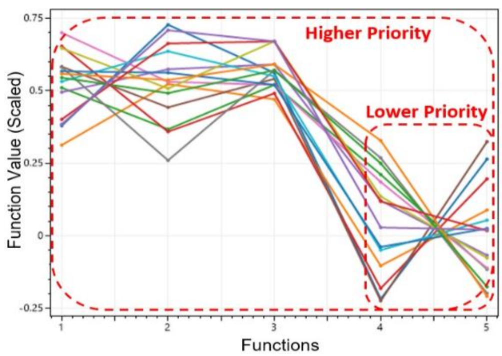


Figure 5: Non-dominated Solutions for Optimizing with Weak Priority Order

7.3.5 MO Expression for Attaining Multiple Goals with "OR" Operation

DM can adopt logical statement in the MO expression to visit different region of trade-offs in objective space in a single run. For demonstration, let a MO optimization is to visit two separate trade-off regions directed by two goals as expressed below:

$$\text{Minimize: } \langle \theta_{4,1}, \theta_{5,1} \rangle \vee \langle \theta_{4,2}, \theta_{5,2} \rangle$$

where,

$$\begin{aligned} \theta_{4,1} &= (f_4, null, null, 40), \\ \theta_{5,1} &= (f_5, null, null, 16), \\ \theta_{4,2} &= (f_4, null, null, 50), \\ \theta_{5,3} &= (f_5, null, null, 13) \end{aligned} \quad (28)$$

It can be understood from the above expression that the optimization involves functions (f_4, f_5) and there are two goal vectors, i.e. (40,16) & (50,13), in disjunction (Or) operation to find the unification of solutions for each goal-specified region. The scatter plot of solutions in the function space (f_4 vs f_5) is shown in Figure 6. The connected solid dots represent the non-dominated solutions while the open circles represent the dominated ones. Result show that the above MO expression is able to direct the search towards attaining multiple goal-specified regions concurrently in single run.

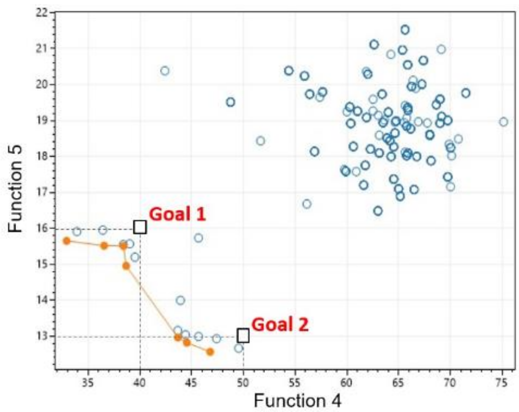


Figure 6: Non-dominated Solutions for Optimizing Multiple Goals with OR Operation

7.3.6 MO Expression for Multiple Non-Dominated Sets with "AND" Operation

Besides the use of logical operation for multiple specifications on the set of objectives, DM can also apply the MO Expression for multiple sets of objectives with logical operation. Here is an example to attain two different sets of goals in two different function spaces.

Minimize:

$$\langle \theta_1, \theta_2 \rangle \wedge \langle \theta_4, \theta_5 \rangle$$

where,

$$\begin{aligned} \theta_1 &= (f_1, null, null, 0.7), \\ \theta_2 &= (f_2, null, null, 0.65), \\ \theta_4 &= (f_4, null, null, 50), \\ \theta_5 &= (f_5, null, null, 18) \end{aligned} \quad (29)$$

The "AND" operation specify that the solution must be non-dominated within both (f_1, f_2) and (f_4, f_5) spaces, where the requirement is more stringent than being non-dominated within (f_1, f_2, f_3, f_4) spaces. The optimized non-dominated solutions are shown in Figure 7. It can be seen that besides meeting multiple goals in multiple sets of functions, the solutions are non-dominated in each respective set of functions, i.e. the lines are crossing each other both sets of functions (enclosed by the dotted boxes). It can also be observed that as the requirement is more stringent than (25), less non-dominated solutions are found.

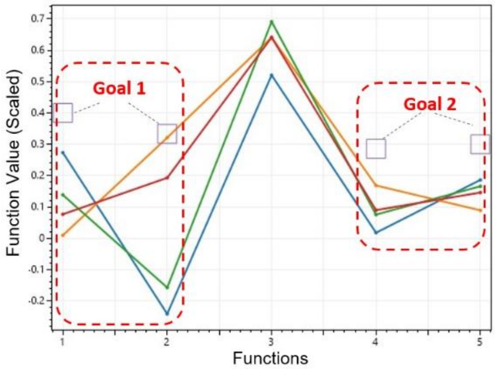


Figure 7: Non-dominated Solutions for Multiple Non-Dominated Sets with AND Operation

7.3.7 MO Expression with Nested Ranking

Ranking operations can be nested and arranged in hierarchies in MO Expression so that DM can subgroup the objectives for ranking (inner) within each group before overall ranking (outer) among the subgroups. An example of nested ranking is given by

Minimize:

$$\langle\langle\theta_1,\theta_2\rangle,\langle\theta_4,\theta_5\rangle\rangle$$

where,

$$\begin{aligned}\theta_1 &= (f_1, null, null, 0.7),\\ \theta_2 &= (f_2, null, null, 0.65),\\ \theta_4 &= (f_4, null, null, 50),\\ \theta_5 &= (f_5, null, null, 18)\end{aligned} \quad (30)$$

The resulted non-dominated solutions are shown in Figure 8 in rank space of $\langle\theta_1,\theta_2\rangle$ vs $\langle\theta_4,\theta_5\rangle$. It can be observed that they form Pareto front (outer rank =1) in terms of the two inner ranks, instead of function values. The global trade-off is formed from the left-most solution, having low first inner rank value but large second inner rank value, to the right-most solution, having the opposite feature. With nested ranking, DM is able to examine more structured trade-off among subgroups of objectives in hierarchies, rather than treating all the objectives equally in a big group.

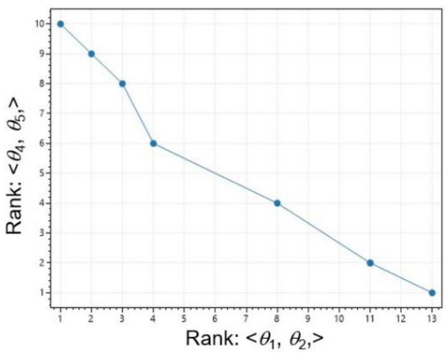


Figure 8: Non-dominated Solutions for Nested Ranking in Rank Space

7.3.8 MO Expression with Nested Operations

Besides ranking, the logical operations can also be arranged in nested structure in the MO Expression to enable DM to formulate their preferences to direct the search towards searching for optimal solutions in more complicated requirements. Here is an example.

Minimize:

$$(\neg\langle\theta_1,\theta_2\rangle \wedge \langle\theta_4,\theta_5\rangle) \vee (\langle\theta_1,\theta_2\rangle \wedge \neg\langle\theta_4,\theta_5\rangle)$$

where,

$$\begin{aligned}\theta_1 &= (f_1, null, null, 0.7),\\ \theta_2 &= (f_2, null, null, 0.65),\\ \theta_4 &= (f_4, null, null, 50),\\ \theta_5 &= (f_5, null, null, 18)\end{aligned} \quad (31)$$

The expression is to find the unification of two sets of optimal solutions. Set one is superior in θ_4 and θ_5 but not superior in θ_1 and θ_2, with their respective goal specifications. On the other hand, set two has the opposite characteristics with former set. Figure 9 shows the resulted non-dominated solutions. It can be observed that solutions 1 & 2, as labeled in the figure, is superior in set one but otherwise in set two. In contrary, solution 3 has totally opposite characteristics. As shown in this example, DM can apply MO expression to describe more complicated MO specifications to search for optimal solutions. Instead of minimizing the cost functions alone, MO Expression also enable DM to search for solutions that best characterized by the expression without the complication to reformulate the MO optimization process.

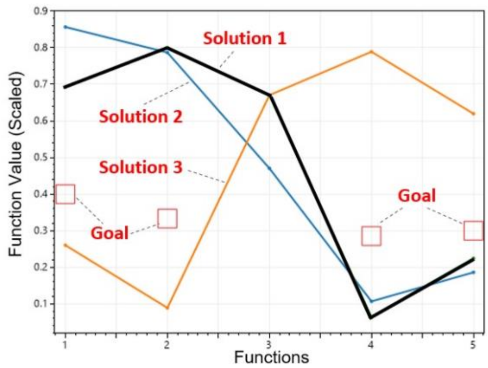


Figure 9: Non-dominated Solutions for Optimizing with Nested Operations

8 Conclusions

This paper proposed a syntactic construct, called General MO Expression, for higher-level decision making to specify their preferences in MO performance assessment. The expression is intuitive, flexible and general enough for DM to incorporate any prior knowledge and/or requirements, such as dominating threshold, satisfactory level and goal, to be optionally augmented in the objectives in addition to the optimization functions. In addition, it also enables DM to elaborate their preferences in terms of logical relations such as "AND", "OR" and "NOT", as well as priority order among the objectives in the formal syntactic construct. Computational formulation is given to systematically compute the performance assessment based on the General MO Expression which can be applied in many population-based search and optimization, including evolutionary search methods. Besides illustrative examples, practical applications in MO student internship planning is implemented and the usefulness for DM specify wide variety of preferences with General MO Expression is demonstrated and discussed. The results have shown that the method is able to search for dominant solutions that are the best in exhibiting the characteristics specified by the MO expression, which is based on DM choice and preference. On the whole, the results have demonstrated the main objective of this paper to provide flexibility for DM to elaborate their preferences more precisely and in wider coverage in a systematic mathematical formulation.

References

[1] C.M. Fonseca and P.J. Fleming, “Multiobjective Optimal Controller Design with Genetic Algorithms,” In Proceedings on IEE Control, pp. 745-749, 1994.

[2] A.J. Chipperfield and P.J. Fleming, “Gas Turbine Engine Controller Design using Multiobjective Genetic Algorithms,” In Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, ed. by A.M.S. Zalzala, pp. 214-219, 1995.

[3] T.H. Liu and K.J. Mills, “Robotic Trajectory Control System Design for Multiple Simultaneous Specifications: Theory and Experimentation,” Transactions on ASME, Vol. 120, pp. 520-523, 1998.

[4] H.A. Thompson and P.J. Fleming, “An Integrated Multi-Disciplinary Optimisation Environment for Distributed Aero-engine Control System Arhitectures.” In Proceedings 14th World Congress of International Federation of Automatic Control, pp. 407-412, 1999.

[5] E.F. Khor, K.C. Tan and T.H. Lee, “Multi-Objective Evolutionary Optimization with Non-Stationary Search Space,” In IEEE Proceedings of the Congress on Evolutionary Computation, in press, 2001.

[6] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[7] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Massachusetts, 1989.

[8] C.M. Fonseca and P.J. Fleming, “Genetic Algorithm for Multiobjective Optimization, Formulation, Discussion and Generalization,” In Genetic Algorithms: Proceeding of the Fifth International Conference, ed. by S. Forrest, pp. 416-423. Morgan Kaufmann, San Mateo, 1993.

[9] N. Srinivas and K. Deb, “Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms,” Journal of Evolutionary Computation, MIT Press, Vol.2, No.3, pp. 221-248, 1994.

[10] J. Horn, N. Nafpliotis and D.E. Goldberg, “A Niched Pareto Genetic Algorithm for Multiobjective Optimisation,” IEEE International Conference on Evolutionary Computation, Vol.1, pp. 82-87, 1994.

[11] M. Valenzuela-Rendón and E. Uresti-Charre, “A Non-Generational Genetic Algorithm for Multiobjective Optimization,” In Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kauffmann, San Francisco, California, pp. 658-665, 1997.

[12] 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.

[13] Tan, K. C., Lee, T.H., Khor, E. F. (2001). ‘Evolutionary algorithm with dynamic population size and local exploration for multiobjective optimization.’ IEEE Trans. on Evolutionary Computation 5(6): 565-588.

[14] M. Giuliani, J.D. Quinn, J.D. Herman, A. Castelletti and P.M. Reed, “Scalable multiobjective control for large-scale water resources systems under uncertainty,” IEEE Trans. Control. Syst. Technol. Vol. 26, pp. 1492–1499, 2017.

[15] W. Yu, B. Li, H. Jia, M. Zhang and D. Wang, “Application of multi-objective genetic algorithm to optimize energy efficiency and thermal comfort in building design,” Energy Build, Vol. 88, pp. 135–143, 2015.

[16] M.J. Mayer, A. Szilágyi and G. Gróf, “Environmental and economic multi-objective optimization of a household level hybrid renewable energy system by genetic algorithm,” Appl. Energy, Vol. 269, 115058, 2020.

[17] O.C. Haas, K.J. Burnham and J.A. Mills, “On Improving Physical Selectivity in the Treatment of Cancer: A Systems Modelling and Optimization Approach,” Control Engineering Practice, Vol. 5, No. 12, pp. 1739-1745, 1997.

[18] K.C. Tan, T.H. Lee and E.F. Khor, “Control System Design Automation with Robust Tracking Thumbprint Performance using a Multi-Objective Evolutionary Algorithm,” IEEE International Conference on Control Applications and System Design, pp. 498-503. 1999.

[19] H.V.H. Ayala and L. Coelho, “Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator,” Expert Syst. Appl., Vol. 39, pp. 8968–8974, 2012.

[20] K. Nisi, B. Nagaraj and A. Agalya, “Tuning of a PID controller using evolutionary multi objective optimization methodologies and application to the pulp and paper industry,” International Journal Machine Learning and Cybernetics, Vol. 10, pp. 2015–2025, 2019.

[21] T. Morimoto, T. Torii and Y. Hashimoto, “Optimal Control of Physiological Processes of Plants in a Green Plant Factory,” Control Engineering Practice, Vol. 3, No. 4, pp. 505-511. 1995.

[22] M.G.C. Tapia and C.A.C. Coello, “Applications of multi-objective evolutionary algorithms in economics and finance: A survey,” Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, pp. 532–539, 2007.

[23] E.F. Khor, T.H. Lee, R. Sathikannan and K.C. Tan, “Evolutionary algorithm with advanced goal and priority specifications for multi-objective optimization,” Journal of Artificial Intelligence Research, Vol. 18, pp. 183-215, 2003.

[24] K. Deb, and A. Kumar, “Interactive evolutionary multi-objective optimization and decision-making using reference direction method,” In Proceedings of the 9th annual conference on Genetic and Evolutionary Computation, pp. 781-788, 2007.

[25] P.K. Jamwal, B. Abdikenov, and S. Hussain, “Evolutionary optimization using equitable fuzzy

sorting genetic algorithm," IEEE Access, Vol. 7, pp. 8111-8126, 2019.

[26] L. Lai, L. Fiaschi, M. Cococcioni, K. Deb, "Handling Priority Levels in Mixed Pareto-Lexicographic Many-Objective Optimization Problems," In: H. Ishibuchi, et al. Evolutionary Multi-Criterion Optimization, EMO 2021. Lecture Notes in Computer Science(), Vol. 12654, Springer, Cham., 2021.

[27] C.A.C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," In Proceedings of the 2000 Congress on Evo-lutionary Computation. CEC00 (Cat. No. 00TH8512), Vol. 1, pp. 30-37, 2000.

[28] L. Rachmawati and D. Srinivasan, "Preference incorporation in multiobjective evolutionary algorithms: A survey," In 2006 IEEE International Conference on Evolutionary Computation, pp. 962-968, 2006.

[29] H. Wang, M. Olhofer and Y. Jin, "A mini-review on preference modelling and articulation in multi-objective optimization: current status and challenges," Complex & Intelligent Systems, Vol. 3, No. 4, pp. 233-245, 2017.

[30] C.A. Coello Coello, An Updated Survey of GA-Based Multiobjective Optimization Techniques, Report No. Lania-RD-98-08, Laboratorio Nacional de Informatica Avanzada (LANIA), Xalapa, Veracruz, Mexico. 1998.

[31] C.M. Fonseca, and P.J. Fleming. "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms–Part I: A Unified Formulation," IEEE Transactions on System, Man, and Cybernetics-Part A: System and Humans, Vol.28, No.1, pp. 26-37, 1998.

[32] E.F. Khor, "Sustainable and Balanced Decision Support for Student Internship Planning', International Journal of Science & Engineering Development Research," Vol. 9:4, pp, 1120 - 1128, 2024.

Author Profile

Eik Fun Khor received Ph.D and B.Eng degrees in Electrical and Computer Engineering from National University of Singapore) and Electrical Engineering from University of Technology, Malaysia, respectively. He has industry experiences in research and development (R&D), manufacturing, process improvement, project management, quality and testing environments as well as lecturing and academic experiences in higher educations. He has 2 US-registered patents, 1 achieved Seagate product awards as recognition for the usefulness in the product, published 2 books, 16 international journals and 17 international conference papers in area of control, computer engineering, optimization and computational intelligence.